Worker-Job Recommendation for Mixed Crowdsourcing Systems: Algorithms, Models, Metrics and Service-Oriented Architecture

A Dissertation
Submitted to the College of Computing and Informatics
of

Drexel University
by

Julian Jarrett
in partial fulfillment of the
requirements for the degree
of
Doctor of Philosophy in Computer Science

June 2017

Abstract

Crowdsourcing is used as model to distribute work over the Internet via an open call to anonymous human workers, who opt to take up work offerings sometimes for some small compensation. Increasingly, crowdsourcing systems are integrated into workflows to provide human computation capabilities. These workflows consist of machine-based workers that work harmoniously on different phases of a task with their human counterparts. This body of work addresses workflows where machines and human workers have the capacity to fulfill the requirements for same tasks. To maximize performance through the delegation of work to the most competent worker, this work outlines a collaborative filtering based approach with a bottom up evaluation based on workers' performance history and their inferred skillsets. Within the model, there are several algorithms, formulae and evaluative metrics. The work also introduces the notion of an Open Push-Pull model; a paradigm that maximizes on the services and strengths of the open call model, while seeking to address its weaknesses such as platform lock-in that affects access to jobs and availability of the worker pool. The work outlines the model in terms of a service-oriented architecture (SOA). It provides a supporting conceptual model for the architecture and an operational model that facilitates both human and machine workers. It also defines evaluative metrics for understanding the true capabilities of the worker pool. Techniques presented in this work can be used to expand the potential worker pool to compete for tasks through the incorporation of machine-oriented workers via virtualization and other electronic services, and human workers via existing crowds. Results in this work articulate the flexibility of our approach to support both human and machine workers within a competitive model while supporting tasks spanning multiple domains and problem spaces. It addresses the inefficiencies of current top-

down approaches in worker-job recommendation through use of a bottom-up approach which adapts to dynamic and rapidly changing data. The work contrasts the shortcomings of top-down approaches' dependency on professed profiles which can be under-represented, over-represented or falsified in other ways with evaluative metrics that can be used for the individual and collective assessment of workers within a labor pool.

Acknowledgement and Dedication

First, I wish to acknowledge **The Lord, God, my Savior Jesus Christ** in providing this life changing opportunity, sustaining me with strength wisdom and providing invaluable support throughout this entire experience.

To all members of my family who have supported me through this process, I will never be able to repay you enough for your continued support. Special mention to my wife, my daily cheerleader and motivation, and my parents, my life coaches. To my son Jace, you don't know yet how much I see myself in you, and I wish that this work will stand as further motivation for your continued scholarship and excellence.

To my academic motivators, Mr. Arnett Campbell for believing in the youth of Jamaica and offering me my initial but crucial foundational experience in Academia. To Dr. Sean Thorpe for encouraging me in the direction of research and connecting me with your supervisor now mine, Prof. Brian Blake.

To Prof. Brian Blake, for continued support, mentorship and guidance throughout this whole process. You have made my experience as a PhD student very comfortable in more ways than you may imagine. You have extended assistance even to my personal life while I have had to adapt to a new country, culture and multiple cities throughout the process.

To the University of Technology, Jamaica, my home university, thank you for so adequately giving me a strong foundation. My readiness for graduate studies was far greater than I had expected. To University of Miami, thank you for hosting me throughout my course work requirements and extending an opportunity to study at your noble institution. To Drexel University, my final destination, thank you for the opportunity to be a part of your vibrant community and for providing an environment that enables academic success.

To my country Jamaica, Land We Love, we are a strong nation of proud people of all colours, races, origins and creed. We are a nation of excellence, I am pleased to acknowledge you in creating another success story in me. Thank you for molding me into the individual I am. I am proud to call myself a citizen of Jamaica. “We likkle, but we Tallawah!!”

This work is dedicated to my family and my future descendants. I hope this work will motivate you all that you come from a good lineage of excellence and that all is possible through **God** and hard work!!

Table of Contents

List of Figures

List of Listings

List of Formulae

List of Tables

Worker-Job Recommendation for Mixed Crowdsourcing Systems:

Algorithms, Models, Metrics and SOA

1.1 Motivation

Crowdsourcing, the contraction of the words crowd and outsourcing (Howe, 2006), describes a paradigm that presents new approaches through which work can be distributed and accomplished through human engagement (Kittur, Chi, & Suh, 2008; Yuen, King, & Leung, 2011). The distribution is enabled by an open call via the Internet; a model that allows individuals to be solicited systematically or anonymously to work on tasks, usually those requiring human intelligence (Schulze. Krug & Schader, 2012). It serves as a computational model especially where machines, technology and algorithms are still inefficient in respective problem domains; humans can accomplish such tasks with relative ease (Doan, Ramakrishnan, & Halevy, 2011). Crowds can answer consensus tasks; where the owners of the tasks believe a solution exists, however the solution lies amongst the collective wisdom in the crowd (Kamar, Hacker, & Horvitz, 2012). With these emerging opportunities, organizations including commercial, non-profit, educational, and government, have seen it fit to engage and harness the wisdom in the crowd to accomplish work (Horton & Chilton, 2010). To take advantage of this, the owners of tasks turn to traditional labor markets to actively engage a rich, diverse worker pool ready to take up tasks. Unfortunately, the use of a general labor micro-task labor platform leads to platform lock-in. This occurs as the task owners establish a dependency on a worker pool for services; it subjects them to platform policies, evolution and strategic direction (Schenk, & Guittard, 2009).

Increasingly, there are classes of problems that have combined the benefits of crowdsourcing with virtualization and machine intelligence within workflows. In these systems, machine and human workers are available to collaborate on different phases of a task. Typically, these mixed systems involve some crowdsourced human-in-the-loop configuration alongside resident machine components that are usually not crowdsourced. Depending on the stage and needs of the workflow, humans perform roles ranging from data input, computation or verification (Dustdar, & Truong, 2012). These systems provide an ability to provision both type of workers on demand which allows for scalability in response to environment constraints. Within the context of this work, these types of systems will be referred to as "*mixed crowdsourcing systems*" (Jarrett, Blake & Saleh, 2017).

Despite success stories with major crowdsourcing labor market platforms like Amazon Mechanical Turk, the open call model, the paradigm's enabler, is also cited as being a primary challenge to success amongst the research community. The challenge is a matter concerning the sustenance of the worker pool through the attraction and retention of crowds (Doan, Ramakrishnan & Halevy, 2011; Quinn, & Bederson, 2011). The human labor pool population in mixed crowdsourcing systems are also threatened by this challenge. An additional challenge to mixed crowdsourcing systems include the difficulty in defining formal uniformed models for the integration of both human and machine-oriented workers within workflows (Candra, Truong & Dustdar, 2013). Systems in related literature presented have outlined collaborative models amongst workers in mixed crowdsourcing systems; they have not outlined competitive models where both types of workers can be crowdsourced and compete for the same tasks (Jarrett, Saleh, Blake, Malcolm, Thorpe & Grandison, 2014). Competitive crowdsourcing systems with mixed workers require information and operational models that can be used to evaluate both types of workers objectively.

1.2 Statement of the Problem

The current open call model of crowdsourcing is perhaps most prominent in engaging an anonymous crowd. In contrast, the open call model also threatens the sustainability of crowd worker pools with respect to the attraction and the retention of crowds, perhaps the most challenging for the paradigm in crowdsourcing related literature (Jarrett, Blake & Saleh, 2017). The paradigm currently does not support the integration of machine workers for crowdsourced tasks; its focus is on harnessing human intelligence and intuition for computation. In this work, a hybrid worker-job recommendation approach is introduced for mixed crowdsourcing systems, that comprises of a class of machine learning oriented algorithms, information and operational models, evaluative metrics and a service-oriented architecture (SOA) based framework. This newly introduced approach, the "Open Push-Pull", is an alternative paradigm to the open call approach that capitalizes its strengths and provide solutions to its major weaknesses.

1.3 Purpose of Research

This work addresses the retention and recruitment challenges associated with the open call model and crowdsourcing platform lock-in through an Open Push-Pull model. It also provides a framework for the recommendation and delegation of work to mixed crowdsourcing systems with competing human and machine worker resources based on their performance history.

1.4 Research Questions

This study addresses the two primary questions. Firstly, when managing crowd computing resources spanning human and machine workers, **what general information models effectively define functional, non-functional and evaluative concerns for both types of workers and a wide cross-section of tasks from diverse problem domains?** *(General Research Question 1)*

Secondly, is there an operational approach that (i) enables systematic and reliable **delegation of work across human and machine work** resources, (ii) creates and incorporates metrics that allow for the **evaluation of collective capabilities of a worker pool,** (iii) facilitates **recommendations as a function of changing environment data** (*i.e. jobs, labor pool, and workers' performance*), and (iv) creates new and innovative **on-demand modes of operation** within a service driven infrastructure? *(General Research Question 2)*

1.5 Research Contributions

The first contribution includes an information model consisting of two key abstract data types (ADT). The first being a task ADT capable of modeling a variety of crowdsource viable tasks across diverse problem spaces and platforms. It also supports a worker ADT which provides for the modeling of human and machine workers, available to take up job offerings. The information model also uses a data store to track evaluations that are then used in the operational model to support learning and the delegation of jobs at decision points within the workflow.

The second contribution includes an operational approach that facilitates the recommendation of mixed workers in a crowdsourcing environment. The approach encompasses the application of a class of machine learning oriented algorithms, the use of evaluative metrics and a pattern inspired SOA-based framework for service implementation and integration. The strategy is tightly coupled with the information model which it consumes to make decisions at decision points throughout workflows.

1.6 Organization of the Work

This work continues with related literature reviewing the characteristics of the crowdsourcing paradigm, crowdsourcing management services, challenges, techniques, strategies approaches and the influence of crowdsourcing in mixed human-machine systems. Chapter 3 follows presenting the technical approach used in the body of work including a worker-job framework for mixed crowdsourcing systems. The section details supporting information, computational, evaluative and operational models. The manuscript continues with chapter 4 outlining experimentation highlighting several studies with their respective findings and results. Chapter 6 follows with the conclusion, contrasting the work done in this body of work with existing approaches, future work and outlook for the research area.

1.7 Summary of Findings

An on-demand, pattern inspired architecture provides the flexibility and adaptability to for an Open Push-Pull paradigm that takes advantage of the state-of-the-art crowd management services while addressing issues in the current Open Call model in which crowdsourcing is currently built-on.

Object-Oriented modeling with reification was found as a reliable technique to create a conceptual data model to support competing human and machine workers and tasks across a wide cross-section of problem domains. Reification reduces coupling, conflict and redundancy in the model, increases cohesion, and supports just-in-time adaptability for new characteristics.

Elastic workflow delegation supports both humans and machine workers in a competitive model. Using an elastic index model, a feedback loop and learning service was found to be a reliable mechanism to assist with worker-job assignment given a worker's previous performance.

A top-down approach using professed profiles were found not to be a reliable reference for making worker-job recommendations or assignments. Profiles under this model tend to be under-represented, over-represented or possibly falsified. A bottom-up approach using proposed individual and community metrics was found to be a more reliable indication of possession of skills via inferencing from previous performance.

In assessing the impact of platform growth (increasing volumes of data) on making recommendations, it was found that a larger worker pool has more of an impact on the time it takes to make predictions as opposed to a smaller worker pool with a higher availability of jobs.

It was also found that the recommendations being made were consistent. The recommender engine was able to customize and adapt recommendations to the needs of individual workers given their selection in jobs, previous performance on other tasks and others within the worker pool performing similar tasks.

2.0 Related Literature

Crowdsourcing can be characterized by examining several enabling features of the paradigm. Initially, systems must incorporate a deep understanding of the role of the working consumer. Secondly, crowd management services must facilitate the interaction of the workers and the providers of tasks. Crowd management services (Figure 1). include crowd recruitment, compensation models, optimizing tradeoffs and the management expertise and skill levels (Jarrett & Blake, 2015). First, this chapter presents the working consumer and its vital role to the paradigm of crowdsourcing. It then shows how the working consumer currently advertises his/her services having an online presence through profiles. Next, the chapter discusses the alignment of the working consumer to jobs through the task selection process and ways providers have employed stimulate recruitment including compensation models. The chapter continues by addressing worker selection for specific tasks through optimizing tradeoffs, the management of expertise and skills, and job recommendation strategies (Table 1). To further access its role in existing systems, it warrants the exploration of ways in which human intelligence through crowdsourcing, is used to facilitate and sustain the functioning of systems via the provision of data input and / or human computation. The chapter presents current work and views pertaining to the features, services and applications of crowdsourcing including its role in mixed crowdsourcing systems. Finally, the chapter concludes with a summary of related work and open concerns / gaps identified.

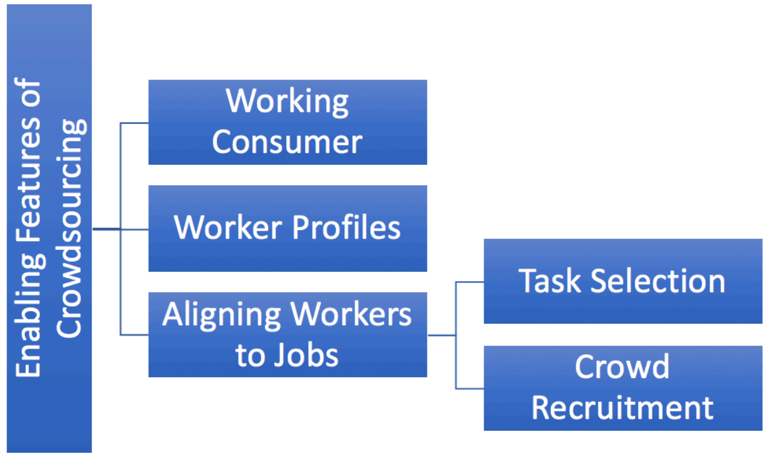

Figure 1. Diagram showing enabling characteristics for crowdsourcing.

2.1 The Working Consumer

The role of the consumer has fundamentally changed within the paradigm of crowdsourcing (Kleemann, Voß & Rieder, 2008). The paradigm provides companies with the opportunity to become consumers of services provided by their clients; in short, the roles are reciprocated where the client becomes the provider and the employers and employees of the companies become the consumers of services outsourced to their clients (Kleemann, Voß & Rieder, 2008). The consumer is no longer seen as the traditional purchaser of products and services, but now as a valued co-worker that can be leveraged for the value creation process (Poetz & Schreier, 2012); they often consume the same products or services influenced by their input. Increasingly, companies have integrated consumers within their production processes with traditional workers, and have assigned to them responsibilities to which work may be delegated (Kleemann, Voß & Rieder, 2008). This integration is valuable especially in early stages, iterative, and continuous development.

2.2 Worker Profiles and Professed Expertise

Worker profiles serve as the online presence for working consumers in a labor force. It typically advertises the worker's reputation and expertise and allows employers to formulate their opinions about the worker. Reputation is an indicator of the worker's reliability as reported by the employers and their collective confidence in the worker. Reputation may consist of a combination of several or more metrics that include, but are not limited to, quality in submissions and timeliness of submissions. Expertise is a showcase of skills, credentials and past experiences that potentially demonstrate competence. It is a common belief that there is a relationship between quality in submissions and worker's profiles; workers with higher reputations submit higher quality submissions (Allahbakhsh, Benatallah, Ignjatovic, Motahari-Nezhad, Bertino, & Dustdar, 2013).

2.3 Aligning Workers to Jobs

Workers are aligned to jobs through several mechanisms. They may align themselves through the task selection process, providers of jobs can align workers through crowd recruitment strategies, and compensation models also incentivize workers towards platforms through a variety of different stimuli.

2.3.1 Task selection process. The configuration of the paradigm of crowdsourcing rests heavily on the open call model; in addition to this, micro-task crowdsourcing specifically implements task selection process. The task selection process is a passive recruitment strategy; it is identified as a core concern threatening the sustainability of worker pool populations, and ultimately the viability of systems that rely on the paradigm (Doan, Ramakrishnan & Halevy, 2011; Schulze, Krug & Schader, 2012; Quinn & Bederson. 2011). Using the open call, it first requires workers to subscribe to a labor market platform, access the platform and retrieve a list of tasks using some form of high level criteria for filtering. On retrieval of the filtered list of tasks, the worker opts to select a given task and peruse the low-level details and requirements which include but are not limited to allotted time, quality standards and criteria for completion involved with performing the given task. The worker then can accept the selected task or seek a new one. Provided the worker accepts the task, the task can be worked on and submitted to the employer for evaluation, feedback and subsequent compensation if applicable. Moving forward, the worker may opt to work on additional available instances of the same task, other tasks or leave the labor market platform (Schulze, Krug, & Schader, 2012).

2.3.2 Crowd recruitment. Crowdsourcing provides a range of benefits to organizations. Outsourcing traditionally entails the soliciting of services from a specific service provider with possible contractual lock-in with vendors; in contrast, crowdsourcing circumvents vendor lock-in via the open call over the Internet. It provides the opportunity to engage a diverse worker pool with demographics spanning gender, age, cultures, geographical borders, education, etc. With a more open selection in vendors, organizations have the flexibility to accept or reject submissions in accordance with an established standard in quality. Accepted submissions are compensated as per contractual agreements while risks, costs and expended effort associated with failure in rejected submissions have been minimized or averted. Despite these benefits, crowdsourcing introduces environmental configuration risks of its own. Organizations leveraging labor markets for soliciting workers are subject to lock-in to the crowdsourcing platform, its evolution, policies and strategic direction of the owners of the platform (Schenk, & Guittard, 2009).

The primary challenge for crowdsourcing has been identified as the attraction and the retention of the crowd (Doan, Ramakrishnan, & Halevy, 2011; Quinn, & Bederson, 2011). Various platforms employ different methods of compensation to attract workers. The most common types include but are not limited to entertainment, access to information, volunteerism, altruism and attention from others and financial incentives. The most popular type of compensation is financial and may be most evident with the advent of labor market platforms (Horton & Chilton, 2010).

Other approaches such as Internet advertising has been used to attract unpaid workers. Clickable ads are used to solicit interests and allow for redirection upon being clicked. It is believed that this approach provides a niche audience that may translate to individuals being valuable contributors for specific tasks. In addition to rendering a niche audience, the approach resulted in lower costs, wider diversity and a specialized worker pool with competent individuals when compared to paid crowdsourcing approaches (Ipeirotis & Gabrilovich, 2014).

Content creation traditionally originated from small groups of people or organizations for the consumption by large audiences. Enabling technologies such as Web 2.0 and the Internet have changed the ways in which content is created and consumed. Platforms facilitating informally created content receives high levels of attention, often rivaling traditional, controlled and formal sources. This is evidenced by high traffic to sites such as YouTube and Wikipedia accounting for considerable percentages of traffic on the Internet; in 2007, YouTube alone accounting for approximately 20% of HTTP traffic and 10% of all Internet traffic (Huberman, Romero, & Wu, 2009). The phenomenon, the “tragedy of the commons” has been a driving factor in informal content creation platforms (Huberman, Romero, & Wu, 2009). In the absence of a central regulator, this is where a group of individuals provide some common good. For the cases of informal content platforms, the common good is freely accessible information that serves all members of the community without being subject to degradation through usage. Some members are motivated by the attention their contributions receive through website hits, likes, downloads or their contributions becoming viral or trending topics. This attention in many cases satisfies the contributing member and is enough to forego monetary compensation. In contrast to this, it is found that members' productivity fall when their respective contributions fail to receive attention (Huberman, Romero, & Wu, 2009).

2.3.3 Compensation models. Commercial returns have far exceeded compensation for input from the consumer while there exist instances of gross under compensation; some instances involve no compensation which has maximum returns to the organization but has led to the exploitation of the consumer. Some instances can be noted in reviews of products, services, movies and seller reliability online (Kleemann, Voß & Rieder, 2008). Under compensation is also evident where the working consumer is not afforded a satisfactory working environment and conditions by their employers when compared to their traditional working counterparts (Kleemann, Voß & Rieder, 2008). In these situations, suitable working environment conditions, if desired, come at the expense of the consumer.

Non-cash forms of compensation usually include psychological compensation and have been difficult to quantify and establish equivalences to the satisfaction of workers (Scekic, Truong, & Dustdar, 2013) especially for tasks seen as laborious (Horton & Chilton, 2010). Non-cash incentives tend to be biased to the benefit of the owners of tasks and becomes unattractive to workers. It is also difficult to adjust the level of incentive such as entertainment proportionately to work done. Despite other forms of compensation used for attractive recruitment, financial compensation has been the most popular. Financial compensation is formatted using several payment models, these include pay-for-performance (PPP), quota systems and discretionary bonuses, deferred compensation, relative evaluation, promotion, team-based compensation and psychological incentives. In the simplest model, the PPP, the compensation is directly proportional to the work through quantitative evaluation. Quota systems utilize a performance threshold and allow for extra compensation upon surpassing the established threshold. With the deferred compensation model, workers are promised payment after work has been evaluated and has met quality standards. Relative evaluation allows peer evaluation where co-workers within a specified group evaluate fellow workers. Using predefined prizes, promotional models catalyzes competition amongst workers. Collective evaluation and compensation are payable to teams when work and effort cannot be easily delineated and attributed to individuals (Scekic, Truong, & Dustdar, 2013).

2.4 Selecting Workers for Specific Tasks

Crowdsourcing provides a range of benefits to organizations. Outsourcing traditionally entails the soliciting of services from a specific service provider with possible contractual lock-in with vendors; in contrast, crowdsourcing circumvents vendor lock-in via the open call over the Internet. It provides the opportunity to engage a diverse worker pool with demographics spanning gender, age, cultures, geographical borders, education, etc. With a more open selection in vendors, organizations have the flexibility to accept or reject submissions in accordance with an established standard in quality. Accepted submissions are compensated as per contractual agreements while risks, costs and expended effort associated with failure in rejected submissions have been minimized or averted (Schenk, & Guittard, 2009).

Despite circumventing traditional outsourcing risks, crowdsourcing introduces environmental configuration risks of its own. Organizations leveraging labor markets for soliciting workers are subject to lock-in to the crowdsourcing platform, its evolution, policies and strategic direction of the owners of the platform (Schenk, & Guittard, 2009). In contrast to valuable contributions of the working consumer, there are instances where they affect the quality in service delivery. Their competencies are either non-existent or below expected standards when compared to their traditional counterparts employed to a firm which affects their performance and ultimately the overall quality in the final product. As such organizations employ different strategies for monitoring, management and regulation to maintain quality standards in contributions throughout the workflow (Kleemann, Voß & Rieder, 2008).

2.4.1 Optimizing tradeoffs. Crowdsourcing presents several opportunities and challenges to optimize costs associated with recruitment, observers and equipment. Some approaches are more successful than others in the timely and economic acquisition of workers and their input to have meaningful developmental impact. As such, practices have been employed accepting compromise and accepting trade-offs (Kittur, Chi, & Suh, 2008). It is also accepted, that with the increased benefit in scalability of the crowd that results in a flexible on-demand workforce, there is consequential reduction in quality control in some tasks (Satzger, Psaier, Schall, & Dustdar, 2013).

The traditional research through surveys and other like instruments have reaped benefits through crowdsourcing. Many that have been limited to convenient sampling within universities have now expanded their participant pools through crowdsourcing approaches. Using online labor markets to engage respondents, Behrend et al. (2011) engaged an older, more ethnically diverse crowd with more work experience than the typical university only setting. The quality and robustness of the data was also found to be greater than or equal to the university only data.

Heer and Bostock (2010) experiment with popular crowdsourcing platform Amazon Mechanical Turk (AMT), to assess its feasibility in evaluating visualizations. The experiment indicated that crowdsourcing was a more economical option to traditional recruitment of human participants. Reduction in expenses were attributed to automated administration of the experiment through the labor market platform and the lower compensation to participants. Although there were cost savings, there were time differences with respect to completing a task. It was found that crowdsourced tasks required an average of 42 seconds while the corresponding face-to-face tasks in a controlled lab environment required on average of 5 seconds. Despite the differences in completion times for tasks in both environments, the crowdsourced tasks were all completed during a day as opposed to weeks for the traditional setting; this was due to additional time and effort required for recruitment and other scheduling issues for facilitating face-to-face contact. The experiment also attributed access of a more diverse population to crowdsourcing which was equally able to preserve the quality in results and mandate qualifications for jobs.

To capitalize on crowdsourcing provisions with optimization, Boer and Bernstein (2016) proposed a process engineering package "People Lib (PPLib)". It is a programmable repository using crowd process patterns and fragments to assist with the automation and optimization of workflows. For a specific problem, it analyzes a problem and its current workflow. Utilizing an exhaustive search of candidate processes, the framework chooses an optimal process to be implemented.

2.4.2 Managing expertise and skill levels. By opting to use crowdsourcing platforms, task owners open themselves to new challenges that include the filtering and selection of qualified and unqualified workers (Satzger, Psaier, Schall & Dustdar, 2013), the determination of quality standards and completeness in task submissions which leads to their subsequent acceptance or rejection. To address this problem, crowdsourcing platforms require rigorous and robust infrastructure with quality control facilities to mitigate and counter errors (Kulkarni, Gutheim, Narula, Rolnitzky, Parikh & Hartmann, 2012); this is extremely valuable when there is speculation about general reliability of the labor force.

Several approaches have been taken to implement and bolster quality control. Gold standard tests, carefully designed questions that identify workers with malicious intent, were used to prevent detect and prevent the task advancement of such individuals, subsequently omitting their submissions (Gadiraju, Kawase, & Dietze, 2014).

Iterative refinement of submissions has been used to improve non-expert language translations. Translations were obtained from non-expert speakers and redundantly edited. Scores were computed and assigned to their corresponding translations; a metric indicative of the translator's competence. Machine-learning techniques were then used to select the optimal translation from scores. The experiment proved that low-cost translations could be obtained from non-expert speakers that were comparable to professional translators (Zaidan & Callison-Burch, 2011).

Inferences were drawn from Bayesian predictive models to show the strengths of humans in a crowd; they were used to influence crowd recruitment. The inferences were obtained through probabilistic models that predicted how humans behaved given data from both contributions requiring human intelligence and a machine vision component (Kamar, Hacker, & Horvitz, 2012).

Trust-relations and other link properties were used to make estimates on the competencies of workers in hubs and other online web environments. This method was used for systems involving mixed workers through Human-Provided Services (HPS) and Software-Based Services (SPS) (Schall, 2011; Schall, & Skopik, 2010).

Table 1
Crowdsourcing Challenges and Approaches

Challenges	**Solution Approach**
Recruitment	Entertainment Altruism Volunteerism Attention Access to Information Monetary Incentives
Compensation Models	Pay for Performance Quota Systems Discretionary Bonus

	Deferred Compensation Relative Evaluation Promotion Team-Based Compensation Psychological
Optimization and Trade-Offs	Cost Time to Recruit Time to Complete Tasks Number of Participants Automation Population Diversity
Managing Expertise and Skills	AI Recommenders Probabilistic Models Solution Filtering Redundant Editing

2.4.3 Job recommendation strategies. In managing skills and expertise, recommendation approaches have been used to match workers with jobs; approaches used both externally and internally to the crowdsourcing paradigm. External to the paradigm, the person-job fit model is one such model used for this application; it evaluates workers' skillsets against the skillsets required to perform the job successfully. It also implements a bi-directional evaluation where the suitability of the job is measured against the workers' goals, interests and values (Schulze, Krug, & Schader, 2012). Using the, person-job fit model, a CV-recommender was used to leverage electronic repositories of CV's to match workers and jobs and other workers for collaboration. This resulted in an increase in sales in e-commerce platforms (Keim, T. 2007; Malinowski, Keim, Wendt, & Weitzel, 2006). Using work history, experience and other demographics in Internet based worker profiles, job transitions and the terminal work organization were also predicted using a machine-interpreted recommender (Paparrizos, Cambazoglu, & Gionis, 2011). Approaches in the aforementioned (Schulze et al., 2012; Keim, 2007; Malinowski et al., 2006; Paparrizos et al., 2011), made recommendations based on skills, and characteristics provided in their worker professed profiles. It is assumed that given the presence of such elements in profiles, workers' performance will yield quality submissions.

Using probabilistic models on worker's historical performance, task recommendation frameworks have been used to recommend jobs that are in line with the workers' preference; this increases the likelihood of workers accepting the task. It also reduced time workers would consume to seek and filter through jobs and increased the time that workers could focus on working on tasks (Yuen, King, & Leung, 2011).

Using crowdsourced, social network data, social network analysis was applied to evaluate friend-of-a-friend associations. A system "StakeSource", was used to identify stakeholders within a software project given their associations with existing stakeholders with the objective to minimize the omission of the input of vital stakeholders who could provide useful requirements during elicitation (Lim, Quercia, & Finkelstein, 2010). Pick-a-Crowd (Difallah, Demartini, & Cudré-Mauroux, 2013), is another system using crowdsourced, social network networking profile data to construct worker profiles. Interests found in the social network profiles are cross-referenced using semantic technologies with terms in the Linked Open Data Cloud. Upon finding term equivalences, worker profiles are established and evaluated against task descriptions using text, category or graph based techniques. Finally, using some form of push method, tasks are assigned to workers.

2.5 Mixed Crowdsourcing Systems

With continued success in crowdsourcing, other special interest areas such as virtualization and cloud computing, have sought the benefits of human computation merging the two to create a mixed computational approach with both machines and humans. Dustdar and Bhattacharya (2011) asserted that neither software nor humans exclusively, can tackle complex tasks within a single conceptual framework. Machines are better suited for rapid task computation and the results thereafter can be proofed leveraging human expertise (Dustdar & Bhattacharya, 2011; Riveni, Truong, & Dustdar, n.d.). In some problem domains, modern systems utilize both machine and human workers to perform tasks (Vukovic, 2009). Consequently, this combined effort combined with the infrastructure of virtualization and cloud computing approaches requires the potential for dynamic scalability and proactive provisioning of human and machine working units in response to costs, magnitude of the task, complexity, time constraints among other criteria (Candra, Zabolotnyi, Truong, & Dustdar, 2014; Dustdar, & Truong, 2012).

Crowdsourcing is well suited for this mixed approach using as the engine for human computation to its machine counterpart. Human-in-the-loop models are utilized where humans perform designated role at a specific phase of a given workflow; tasks may span descriptions including data collection, computation and verification (Dustdar & Truong, 2012).

Implementations using humans as the primary mechanism for the collection of data are easily facilitated through crowdsourcing. Typically, the data is knowingly or voluntarily offered and may but not necessarily involve some type of compensation. Other systems use crowdsourced data from existing systems like social media (Barbier, Zafarani, Gao, Fung, & Liu, 2011); this form of data is not necessarily obtained directly from the creator of the data, however through their association with an existing system. The data upon request can be provided or sold to the requesting party. When crowdsourced data used, it may not be viewed as a mixed crowdsourcing process; the results however still reflect a mixed approach with human and machine inputs and processing. Enabled by the Internet, Web 2.0 crowdsourcing platforms and smart devices with inexpensive sensor capabilities (Chatzimilioudis, Konstantinidis, Laoudias, & Zeinalipour-Yazti, 2012), increasing opportunities emerge to readily engage massive crowds for data input at very low costs if any at all (Vukovic, 2009).

Systems engaging human intelligence for computation and verification assign roles and responsibilities to humans to perform tasks that machines, algorithms and modern technology are either incapable of handling, handle inefficiently or ineffectively. The role can either be for phases requiring human preprocessing producing input for a future phase or as processing to produce a final output. In roles requiring verification, humans assess information produced by their machine counterparts before it is released as final output or used as input in a latter phase of the workflow (Dustdar & Truong, 2012).

It has been established that systems use mixed approaches engaging human computation via crowdsourcing combined with traditional machine computing elements. Subsets of these systems use the computing components data collection or input driven mechanisms while others use them for computation. Moving forward, they will be distinctly referred to as data-driven and computational elasticity.

2.6 Data Driven Elasticity

Data required for workflows can be directly solicited through crowdsourcing using the open call or indirectly using crowdsourced data obtained from open sources or through requests from closed controlled sources. Both cases offer data to be used in systems (Gao, Barbier, & Goolsby, 2011; Park, Parameswaran, & Widom, 2012); this data, enables the same systems to fulfill their mandates and function efficiently. Given this reality, human efforts are naturally focused at the beginning to assist in maintaining the sustainability of these systems through data input as there are non-existing to minimal roles of computational responsibility. In this section, we refer to such human participation as data driven elasticity.

Data driven elasticity applications have found favor with disaster relief organizations in the face of unfortunate events. Relief organizations used information from free SMS texts amongst survivors to create crisis maps across affected regions in the aftermath of the 2011 earthquake in Haiti; the information was shared amongst multiple organizations to coordinate relief missions to target and timely effect specific needs (Gao, Barbier, & Goolsby, 2011). Other success stories with disaster response include the 2007 to 2009 wildfires in Santa Barbara where volunteered geographic information (VGI) was used. Despite questions in the quality of the data, origins and collection processes of the data, VGI served as a quick and insightful resource for mapping agencies and organizations to plan disaster relief efforts (GoodChild & Glennon, 2010). The 5W model has also received success in the detection of urban emergency events using social media. Answers for instances of urban emergency are obtained from the questions modeled with "what, where, when, who and why" constructs for event specifics (Xu, Liu, Yen, Mei, Luo, Wei, & Hu, 2016).

Crowdsourced data has played a part in digital forensics through the provision of evidence. Scenes of the Boston Marathon were reconstructed from data from the social network participants in the aftermath of unfortunate events of bombing which aided in the identification and subsequent apprehension of the perpetrators (Tan, Blake & Saleh, 2013). Crowdsourced data has also made its mark in customer relations and marketing across multiple industries, this includes but is not limited to automotive, airline, hotels, food and beverage industries. Major corporations spanning industries rely on posts from Facebook and tweets from Twitter, to discover customer experiences and impressions of their brands. Through insight gained from customer impulses, there are increased opportunities in making fast corrective decisions and corporate responses as opposed to hypothesizing after months of damage (Wong, 2012).

Local authorities and other entities have found value in crowdsourcing to support civic needs. It is used to mobilize citizens and leverage their input to increase their engagement in public affairs. One such success story lies with Urban planning and public projects; through a web based system, crowdsourcing has led to far more citizen input participation than face-to-face meetings (Brabham, 2009). Sensor rich smartphones have enabled citizens to contribute to civic causes through crowd-sensing. Using NoiseTube, citizens used their low-cost sensors on their smartphones to detect and collect data on urban noise pollution. This allowed the citizen to measure their personal exposure to noise and cumulatively, all data was used to create noise maps across communities (Stevens & D'Hondt, 2010). Other approaches of crowd-sensing have responded by experimenting with the idea of a ubiquitous sensing platform to leverage low cost sensors of smart devices. Using Twitter, the sensing platform was used create noise and weather radars using the smart devices of the users in the crowd. Results of this experiment were promising revealing the feasibility, opportunities and challenges for implementing such systems (Yan, Marzilli, Holmes, Ganesan, & Corner, 2009). Another success story mCrowd, used crowdsourcing platform AMT and ChaCha via a proxy to connect an iOS based application that allowed users in the crowd to perform crowd sensing with their iPhones. Tasks include image tagging, offering geo-location and road traffic monitoring (Yan, Marzilli, Holmes, Ganesan, & Corner, 2009).

Citizen science projects have gained success through data-driven elastic crowdsourcing. They foster communities of ecologists, biologists and environmentalists in meeting their individual and collective mandates. Such success is reflected in eBird, a massive container of information on the population density and distribution of birds across geographic regions measured on temporal scales. The data is sourced and provided by a diverse user base of environmental advocates, recreationalists, land managers, biologists, ornithologists and professional bird watchers (Sullivan, Aycrigg, Barry, Bonney, Bruns, Cooper, Damoulas, Dhondt, Dietterich, Farnsworth, & Fink, 2014).

2.7 Computational Elasticity

Many workflows have humans directly integrated to provide processes with some human intelligence, computation and intuition. These systems require humans to perform a designated role or task that includes computation or verification of some output at some stage. In this section, these systems combining human and machine effort in this fashion are classified as computational elastic effort.

CrowdDB uses human computation to support the servicing complex queries that are often challenging to be served by traditional database systems and search engines. Humans are used to augment information missing in the database, perform complex computations and functions and data manipulation with respect to results from vague search criteria (Franklin, Kossmann, Kraska, Ramesh, & Xin, 2011).

Crowdsourcing has been used to facilitate collaboration of geographically dispersed citizens who are engineers. Engineers vary in their levels of expertise and credentials ranging from students, to researchers and industry practitioners; they are engaged to work on real world problems through the Internet through crowdsourcing platform “Citizen Engineering” (Zhai, Sempolinski, Thain, Madey, Wei, & Kareem, 2011).

Combining the translations of human and machine workers, Active Crowd Translation uses a mixed approach with a human in the loop implementation to translate documents from one language to another. Human workers come from the crowd and consists of non-professional translators that provide translations. Using a proposed model, the system selects the best translation from those provided by humans (Ambati, Vogel, & Carbonell, 2010).

Crowdsourcing has proved as extremely powerful and feasible in the analysis of satellite imagery when compared to machines. In the 2007 episode of Jim Gray, many people around the world voluntarily evaluated 560 000 images representing 3500 squared miles of open sea to find the missing computer scientist. Despite the unfortunate ending where Gray was not found, the event was a landmark in harnessing mass human computation and analysis (Doan, Ramakrishnan, & Halevy, 2011; Quinn, & Bederson, 2011).

The labeling of concepts and entities have been implemented through mixed elastic approaches. In Galaxy Zoo, humans through crowdsourcing and machines in the form of Bayesian predictive models, are used to classify celestial bodies. The experiment was aimed at investigating how these joint efforts could be used to solve a consensus task. A consensus task is one where the owner of the task believes a solution exists and can be obtained through the mass wisdom of crowds. Human and machine efforts were supported by probabilistic models that were used to predict human behavior (Kamar, Hacker, & Horvitz, 2012).

Crowdsourcing has been used as the computing engine for big data applications to support the parallel processing of data for tasks requiring human computation. Deficiencies were identified in inter-dependent tasks in general purpose micro-task labor markets. Through the combination of concepts from organizational behavior and distributed computing, tasks were decomposed, parallelized and mapped to workers in crowd through a MapReduce framework “CrowdForge”. The framework manages all dependencies amongst workers and enforces quality constrains. Workers receive one or more discrete subtasks for processing after decomposition with the results merged into a single output (Kittur, Smus, Khamkar & Kraut, 2011).

Peer grading and machine grading have been used to evaluate open-ended answers in assignments using the predictions and confidence levels of an algorithm. The evaluation suggests the number of human peers needed for continued evaluation of the answers. The answers are then further evaluated by human peers guided by a rubric; given this evaluation a feature set is developed. Another set of peers revise the correct application of the feature set (Kulkarni, Socher, Bernstein & Klemmer, 2014).

To support the dynamic scaling of human and machine components, the Vienna Elastic Computing Model (VieCom) was proposed. The system scales in response to dynamically changing runtime contexts impacting quality and costs. VieCom transforms crowdsourcing from an active to passive model to achieve task assignment in a timely manner. Active models embody the traditional open call where the crowd opts in to engage tasks that are posted. Passive models assign tasks to workers based on their posted profile qualifications (Candra, Zabolotnyi, Truong, & Dustdar, 2014). Both humans and machines are abstracted as computing units under a uniformed service model (Candra, Truong & Dustdar, 2013) awaiting incoming tasks (Candra et al., 2014).

To enhance IT service delivery in an enterprise, a web-based system PeopleCloud was proposed to manage and scale virtual teams of experts, tasks and provision services for tasks. Through the discovery of experts, the system aims at building the capacities in virtual teams using knowledge networks built from organizational and external human resources to perform complex tasks and tasks requiring knowledge transformation (Lopez, Vukovic, & Laredo, 2010).

Citizen science project, Wildlife@Home, is a digital surveillance platform allowing biologists to analyze recorded footage of animals in their natural habitat. Pre-recorded video is sourced from cameras in the wild, is retrieved and uploaded to the platform for viewing and analysis by its member community where they record instances of various events from footage. The system has earned its credibility as its results are found to be statistically comparable to that of experts; a notable study enabled by this platform monitored the effect of North Dakota's oil development on federally listed endangered species (Desell, Goehner, Andes, Eckroad, & Ellis-Felege, 2015).

2.8 Summary

Crowdsourcing is most prevalently configured upon the open call model. This recruitment strategy is passive and threatens the sustainability of workers available in labor pools; as such the attraction and retention of the crowd is a primary concern for the paradigm. Despite multiple stimuli that encourage recruitment, current strategies still subscribe to the open call.

> *In this work, the state-of-the-art open call systems is extended by developing a customized Open Push-Pull model that maintains core crowd management services. The Open Push-Pull model also addresses the issue of crowdsourcing platform lock-in by promoting a platform for integration to interface with existing crowd and provider platforms.*

Current systems integrate crowdsourcing as a strategy to facilitate human contributions to augment workflows with machine components. In these mixed crowdsourcing systems, human input can come in two forms, as a provider and as a processer of data. Data facilitated through human input, can be used as raw input to sustain and enable functionality of systems using machine components for processing. Computation facilitated through human intelligence serve as computing units within workflows that consist of phases where machines, current technology, and algorithms are inadequate. Humans computation can also serve as verifiers to intermediate data to be used in later phases of a workflow. Despite this integration of humans to collaborate with machines in mixed crowdsourcing systems, literature does not represent models where both humans and machines can compete for the same tasks.

> *As such, contributions here provide recommendation strategies in conjunction with the Open Push-Pull model to uniquely facilitate the delegation of work to* ***both types of workers, machines and humans interchangeably****.*

Job recommendation strategies and models present a top down approach. They evaluate the requirements of the job against the skillsets professed in resumes of potential working candidates. This approach leads to question the validity of information professed in resumes; if they are indeed accurate and representative of the true competencies of the candidate worker.

> *Contributions in this thesis create a bottom up approach where recommendations are computed based on historical performance of human* ***and*** *machines.*

Given that a worker had satisfactory performance for a specific job, it will be inferred that he or she possesses the requisite competencies to do jobs characteristically similar to those previously completed.

3.0 Technical Approach: A Worker-Job Framework for Mixed Crowdsourcing Systems

Gaps identified in related literature lead to four open concerns, including the recruitment and retention deficiency in the open call model, platform lock-in with crowdsourcing platforms, an unexplored model of competing work elements in mixed crowdsourcing systems and assumed accuracy in data used in top down job recommendation strategies. Succinctly, the first research question is reiterated below:

When managing crowd computing resource spanning human and machine workers, what general information models effectively define:

- functional, non-functional and evaluative concerns for both types of workers?
- A wide cross-section of tasks from diverse problem domains?

Through the proposal of a paradigm shift towards an Open Push-Pull model, Research Question 1 articulates deficiencies outlined in the open call model. This technical approach details an adaptive information model, required to support the Open Push-Pull model in allowing for the aggregation of data from diverse sources including providers of jobs and crowds of workers. It must also support both human and machine worker profiles allowing them to competitively solicit and perform the same tasks. The information model should support evaluative metrics that enable decision support for the recommendation and delegation of work to both types of workers. Research Question 2 is restated briefly as outlined below:

Is there an operational approach that enables systematic and reliable:

- Delegation of work across human and machine work resources?

- Recommendations as a function of changing environment data such as jobs, labor pool and performance history?
- Metrics that allow for the evaluation of collective capabilities of a worker pool?
- On-Demand modes of operation within a service driven infrastructure?

To address Research Question 2, the technical approach must be an operational strategy to enable the delegation of work to both machine and human workers. To optimize this delegation to most appropriate workers, the strategy should enable recommendations in response to evolving environment data to include availability of jobs, workers in the labor pool and job uptake and workers' performance history. To understand the worker pool, the strategy should include community metrics to evaluate the collective capabilities of all workers available for work. Finally, to support the vision of the Open Push-Pull model, the strategy should support on-demand modes of operation through a service-oriented architecture facilitating adaptive behavior to support data originating from diverse sources. This architecture should be able to scale gracefully as jobs are added, the worker pool increases, and as worker profiles expand in response to their respective uptake in jobs.

In this chapter, the overarching contribution of this work is introduced, a comprehensive worker-job framework for mixed crowdsourcing systems. It begins with the vision of the Open Push-Pull model at a high level of abstraction followed by detailed descriptions of components. The first component is a pattern inspired framework that facilitates adaptive behavior in a middleware. The middleware enables service delivery through a uniquely combined service-oriented architecture. The chapter continues with an adaptive information model that supports worker and job data from diverse sources within the Open Push-Pull model. Next, there is a system workflow detailing the procedure from gathering data through making recommendations. A computational model follows outlining all computational models used in making recommendations. Finally, an operational model encompassing an elastic workflow and configurable workflow algorithms is introduced. The chapter concludes with a metric model that is used to evaluate a labor force's self-perception against their actual performance.

3.1 Open Push-Pull Model

An Open Push-Pull model is proposed to address the challenges faced with the Open Call model while adopting strengths and other working configurations of traditional core crowdsourcing services. It also mitigates against lock-in to a specific labor platform by including them and integrating them into a more flexible and fluid model . As an enabler to this model, a service synchronization and coordination middleware (SSCM) sits at the center of the model to interface with, manage and coordinate with diverse external services (Figure 2).

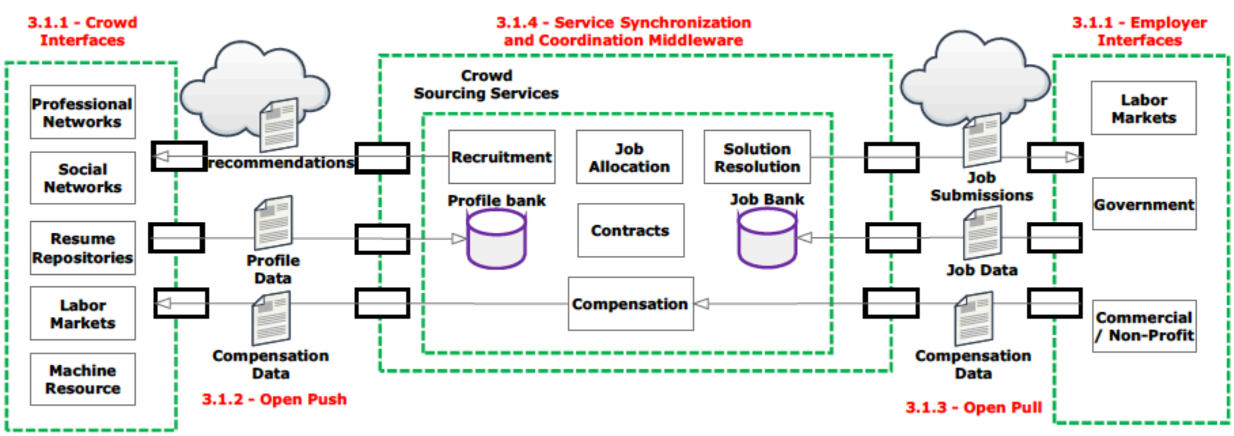

Figure 2. Illustration of the Open Push-Pull model for crowdsourcing showing diverse repositories and the service synchronization and coordination middleware.

3.1.1 Diverse repositories. In the architecture, worker data is driven from existing communities and repositories (Figure 2). These sources include professional networks (e.g. LinkedIn) and resume repositories (e.g. Monster.com and Indeed.com) consisting of CV's, worker skillsets, job experiences amongst other desirable professional information. They also include social networks (e.g. Google+, Facebook) where it is assumed that there are high possibilities that the same interests and credentials may be associated within social circles. Labor markets (e.g. Upwork, Amazon Mechanical Turk) are also a viable source where they have already built communities targeted for labor and are ready to accept new offerings for work.

Job data originates from various provider entities (Figure 2). Multiple employers from varying entities, including but not limited to labor markets, government agencies, commercial and non-profit organizations establish work and task orders that can be translated to crowdsourced jobs. The jobs consist of tasks across varying problem spaces and domains in diverse formats.

3.1.2 Open pull. Using crowd and provider interfaces, an open pull mechanism is employed to attract a crowd and build a labor pool (Figure 3). Existing e-platforms traditionally provide web API's to allow for programmable integration with an electronic data exchange between external entities. With these web API's, worker-based queries can be tailored to meet the needs of employers for the requisite jobs. Customization includes passing varying parameters with necessary hard and soft conditions outlined in job descriptions; as such a filtered list of potential candidates are obtained for specific instances or categories of tasks. Listing 1 outlines a WSDL file with potential types that can be exchanged with the SSCM in the exchange of worker and job data; this file is discussed in detail in a later section.

3.1.3 Open push. An open push mechanism is used in the retention of workers to maintain the capacity of the labor pool. Using recommenders grounded in machine learning techniques, jobs are recommended to members of the labor force via communication channels within the SSCM in conjunction with available web API's of crowd interfaces. Parameters for recommendations are defined by the owners of the jobs and can vary in regard to description, job complexity, skill qualifications and requirements (Figure 3).

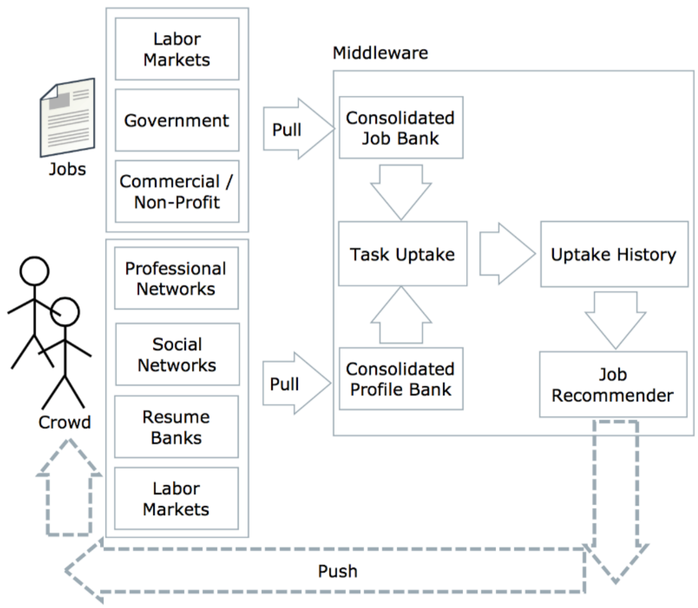

Figure 3. Workflow showing sequence of processes in the open pull-push architecture.

3.1.4 Service synchronization and coordination middleware. The Service Synchronization and Coordination Middleware (SSCM), is a layer designed to provide crowd management services (outlined in sections 2.3 & 2.4) found in state-of-the-art crowd sourcing platforms (Figure 1). These services include worker recruitment, compensation mechanisms, job allocation and contractual mechanisms and decision support for solution resolution for job submissions outlined in crowd management services in Section 2.

SSCM modules incorporate decision support mechanisms to enable worker-job match-making. Prospective mechanisms include case-based reasoning, semantics, collaborative filtering and other customized machine-learning approaches. Compensation can be negotiated on the grounds of the worker expertise, reputation, job complexity, and the urgency of job offerings. SSCM allocates whole jobs or sub-parts thereof to an available pool of workers including machines and humans. As the SSCM streams information, it faces several challenges and needs to adapt to needs of the stakeholders of the platform, constantly changing data (Table 2) and the management of the streams of data from repositories (Table 3).

Table 2
Service Synchronization and Coordination Service Component Challenges

Service Synchronization and Coordination (SSCM)	**SSCM Challenges**
Recruitment Management	Profile Matching, Collaborative Filtering, Case-Based Reasoning, Machine Learning
Compensation Processing	Multi-Dimensional Optimization
Job Allocation	Elasticity
Solution Resolution	Database Management, Stream Processing

Table 3
Repository Interface Component Challenges

Repository Interfaces (RI) Components	**RI Challenges**
Specialized Interfaces	Rapid Interface Extensibility
..	Flexible Communication, Publish/Subscribe, Unpredictable Alerts
..	Flexible Profile Data Management

3.2 Realizing Services through Patterns

To support the notion of elasticity, scalability and adaptability, this section outlines a unique system design built using a combination of architectural and design patterns (Figure 4). The system design embodies a major software design principle, the separation of concerns and achieves this through the fusion of two major software architectural patterns (Fowler, 2003); the Model-View-Controller (MVC) (Figure 4, parts 4a, 4b and 4c) (Fowler, 2003) and the N-Tier architecture (Figure 4, part 4c) (MSDN, 2017). The model of the MVC is abstracted under a uniformed web service interface (Figure 4, part 4c) which corresponds to the business logic (Figure 4, part 4.1), service (Figure 4, part 4.2) and domain (Figure 4, part 4.3) layers of the N-Tier. The MVC's view (Figure 4, part 3a) and controller (Figure 4, part 4b) sits above this interface corresponding to the presentation layer (Figure 4, part 4.0) of the N-Tier.

Most patterns are housed and coordinated under a web service driven interface exposed to and invocable by diverse types of clients. The first pattern lies within N-Tier's business layer (Figure 4, part 4.1) and consists of a generic layer super type manager (Figure 4, part 4.1a) providing service loading related mechanisms that cross cuts across all specific managers (Figure 4, part 3.1b). Specific managers contain workflow logic to handle their corresponding domain data.

The service layer (Figure 4, part 4.2) of the N-Tier is driven by a strategy pattern (Figure 4, part 4.2a). It consists of a decoupled, abstract singleton factory which facilitates inversion of control, capable of dynamically loading services for diverse providers of workers and jobs. The factory is capable of hot swapping these services, polymorphically loading appropriate services depending on a context. This contextual loading of services is possible using 3 additional patterns, the super interface, separated interface and plugins (Figure 4, part 4.2c). The super interface (Figure 4, part 4.2b) serves as a marker interface and parent to job and worker interfaces. The separated interfaces for job and worker services define the behavioral contract for the underlying and corresponding plug-ins. Plug-ins include API logic and libraries to connect to external services including crowd and provider interfaces.

The domain layer (Figure 4, part 4.3) consists of value objects (Figure 4, part 4.3a) that meet the data storage requirements of the platform. Domain objects are sequentially passed as messages through the tiers of the N-Tier. Layers directly interact their neighboring layers; upper layers make requests of lower layers and lower layers serve the requests of the upper layers (Figure 5).

Domain transport objects (DTO's) are used to export and import data between the web service and the connecting clients. They allow for easy serialization of data to an intermediate language like XML or JSON than value objects as they contain simple constructs and common data types pervasive across a variety of languages. For this implementation, DTO's are used to support interoperability in exporting data to and receiving messages from entities outside the service sandbox. Six primary types are defined using WSDL file namespaces, these include employer, skill, job, compensation, category and worker as outlined in Listing 1.

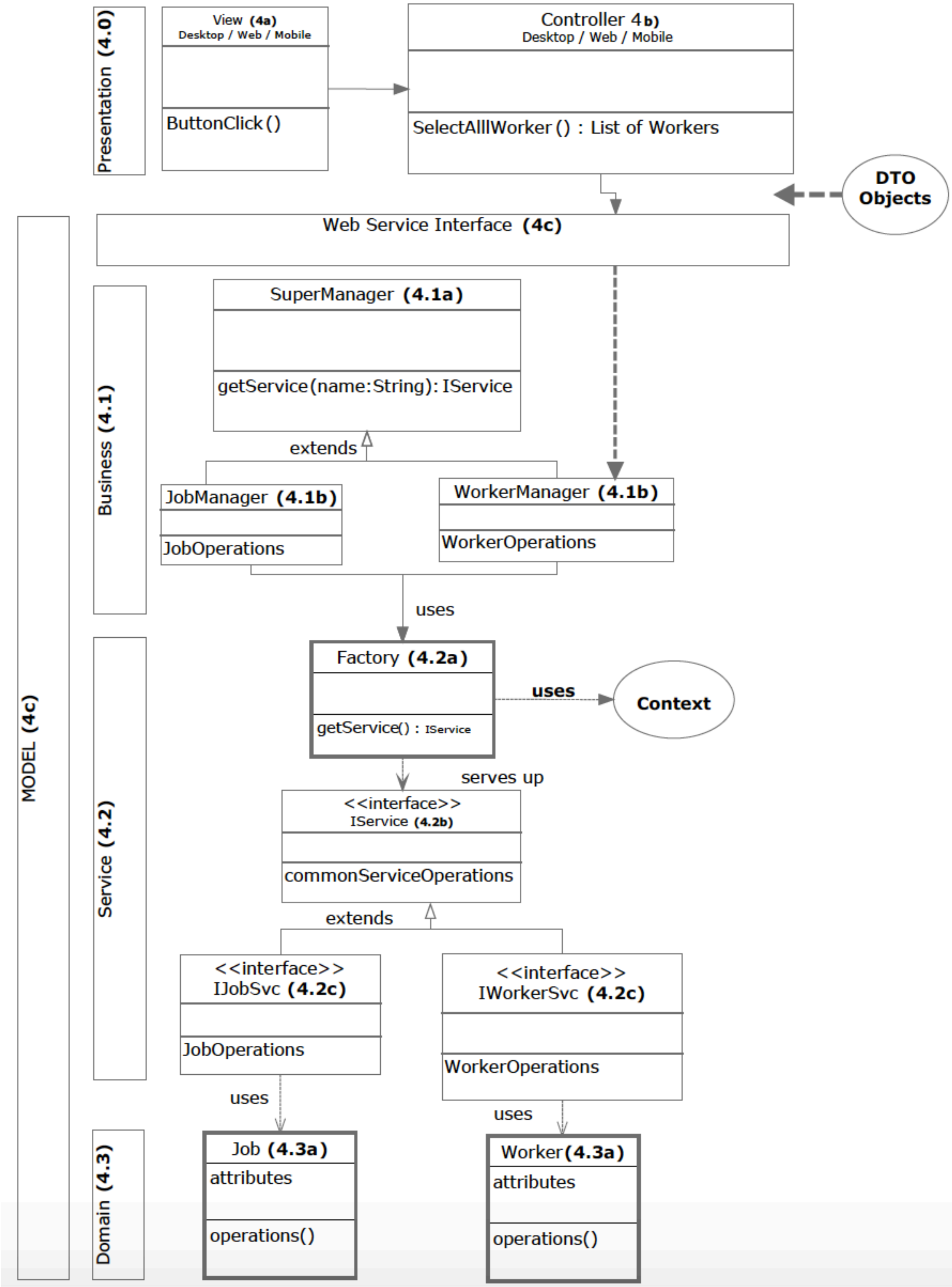

Figure 4. Pattern-Oriented Design showing Service-Oriented Architecture for the SSCM.

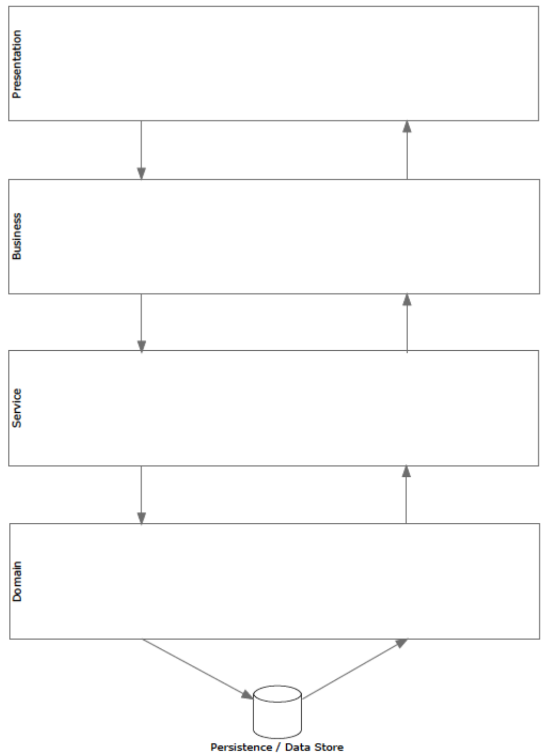

Figure 5. N-Tier architecture showing movement of data and workflow of messaging between consumption and service layers.

```
<xs:complexType name="SkillDto">
<xs:sequence>
<xs:element name="SkillDescription" type="xs:string"/>
<xs:element name="SkillId" type="xs:string"/>
<xs:element name="SkillName" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="JobDto">
<xs:sequence>
<xs:element name="CompensationId" type="xs:string"/>
<xs:element name="EmployerId" type="ser:guid"/>
<xs:element name="JobCompensationValue" type="xs:double"/>
<xs:element name="JobDescription" type="xs:string"/>
<xs:element name="JobExperienceLevel" type="xs:double"/>
<xs:element name="JobId" type="ser:guid"/>
<xs:element name="JobName" type="xs:string"/>
<xs:element name="JobQuota" type="xs:int"/>
</xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="CompensationDto">
<xs:sequence>
<xs:element name="CompensationDescription" type="xs:string"/>
<xs:element name="CompensationId" type="xs:string"/>
<xs:element name="CompensationType" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="CategoryDto">
<xs:sequence>
<xs:element name="CategoryDescription" type="xs:string"/>
<xs:element name="CategoryId" type="xs:string"/>
<xs:element name="CategoryName" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="EmployerDto">
<xs:sequence>
<xs:element name="EmployerId" type="ser:guid"/>
<xs:element name="EmployerName" type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="WorkerDto">
<xs:sequence>
<xs:element name="WorkerId" type="ser:guid"/>
<xs:element name="Email"  type="xs:string"/>
<xs:element name="FirstName"  type="xs:string"/>
<xs:element name="LastName"  type="xs:string"/>
<xs:element name="Gender"  type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

Listing 1. Formal definition of SSCM compatible types of data supporting data exchange and interoperability.

3.3 Information Models

This section outlines the conceptual models used to support job data from diverse sources, machine and human workers within a uniformed crowdsourced ecosystem. It presents two models in the form of abstract data types for modeling both tasks and workers. The process for developing the ADT's was inspired by the COMET method for designing concurrent, distributed and real-time applications (Gomaa, 2001). The method uses object-oriented concepts to justify software design utilizing UML as the language to express the design. The method emphasizes object structuring in which it outlines several criteria for a good design. These include objects that can be entities, have relationships between others, provide interfaces, support control and application logic. Architecturally, COMET also promotes sub-system structuring, configurable components, division of responsibility between distributed components such as clients and servers, message communication interfaces especially for decisions made in concurrent tasks and decisions requiring context. Finally, it promotes concurrent real-time assessment of the entire architecture in meeting performance targets.

3.3.1 Modeling Tasks as ADTs. A challenge in this approach is representing tasks from different problem domains, bearing different characteristics and originating from diverse sources. To address this problem, this work proposes a customized task abstract data type (ADT) (Figure 6). Each task is modeled with an identification number, name, text description and a set of decision support metrics. These metrics include the task complexity index (TCI), elastic index (EI) and domain relevant threshold for elasticity (EI_threshold). TCI and EI scores are calculated from the weights gathered from the corresponding metrics. Each decision support metric is detailed later in this chapter.

3.3.2 Modeling Workers as ADTs. Likewise, workers are modeled as ADT's with the proposition of the worker abstract data type (Figure 6). This provides a uniformed abstraction of the worker from their actual form, whether human or machine. The ADT facilitates both types to be evaluated equally as potential candidates for the assignment of jobs. The rating of a worker is calculated as a cumulative measure of all employer feedback, stored as performance evaluations. In contrast, ranking is an indicator of a worker's rating relative to other workers being tracked in the system. Workers have multiple competencies modeled as skills; though optional, their respective levels of mastery / expertise can be provided. Concrete worker classes are indicative of the discrete types of workers and their characteristics are not exhaustive; it is rather suggestive to detail other demographics of the worker.

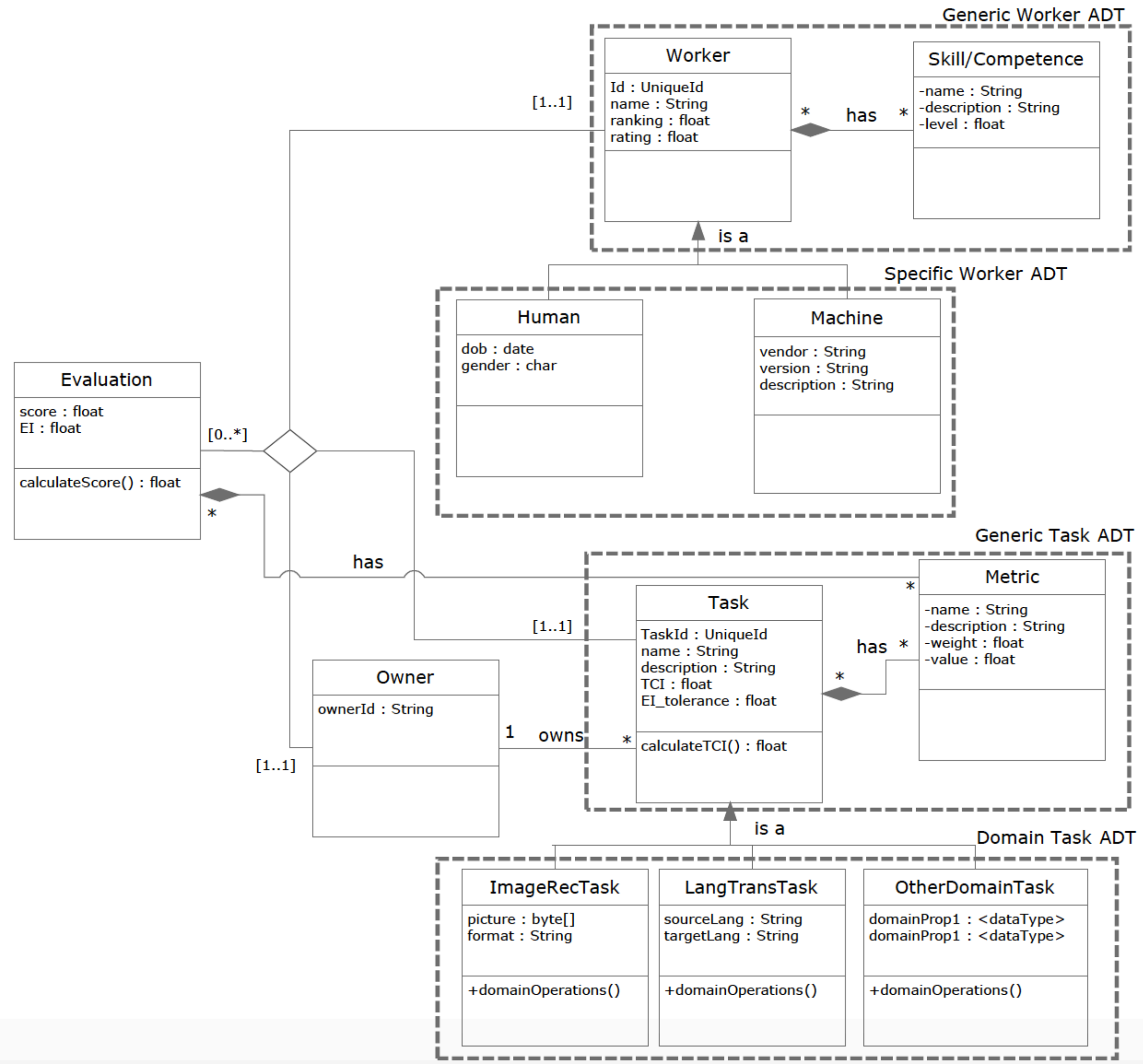

Figure 6. Conceptual model showing task and worker abstract data types along with evaluative data stores used to support employer feedback.

3.4 System Workflow and Adjustments

In this section, the workflow of the worker-job recommender engine of the SSCM is outlined and detailed in five subsections. Each section discusses the inner workings of stages illustrated in Figure 7 which corresponds to processes outlined in Listing 2 for respective engine modules.

3.4.1 Data Gathering. The workflow is initiated by an open pull from diverse crowd and employer sources via available web services APIs from crowd and employee interfaces. These interfaces provide invocable parameterized remote procedure query calls capable of returning data to our platform. Queries are customizable to dynamic requirements of the employers and iteratively refined recommendations as data evolves. Listing 1 outlined compatible WSDL exchange definitions DTO's that are used to support the import of such external data.

3.4.2 Data Transformation. Prior to recommendation, all data must be transformed into the required matrix form through data preprocessing provided through this module. Textual and categorical information are converted to discrete or numerical values. There are 5 major 2-D matrices that serve as input to the recommender; these are matrices U, J, X, Y and R. Matrix U consists of 2 columns, with column 1 representing the worker's unique ID and column 2, their self-professed skill level (this value is optional); each row in the matrix represents a worker. Matrix J consists of a vector of job ID's; each job corresponds to a row in Matrix X which represents the feature set of the job. The features are indicated as either present or absent using binary indicators. Matrix Y are the scores that users received for a job; each row in Y represents a user in the corresponding row in Matrix U. Jobs are represented by columns; column numbers in Y represent corresponding row numbers in matrix J containing the vector of jobs. Jobs too new to have ratings due to the absence of work history, and missing feedback due to employer negligence, are valued at 0. Matrix R is equivalent to matrix Y with the same corresponding row and column properties to matrices U and J respectively; R contains a binary indicator that the engine uses to determine which ratings should be considered during the calculation of recommendations. Listing 3 outlines a small example of the matrices and their contents.

```
Workflow Worker-Job Recommender
Start
  Data{Jobs,WorkerProfiles}=OpenPull(WebServiceCalls)  // 3.4.1 Data Gathering
  Matrices{X, Y, R, J, U} = Transform(Data)   // 3.4.2 Data Transformation

// 3.4.2 Begin Recommendation
  Theta{# of users} = randomNormDist();
  Theta{# of users} = OptimizeWeights{X, Theta}
  For I = 1 to 100
      Theta{# of users} = CollabFiltOptimizer(Matrices)
  End For
  meanY = mean(Y, R)
  Predictions{# users, # jobs} = X * Theta^T

  If Predictions(U{specific worker}) contains zero
      Predictions(U{specific worker}) + = meanY
 End If
//End Recommendation

  Reports{community, users} = Analysis(Matrices, Predictions)
Stop
```

Listing 2. Psuedo-code definition of the workflow of SSCM operations involved in the recommendation process.

U

A1	9
B1	5
C1	8
D1	6

J

J1
J2
J3
J4

X

0	0	1
0	1	0
0	1	1
1	0	0

Y

5	2	2	1
4	2	5	0
6	9	1	10

0	7	0	9

R

1	1	1	1
1	1	1	0
1	1	1	1
0	1	0	1

Listing 3. Example of Matrices produced after data transformation and their contents.

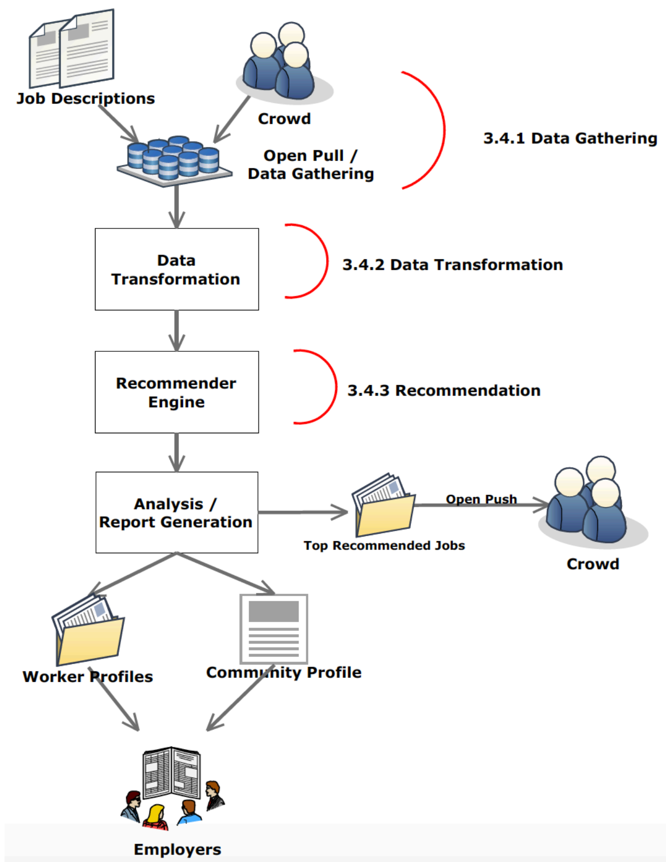

Figure 7. Workflow showing SSCM modules and operations involved in the recommendation process.

3.4.3 Recommendation. Using collaborative filtering techniques outlined in (Section 3.5), the recommender engine of the SSCM is configured to predict a rating for each job per worker given the current collaborative employer ratings of all users against the current worker's performance on jobs in its work history. Despite having calculated a recommendation for all jobs, the engine is designed to recommend the top N ≤ 10 jobs with a designated similarity to jobs already performed by the current worker using a distance function (Section 3.5.4). Recommendations are also made to workers with no prior history via the use of a mean normalization technique (Section 3.5.4).

3.5 Computation Model

3.5.1 Task Complexity Index. The task complexity index (TCI) is a weighted quantitative metric designed to describe the difficulty or complexity for a specific task. It is formally defined as the weighted average of available metrics that conforms to a context-defined nominal scale. Weights can be manually fixed by domain experts or adjusted over time through interpretive techniques. Eq. 1 outlines the TCI, where $W_1 \ldots W_x$ are weights assigned to their corresponding metrics $M_1 \ldots M_x$; subscript x denotes the index of the weighted metric pairs. W_i are all positive integers including 0 while M_i are positive values on a scale 1 to N where N is the maximum value for the said scale.

$$TCI = \frac{W_1 \times M_1 + \ldots + W_x \times M_x}{x} \qquad (1)$$

where

- $x \in Z^+$ and $x \geq 1$.
- $W_i \in Z^+$ and $W_i \geq 0$, $i \in 0, 1, \ldots, x$.
- $M_i \in [1, N]$, where
  - ⋆ $N \in Z^+$ and $N \geq 1$.
  - ⋆ $i \in 0, 1, \ldots, x$.

3.5.2 Gradient Descent / Linear Regression Models. Leveraging the principle of supervised learning, gradient descent and linear regression (least squares) models are used to provide multi-variate analysis of features against known outcomes to make predictions for new / unknown instances. The algorithm iteratively chooses the steepest direction towards some local minimum and takes steps towards convergence. These minima differ on the number of features in the dataset and the random starting point used to begin the descent.

At the center of the gradient descent algorithm are two key steps; the first step includes a minimization / optimization step (Eq. 2), and second, the gradient step achieves convergence (Eq. 3). At the minimization step, the algorithm minimizes cost producing a vector of weights corresponding to the worker's competence for skills in the available global feature set used to characterize all jobs. The gradient step is iteratively performed for a fixed number of iterations and is used to update the precision of the cost and ultimately the produced weights. Regularization is applied to both the cost and gradient step functions to prevent overfitting in the model (Ng, n.d.).

$$J(\theta) = \frac{1}{2m}\left(\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2\right) + \frac{\lambda}{2m}\left(\sum_{j=1}^{n}\theta_j^2\right) \tag{2}$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \left(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}\right) + \frac{\lambda}{m}\theta_j \tag{3}$$

3.5.3 Elastic Index, Collaborative Filtering and Predictions. For collaborative filtering, the more specialized low rank matrix factorization algorithm is used to optimize both the cost of the weights and learn characteristics in the feature set simultaneously (Eq. 4). Job predictions for a worker are then produced using a hypothesis function (Eq. 5) that calculates the summation of the products of weights by their corresponding characteristic in the feature set (Eq. 6). We formally call this prediction the ***elastic index*** of the job for the respective worker.

$$J(x^{(1)}, ..., x^{(n_m)}, \theta^{(1)}, ..., \theta^{(n_u)}) = \frac{1}{2}\sum_{(i,j):r(i,j)=1}((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 \tag{4}$$

$$h_\theta(x) = \theta^T x \tag{5}$$

$$EI = W_1 \times A_1 + \cdots + W_x \times A_x \tag{6}$$

where

- $x \in Z^+$ and $x \geq 1$.
- $A_i \in R^+$ and $A_i \geq 0$, $i \in 0, 1, \ldots, x$.

3.5.4 Mean and Distance Functions. To provide the top N ≤ 10 jobs with a designated similarity to jobs already performed by the current user, the absolute of the result from a Manhattan distance function (Eq. 7) was taken to apply similarity testing between the feature vectors of jobs in the worker's history against feature vector of the new jobs being recommended. The difference of 0 implies 100% similarity in job characteristics. A difference in either a positive or negative direction implies differences in job characteristics. The difference may be positive or negative depending on the order of the subtracting feature vectors which implies no meaning; as such the absolute of this result used to denote differences in the calculation of similarities.

In the event a new worker has been added to the system, the recommender has no prior history of work to customize recommendations, as such we apply a mean normalization (Eq. 8) calculation. Using the average rating of all workers, the engine will still be able to make N ≤ 10 top recommendations to the new worker in lieu of a work history or employer feedback. The new worker will begin to receive customized recommendations with at least 1 completed job with corresponding feedback in its work history. Recommendations will adjust and be further customized as the new worker takes up job offerings and as others in the worker pool complete additional jobs.

$$1 - \Sigma(\,|\,(rec) - X(perf)\,|\,) \quad (7)$$

$$Mean\ (\ Y(job)\ ) \quad (8)$$

3.6 Operational Model

3.6.1 Elastic Workflow Model. Figure 8 outlines an elastic workflow model. It consists of 3 major modules, namely: elasticity manager, resources manager and a solution resolution module. The elasticity manager houses an elastic service, that uses the elastic index computational model (Section 3.5.3) to determine the best combination of available workers to assign tasks with the goal of achieving optimum performance. It also contains a learning service which employs machine learning techniques and computational models (Section 3.5) to provide intelligence to the workflow and optimize the assignment of tasks to the best available worker candidates. The resource manager consists of 2 sub-components. It consists of the machine computing element (MCE) and the human computing element (HCE) services. The MCE service manages the automated execution of tasks by machine-oriented workers while the HCE manages the execution of tasks via a crowd of people. The solution resolution module uses a consolidation service that combines the submissions of both HCE and MCE services. Submissions are measured against quality thresholds and they can either be accepted, rejected or ignored. Feedback is used by the learning service of the elastic manager to enhance future decisions.

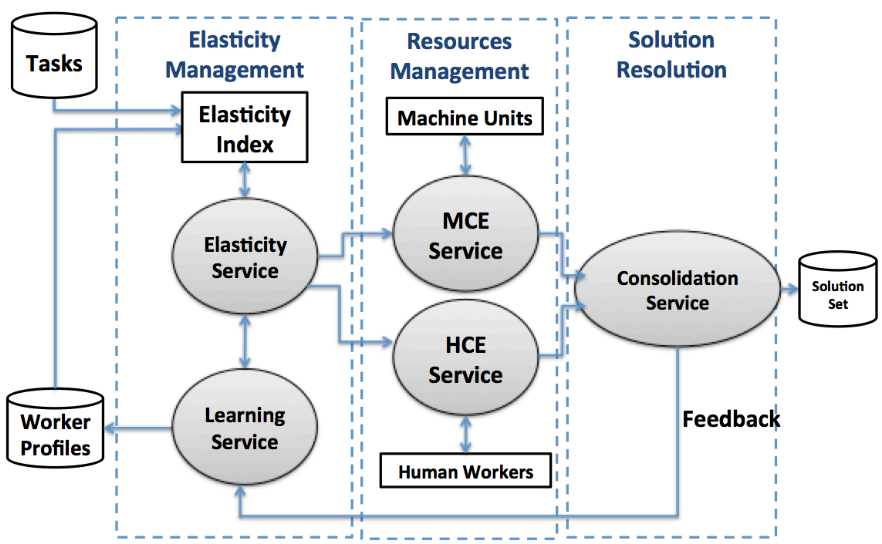

Figure 8. Diagram showing elastic workflow model used in the assignment of jobs.

Listing 4 provides additional detail into the sequencing of the workflows for the elastic and consolidation services. Using recommendations from the learning service, the elasticity manager makes assigns jobs to individuals in the joint human (HCE) machine (MCE) worker pool. Workers with assignments produce solutions that are collected in the consolidation service where they can be accepted, rejected or combined for more optimal solutions by the employers / owners. Results from the consolidation service serve as input to the learning service; it is used to update worker profiles, support future recommendations, and job assignment.

To model a workflow with sequenced tasks or work phases consisting of its own workflows, the notion of daisy chaining the elastic workflow model is introduced in Figure 9. This allows for provisioning of human and / or machine effort at distinct steps within a workflow requiring crowdsourced effort. Chaining also serves the cases where the output from one task or prior workflow is required for another task or workflow to be invoked in the next phase. Iteratively, an entire system workflow can be modeled using indefinite instances of the model or variations of the model introduced in Figure 8.

```
Elasticity Service
Start
        Foreach Task T in the queue
            EI = Calculate_EI(T)
          //Based on EI, assign T tasks
          //to be executed by HCEs and MCEs
            {T_HCE,T_MCE}= Assign (T, EI)
            Output {T_HCE,T_MCE}
         EndFor
Stop

Consolidation Service
Start
         Foreach SubTask T{T_HCE, T_MCE} in the queue
            //Based on EI, calculate the optimal result
            {S_HCE, S_MCE}= Evaluate (T_HCE, T_MCE, EI)
           //Get Task solution from HCE and MCE Services
            S_ECE = getSolution()
            Output S_ECE
         EndFor
Stop
```
Listing 4. Psuedo-code definition of the Elastic Workflow Model used in job assignment.

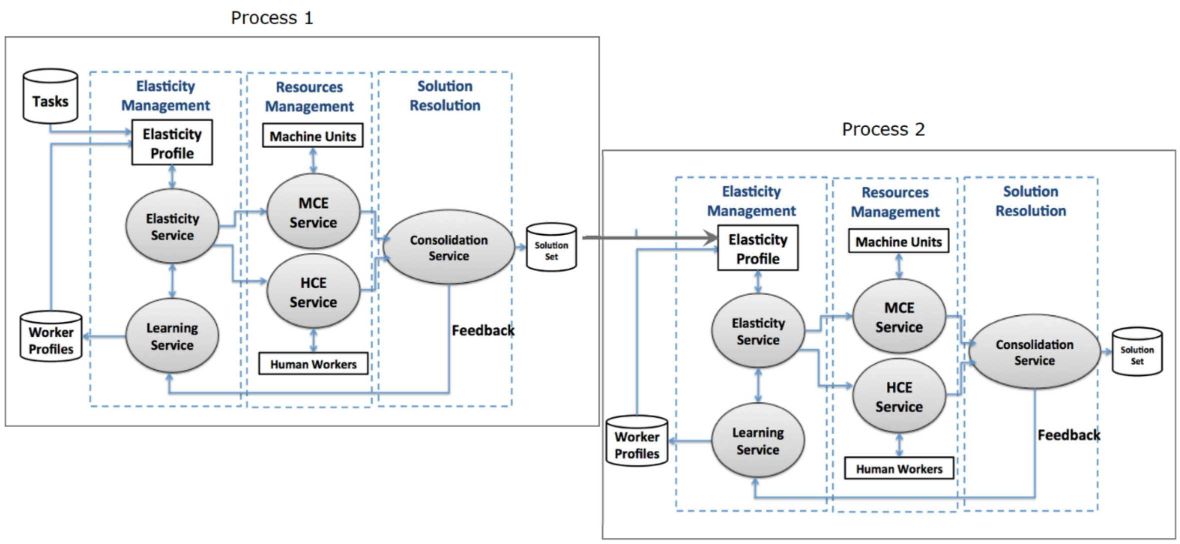

Figure 9. Illustration of daisy chained elastic workflow model for multi-phased jobs.

3.7 Elastic Workflow Algorithms

Building on top of the elastic workflow model, two algorithms were designed to consider human and machine elements in workflows given customizable thresholds and conditions; they are the maximum performance index (MPIA) and weighted metric performance index algorithms (WMPIA). These algorithms operate with workflows for tasks in distinct problem domains.

3.7.1 Maximum Performance Index Algorithm. Using the TCI for a task, the maximum performance index algorithm was designed to maximize precision results of the workflow while ignoring all costs, time and other environment constraints. The algorithm compares the TCI of the current task against an MCE_TCI_Threshold; the MCE_TCI_Threshold represents the average complexity of tasks that MCE’s have satisfactorily completed. If the TCI of the current task exceeds the MCE_TCI_Threshold, the algorithm will then provision HCE’s to complete the task. If the algorithm evaluated the current task’s TCI as being less than the MCE_TCI_Threshold, it allows MCE’s to operate on the tasks. Provided the results of the task meets an acceptable standard in precision, the algorithm terminates; otherwise it sends the tasks to HCE’s for further processing to reduce uncertainty.

3.7.2 Weighted Metric Performance Index Algorithm. Using the elastic index, the weighted metric performance index algorithm was designed as more holistic and practical approach to provisioning MCE's and / or HCE's. The EI considers multiple characteristics of the task modeled as a feature set including the complexity represented by the TCI and environment constraints such as time and costs. It compares the EI for a task against the EI_Threshold of the system; if the system EI_Threshold matches or exceeds the EI for the current task, the algorithm provisions HCE's for the task. In contrast, if the EI for the current task exceeds the system EI, the algorithm provisions MCE's for the task. Like the MPIA, the WMPIA compares the precision of the results of MCE processing to some acceptable standard in precision, the algorithm terminates once precision is acceptable. If unacceptable, the algorithm evaluates whether an increase in environmental constraints such as time and costs (defined by EI_Tolerance) allow for the further provisioning of HCE's to improve precision in results.

```
Algorithm_Maximum_Performance_Index
Start
  Set System MCE_TCI_Threshold
  Set Acceptable_Uncertainty
  Input Task
  Task.TCI = Task.Calculate_TCI()
  If Task.TCI>= MCE_TCI_Threshold
        Task.Precision_Result = processTaskHCE(Task)
  Else
      Task.Precision_Result = processTaskMCE(Task)
     If Task.Precision_Result<Acceptable_Uncertainty
        Task.Precision_Results = processTaskHCE(Task)
     End If
  End If
Stop
```
Listing 5. Psuedo-code definition of the Maximum Index Performance Algorithm.

```
Algorithm_Weighted_Metric_Performance_Index
Start
       Set System_EI_Threshold
       Set Acceptable_Uncertainty
       Set Task.EI_Tolerance
       Input Task
       Task.EI = Task.Calculate_EI()
       If Task.EI<= System_EI_Threshold
           Task.Precision_Result= processTaskHCE(Task)
       Else
            Task.Precision_Result = processTaskMCE(Task)
            If Task.Precision_Result<_
                Acceptable_Uncertainty AND (Task.EI <  (System_EI_Threshold * (1+Task.EI_Tolerance)))
                    Task.Precision_Result = processTaskHCE(Task)
           End If
      End If
Stop
```

Listing 6. Psuedo-code definition of the Weighted Metric Index Performance Algorithm.

3.8 Evaluative Metric Model

3.8.1 Performance and Self Perception Indexes. For each available worker, there exists 2 metrics, average performance as calculated from feedback denoted by P, and the self-perception index as given by the worker based on his or her perceived mastery of a skill. P as outlined by Eq. 9 is the mean performance of a given worker; the mean is calculated by the finding the average performance rating of all tasks (denoted by T) that are at least K% similar to the top ≤ 10 recommended jobs (denoted by R). The SPI is given Eq. 10; it is the division of the self-professed mastery of the worker by their average performance then subtracting from the whole.

$$P(worker) = Avg.\ (P\ for\ T \mid T \in R) \tag{9}$$

$$SPI(worker) = (S / P) - 1 \tag{10}$$

3.8.2 Community Capability and Perception Indexes. The community capability index (CCI) is designed to provide insights into the community's general performance using performance history from completed tasks (Eq. 11). It is calculated as the mean of all the performance indicators (P) for each active worker. The community perception index (CPI) is indicative of the community's own perception of its capability. It is calculated by the finding the mean SPI for all active workers in the system (Eq. 12).

$$CCI(community) = Avg.\ (\ P\ ) \tag{11}$$

$$CPI(community) = Avg.\ (SPI\ (\ \forall\ worker)\ ) \tag{12}$$

4.0 Experimentation and Results

This section consists of 5 studies. All studies consist of objectives and a description of the experiment including but not limited to data sources, methodology, pre-cautions and threats. Results of the studies immediately proceed their respective descriptions. The results are presented in the various graphical forms, such tables and graphs accompanied by appropriate annotations and expansive discussions.

4.1 Study 1 – Formulating a Conceptual Data Model

This study using a data first bottom-up approach, identifies entities and creates an information model with meaningful relationships among entities that allow for the modeling of worker and job data from diverse sources or repositories (Jarrett & Blake, 2016; Jarrett & Blake, 2017). Such a model must be adaptable to diverse sources of information including those illustrated in a distributed architecture supported by the Service Synchronization and Coordination Middleware (Figure 2) and by extension, the elastic workflow operational model (Figure 8). In short, this study seeks to answer ***Research Question 1***:

When managing crowd computing resources spanning human and machine workers, **what general information models effectively define functional, non-functional and evaluative concerns for both types of workers and a wide cross-section of tasks from diverse problem domains?**

To provide further scope for the research question, the experiment was designed with the objective to define *a universal information meta-model that can be used to model crowdsource viable tasks across diverse problem systems allocating jobs to both machine and human working units.*

All job features and required skillsets were identified and recorded. To further understand pervasive characteristics across jobs in the dataset, overlapping features were identified and their percentage for their occurrences across jobs calculated.

Standard and formal approaches to verify conceptual models are devoid of existence (Shanks, Tansley, & Weber, 2003). In lieu of these methods, Shanks et al. (2003) provided some guidance to checking valid conceptual models. They asserted that the model should faithfully represent its focal domain and must be checked against that domain for validity. Faithfulness of semantics of the model is embodied in 4 characteristics. The model should be accurate, complete, conflict-free and bear no redundant elements. The reified OO model was evaluated against these criteria.

4.1.1 Data and Sources. Data consisting of 300 job instances and their characteristics were gathered from 3 major labor markets, Amazon Mechanical Turk, MicroWorkers and Upwork. Figure 10 illustrates a job pulled from UpWork requiring experience in marketing, possess social media accounts and is at the level of an entry level freelancer.

4.1.2. Results. Table 4 shows all 67 job characteristics found across all 300 randomly selected jobs. As depicted in the figures, the skill of writing in English was required 71 times to make it the most demanded characteristic, with a 23.67% overlap across all jobs in the dataset. Freelance type of employment with an intermediate skill level was also a characteristic that was pervasive across the job dataset; it was required 64 teams with an overlap of 21.33% over all jobs in the dataset. Other levels of freelance type of employment with levels entry and expert were also found within the top 5 characteristics with 11.33% and 10% respectively. This makes freelance employment a pervasive feature across jobs on crowdsourcing labor markets.

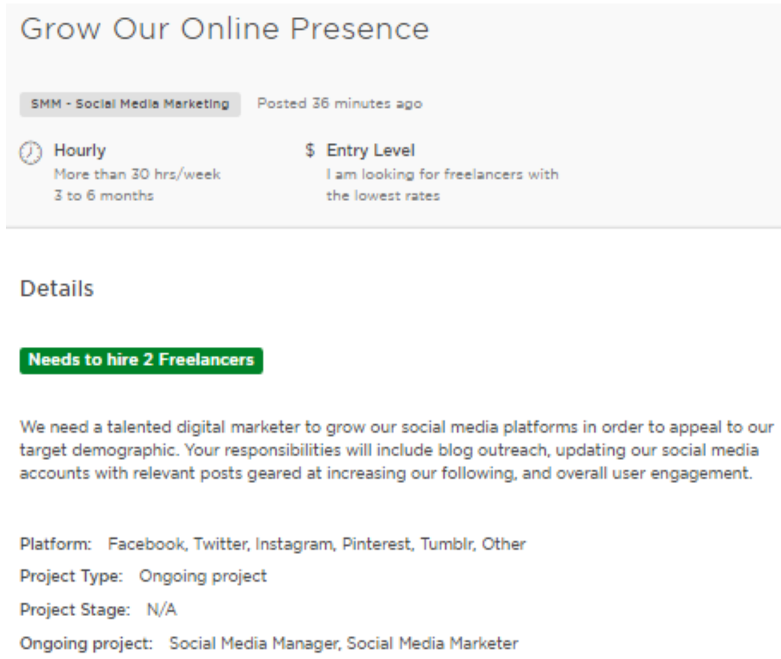

Figure 10. Showing a sample job pulled from UpWork metrics were extracted.

Table 4

Task Characteristics across Diverse Sources

TOP METRICS	Times used	Overlapping (%)
English_Writing	71	23.67
Freelancer_level_intermediate	64	21.33
Graphic_Design	34	11.33
Freelancer_level_entry	34	11.33
Freelancer_level_expert	30	10.00
Email_Account	25	8.33
Wordpress	24	8.00
IOS_Device	21	7.00
English_Reading	20	6.67
Transcription	17	5.67
Adobe_Photoshop	16	5.33
Google_Account	15	5.00
PHP	15	5.00
HTML	14	4.67
Facebook_Account	13	4.33
Web_Development	13	4.33

Youtube_Account	12	4.00
Adobe_Illustrator	12	4.00
English_Listening	11	3.67
CSS	10	3.33
Video_production	10	3.33
Web_Design	9	3.00
Android_Development	9	3.00
English_Speaking	9	3.00
IOS_Development	8	2.67
Video_editing	8	2.67
JAVA_SCRIPT	7	2.33
Android_Device	7	2.33
Marketing_experience	6	2.00
Social_Media_Accounts	6	2.00
Translation	5	1.67
Adobe_After_Effects	5	1.67
Data_Mining	4	1.33
Categorizations	4	1.33
Pinterest_Account	4	1.33
Magento_platform	3	1.00
Photo_Editing	3	1.00
Typing	3	1.00
Adobe_InDesign	3	1.00
Google_Chrome	3	1.00
Ionic_Framework	2	0.67
Instagram_Account	2	0.67
Amazon_Account	2	0.67
Russian_Writing	2	0.67
Windows_Phone_development	2	0.67
Microsoft_Excel	2	0.67
Microsoft_PowerPoint	2	0.67
Microsoft_Word	2	0.67
Mobile_Device	2	0.67
Native_Russian_Speaker	1	0.33
PhoneGap	1	0.33
Linkedin_Account	1	0.33
Portuguese_writing	1	0.33
Russian_Reading	1	0.33
Tumblr_Account	1	0.33
Twitter_Account	1	0.33
US_tax_law	1	0.33

Spotify_Account	1	0.33
Traffic_Geyser	1	0.33
Alignment_Training	1	0.33
Drawing	1	0.33
Dropbox_account	1	0.33
.NET_Framework	1	0.33
AngularJs	1	0.33
Autocad	1	0.33
International_VPN_with_dynamic_IPs	1	0.33
Finnish_Writing	1	0.33

4.1.3. Conclusion. Despite the job data being aggregated from 3 isolated labor markets, it is evident that, jobs offered across major platforms are characteristically similar. Consequently, recruiting efforts can be consolidated and worker pools can be shared to support an Open Push-Pull model. This approach can be used as a collaborative approach to addressing recruitment and retention challenges currently faced in crowdsourcing with the open call model. This model is also suitable for freelance type of personnel who seek flexible employment across employers as is reflected in the analysis of the data.

The reified OO model allows for a Just-in-Time approach to modeling tasks and other dynamic characteristics at runtime. Characteristics can be tailored to the specific requirements of the tasks across the domains they belong. This facilitates completeness as tasks that require more details bringing about more accurate representations. With objects only required to carry the characteristics that are specific to its description, the reified OO model reduces conflict and eliminates redundancy.

The design of this reified OO model is further validated by the guidance outlined in COMET (Gomaa, 2001). The model supports entities, meaningful relationships between them and elements supporting control and application logic through the elastic index and evaluative stores.

Using data from multiple data sources, the reified OO model was validated as being a suitable data container as it was able to sufficiently and completely model all features for a wide domain of tasks spanning random 67 characteristics. By induction, it is concluded that it can support additional characteristics.

4.2 Study 2 - Delegation of Work to Different Work Resources

Study 2's primary objective was to ascertain when it is best to assign a crowdsourced task to the most appropriate worker within a labor pool of available machine and human computing elements (Jarrett et al., 2014). The study seeks to answer part 1 of **Research Question 2**: **(Is there an operational approach that enables systematic and reliable delegation of work across human and machine work resources)**?

To investigate this broad idea, the research question was further scoped to the following objectives:

- *O1 – When human and machine elements can perform the same task, is there a general model that can define and evaluate their respective performance outcomes simultaneously?*
- *O2 - Can experimentation in a specific domain, such as face recognition, uncover the most appropriate, shared evaluative attributes that have cross-domain applicability?*
- *O3 - Can the specific performance variations in real-life experimentation enhance our overall understanding and ultimately lead to a more generalized elastic model?*

For objectives (O1 and O2), the elastic service workflow (Figure 8 & Listing 4) was applied to the problem domain of face recognition; Figure 11 illustrates the physical apparatus for the experiment. For O1, an experiment was conducted to evaluate the performance index to construct an elasticity profile for a workflow with mixed workers. For O2, the variability in performance of both machine and human workers were analyzed given the complexity of the task. Reference images of popular individuals were stored and made accessible to an off-the-shelf face recognition tool, the machine computing element (MCE). Using logic of the MPIA (Listing 5), test images of the same individuals were assessed in the elastic workflow for identification. The elasticity server first delegated the image recognition tasks to the machine worker for identification; it assessed the images and provided suggestions pertaining to the identity of the person in the image. Suggestions are considered as test images that meet a criterion of a match of at least 60% similarity to a stored reference image (Figure 12). The tasks coupled with corresponding machine augmented suggestions were then forwarded to human workers via a custom mobile crowdsourcing application for final assessment. Before receiving a task, human workers were first asked to select their area of expertise from the available categories of the individuals (Figure 13). Upon receipt of tasks, the human workers had the option to make a submission on the task by selecting one of the machine augmented suggestions, provide their own answer or state they could not idenfity the individual (Figure 13). Listing 7 outlines the workflow for both machine and human assessment of tasks for this experiment.

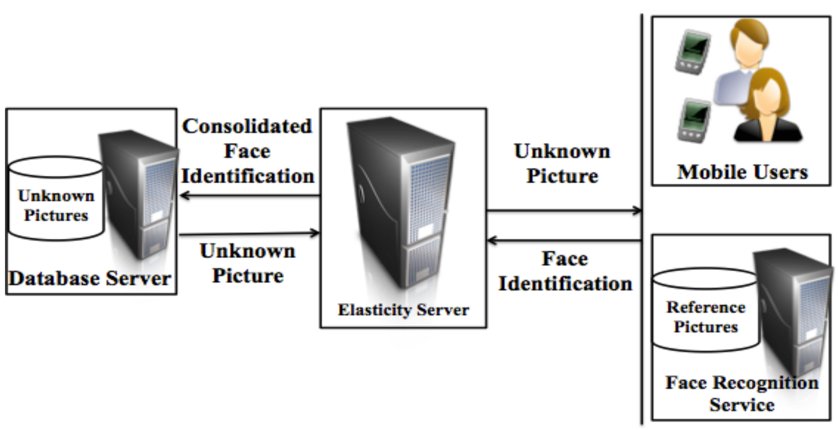

Figure 11. Face Recognition System Architecture Based on the Elastic Framework.

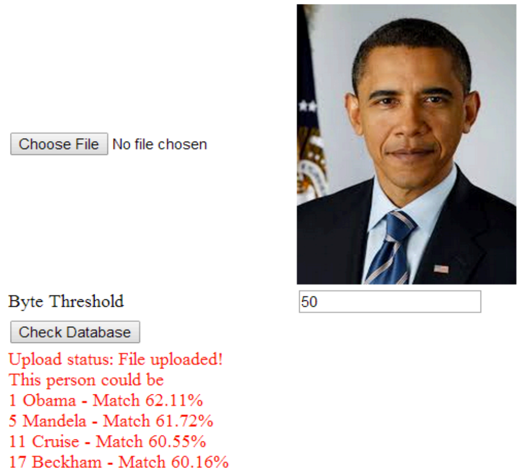

Figure 12. MCE Component giving predictions for testing image.

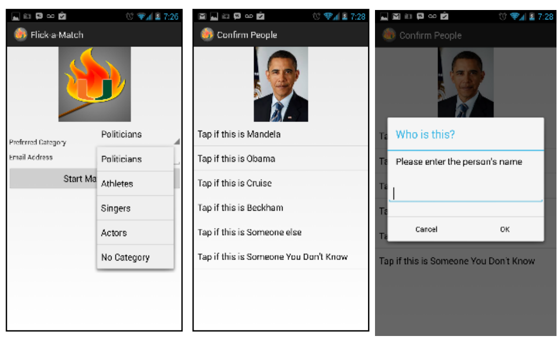

Figure 13. MCE components sent to Android Mobile crowdsourcing application for HCE's to provide feedback based on MCE predictions and their own knowledge.

4.2.1 Data and Sources. The 2 sets of pictures of 23 popular individuals consisting of actors, politicians, singers and athletes were obtained from Google Image Search. The images were selected using 6 identifiable metrics; they are outlined as follows:

- **(M_1)** Face Angle – Face in picture is 0° to an angle of 90°
- **(M_2)** Eyes – Eyes in picture are fully open to Shut
- **(M_3)** Mouth – Mouth is closed to fully open
- **(M_4)** Image Angle – Image is taken at an angle of 0° to 90°
- **(M_5)** Face Magnification – Face is close to far away
- **(M_6)** Image Quality – Quality of Image (lighting, pixels, etc.) High to Poor

```
Procedure WorkFlowECE
Start
        Foreach ReferencePicture to Add to System
                Input ReferencePicture in ReferenceSet
        EndFor

        Foreach TestingPicture to Add to System
                Input TestingPicture in TestingSet
        EndFor

        Foreach TestingPicture in TestingSet
                ListSuggestions=Call_ MCEMatch(accepts_
                TestingPicture, ReferenceSet)
        EndFor

        Send ListSuggestions to HCE via CrowdSourcing

        //WorkFlow executed on Mobile Device to
        //consolidate MCE with HCE to give ECE
        Foreach MCESuggestion in ListSuggestions
                Submit HCE Feedback for _
                MCESuggestion to Learning Service
        EndFor
        //WorkFlow on Mobile Device Ends

        //Begin MCE vs ECE analysis in Learning
        //service
        EI-ECE = Analyze HCE Feedback for
        MCESuggestions  //EI=ECE is EI-MCE //+ EI-HCE
        EI-MCE = Analyze ListSuggestions for Positive
        Identification
        Results = Compare EI-ECE vs EI-MCE
        Show Results
Stop


Function MCEMatch Returns ListSuggestions Accepts_
 TestingPicture, ReferenceSet
Start
        Foreach ReferencePicture in ReferenceSet

                ResultSimilarityMatch=Compare _
                TestingPicture to ReferencePicture using_    MCE Face Recognition with bit _
                Threshold value 50

                If ResultSimilarityMatch > 60%
                        Add to ListSuggestions
                End If
        EndFor
        return ListSuggestions
Stop
```

Listing 7. Showing Algorithm Psuedo-BASIC Workflow of Mobile Face Recognition System.

These metrics characterize various features that affect the ability of human and machine in identifying faces in the experiment. The TCI (Eq. 1) was calculated using the evaluation of metrics using a nominal scale from 1 to 5, with 5 indicating the maximum level of difficulty. Weights for all metrics were set to 1 for equal consideration. Human respondents consisted of 30 volunteers, 18 years and older spanning the United States, Canada, France, the Middle East and Jamaica.

4.2.2 Experimental Precautions. To reduce noise in image recognition, images were first pre-processed, and the faces cropped out discarding unnecessary background portions (Figure 14). The cropped images were then layered using a gray-scale to minimize the impact of color on the recognition process (Figure 15).

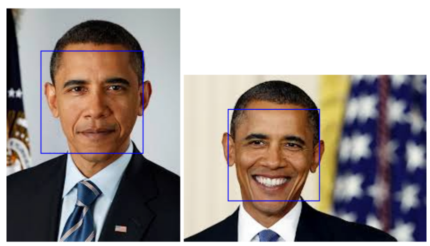

Figure 14. MCE component detecting faces in testing (left) and reference (right) pictures.

Figure 15. MCE component cropping faces and gray-scaling pictures to minimize impact of colors and focus on facial features.

4.2.3 Threats to validity. The background of human participants can potentially affect the outcome of human success for this task. Likewise, the algorithms, techniques and approaches used in the face recognition kit, in addition to the characteristics of the picture impacting facial recognition, can potentially affect machine success.

4.2.4 Results. The face recognition precisions are depicted in Figure 16. As shown in the figure, out of the 23 test case pictures, the recognition precision for 22 test cases were increased by applying our ECE approach. The combined effort of MCE and HCE increased the probability of positively identifying an individual in the pictures in the testing set by a minimum of 16.67% and a mean of 55%. For test cases (1, 2, 6, 8, 16, 18, 20, 21 & 22), the pictures where MCE made suggestions consisting of a postive identification, ECE effort increased the precision by an average of 67.6%. In test case 8, MCE positively identifed the individual in the picture providing one (1) positive match. ECE responses reduced this probability by 53.3% as not all respondents positively identified the individual in the picture despite that the MCE component provided the correct suggestion; this is assumed to be related to the human respondents' prior knowledge of the individal and exposure to affairs that would enable them to identify the individual. Seven (7) of thirty (30) respondents said that their expertise was in identifying politicians, however only three (3) of the seven (7) positively identified the individual in test case 8; hence the majority of respondents had no prior knowledge of the individual.

A minimum increase of 16.67% was observed and an average increase of 51.3% in positive identification of individuals when MCE and HCE efforts were combined; this includes situations where the MCE component failed to provide suggestions. As seen in test case 17, MCE failed to identify or make suggestions for the test portrait of athlete Usain Bolt however ECE was able to identify the athlete. Facial expressions and difference in the angles of the face in the pictures impacted the performance of the MCE to positively identify the athelete (see Figures 17a and 17b). Poor performance of MCE may be attributed to face recognition approaches and techniques employed by the face recognition service. Humans on the other hand positively identified the athelete with an accuracy of 70% irrespective of facial expressions or other variances of the athelete's portrait.

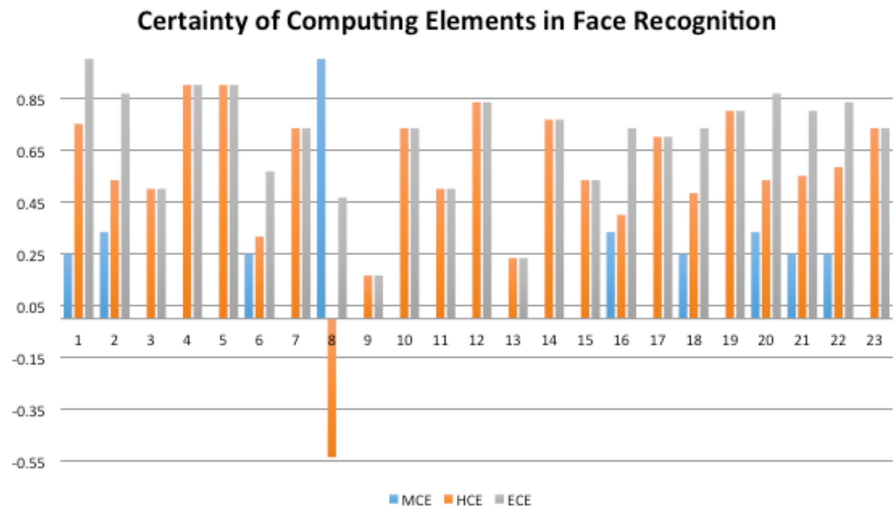

Figure 16. Performance Index bearing certainty for Computing Elements in elasticity framework for face recognition.

(a) (b)

Figure 17. Reference portrait of Usain Bolt Figure 17(a) used by MCE as a reference to identify the athlete in Test Case 17. Figure 17(b) is the test portrait to be identified by MCE and ECE processes.

The median and the mode measurements of the MCE effort were 0; however, there was a mean positive match of 14.13%, as 14 of the 23 cases did not produce any suggestions positively identifying the individual. When using ECE, the median has significantly increased to 73.33% with a mode of 73.33% and a mean positive match of 69.13% (Figure 18). Results clearly show that employing our elasticity approach, the probabilities of positive identification increase significantly.

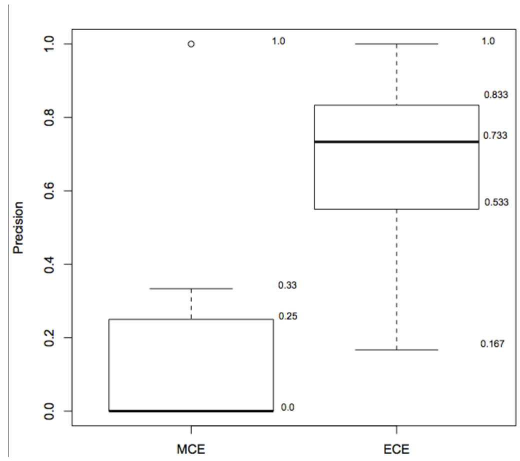

Figure 18. MCE vs. ECE box-plots showing dispersion of probabilities to positively identifying an individual in 23 test cases.

Computing elements' (MCE, HCE and ECE) performance indexes for successful face recognition were measured and compared. As Illustrated in Figure 16, almost all test cases with exception of test case 8, the MCE had the highest levels of uncertainty of the three types of computing elements. Test case 8 had 100% accuracy from MCE with a reduced accuracy of ~53% accuracy from HCE resulting in lower combined ECE performance; further analysis shows respondents in particular countries didn't know the public figure in the test case. It can also be seen in 16 of the 23 test cases, MCE could not identify individuals and as such, recorded no performance index. When HCE's were assigned the task, HCE's increased chances of successful face recognition by an average of 55%. When both MCE and HCE effort are combined to give ECE performance, ECE increases probability of successful face recognition by an average of 69%.

The TCI for each test case was calculated and compared as shown in Figures 19 and 20. The testing data set had an average TCI of 0.32, a range of 0.3, a minimum TCI of 0.233 and a maximum TCI of 0.533. XY-Plots were used to establish correlations between the independent variable TCI and the corresponding performance index for the computing element for each test case.

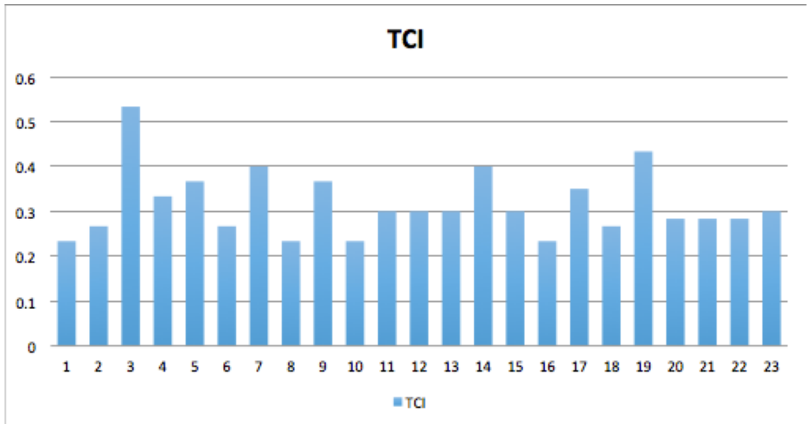

Figure 19 Bar graph showing TCI for each test case in dataset.

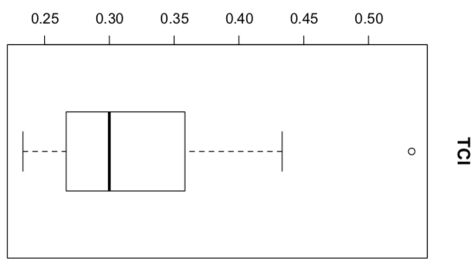

Figure 20. Box plot showing TCI for each test case in dataset.

Figures 21, 22, 23 and 25 are XY-Plots showing task TCI against certainties MCE, HCE, ECE, MCE/HCE/ECE respectively. Figure 21 shows an exponential decrease in the performance index of MCE components as the TCI for the task increased. It also portrays the MCE's inability to function once TCI surpasses a value of ~0.3. The point bearing a Y value of 1 indicating perfect performance is an outlier as this result was not replicated in any other test case. Figure 22 shows a graceful increase of the performance index of HCE components as TCI increased; performance remained relatively consistent between ranges of 0.4 to 0.6. An outlier bearing a Y value of -0.5 resulted in test case 8 (Figure 22). For this test case, MCE performance was perfect with a certainty of 1, however uncertainty was increased when combined with HCE performance. Figure 23 also shows relatively consistent behavior of ECE as TCI increased with performance indexes ranging from 0.6 to 0.8.

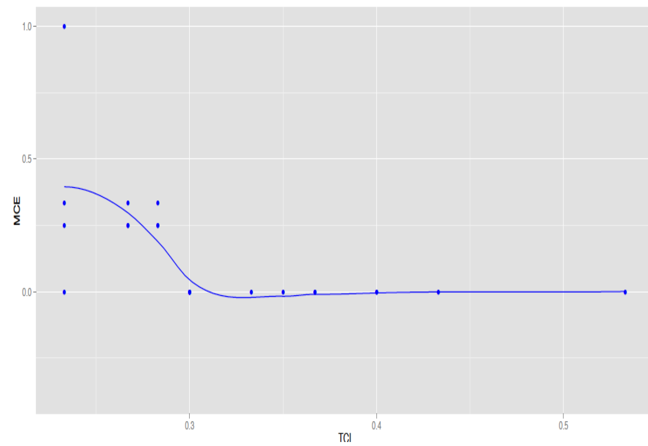

Figure 21. XY-Plot, TCI vs. MCE Certainty.

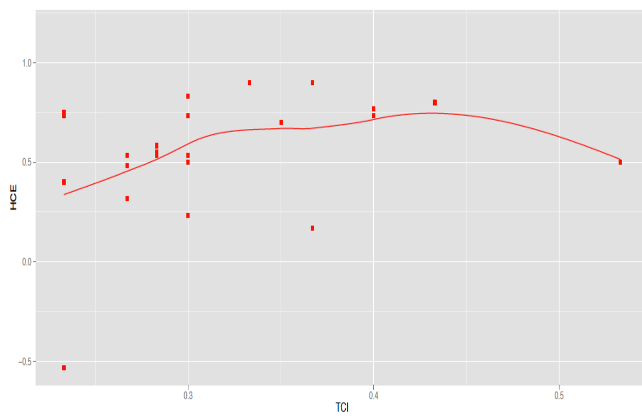

Figure 22. XY-Plot, TCI vs. HCE Certainty.

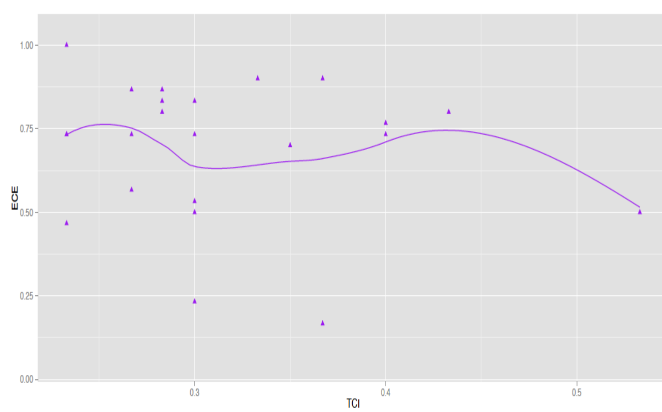

Figure 23. XY-Plot, TCI vs. ECE Certainty.

Superimposing the three graphs above (Figures 21, 22 and 23) into Figure 24, we find HCE and ECE performance converging; this accounts for high failure (16 of 23 test cases) of MCE component where MCE had no direct contributions to ECE results. Consequently, the inability of the MCE component to perform after TCI is ~0.3 or higher records the lowest performance of the three computing elements in Figure 24.

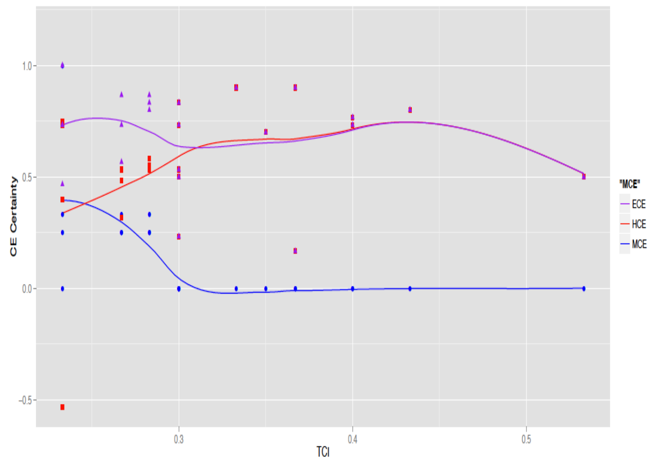

Figure 24. XY-Plot, TCI vs. MCE / HCE /ECE Certainty.

The possible existence of poorly performing types computing elements, further highlights the need for an approach which allows for dynamic assignment of tasks to those with proven performance history. In this experiment, there existed a poorly performing MCE and better performing pool of HCE's for the given tasks. A dynamic approach would assign tasks of similar nature to HCE components possessing reputable work history of tasks with similar characteristics and requirements.

4.2.5 Conclusion. This study demonstrates the implementation of the elastic workflow model to leverage the both human and machine computing elements to solve a complex problem. The study applied the proposed elastic workflow to multiple face recognition tasks. The experimentation in that domain demonstrates that applying elasticity to the task of face recognition significantly increases the probability of positively identifying an individual in a picture. Virtualizing and provisioning humans as computing elements through crowdsourcing and integrating them with automated approaches produced positive results in test cases where the software alone failed to identify faces.

The study verified the effectiveness of the TCI as a model to compute the difficulty of a task given its characteristics (Section 3.5.1). In observing the TCI against the performance index of the various computing elements to complete the tasks (Figure 24), the results showed that a combined machine and human approach in most cases, produced the most optimal results. In the few instances of failure, further analysis revealed that computing elements did not possess any or sufficient background to perform the task with a favorable outcome as described in test case 8. Given this discovery, the Maximum Performance Index Algorithm (Section 3.7.1) and Weighted Metric Performance Index Algorithm (Section 3.7.2) were integrated and shown to be effective as a part of the Elastic Workflow Model.

4.3 Study 3 - Usefulness of Evaluative Metrics

This study investigates the validity of professed skills by a worker against their actual competencies inferred through performance. Using this approach, an extrapolation was used to ascertain the nature of labor force; the labor community's competency and its own awareness of that competency (Jarrett et al., 2015). The study seeks to answer **part 2 of Research Question 2**: Is there an operational approach that creates and incorporates metrics that allow for the **evaluation of collective capabilities of a worker pool**?

The following objectives were formulated to add further scope to the research question:

- *O1 - Is the workers' self-evaluation of expertise a valid measure for employers to use to determine the actual performance of workers?*
- *O2 - What perceived level of workers are most consistent with their actual performance level?*

To test the research objectives and research question, a crowdsourcing task requiring the translation of Portuguese idioms to English was designed. Idioms were independently and subjectively evaluated by 4 native Brazilians acting as evaluators and owners of the tasks; each task was rated on a scale of 1 through 5 with 5 being the most difficult. The average of the difficulty of the tasks as given by the Brazilians, was calculated to obtain the final difficulty of the task.

All respondents were asked to indicate their level of competency in English before being given language tasks (Figure 25). There were 5 levels of competencies that were aligned with the difficulty scale used to indicate the difficulty of the translation task. The competencies are outlined below:

1. Beginner
2. Intermediate
3. Advanced
4. Fluent
5. Native

Upon indicating their English mastery, the respondents were then asked to translate a maximum of 5 idioms from a selection of 34 (Figure 25). All idioms were also translated by invoking Google Translate to represent a machine-oriented worker. All translations were then evaluated by the recruiters on a 5-point scale aligned with the levels of mastery. The average rating for each task was calculated and assigned as the score for the respective respondent. Using the evaluations, a collaborative filtering based recommender (as outlined in section 3.5, equations 2 through 5) was used to recommend $N \leq 10$ newly introduced phrases to translate. The recommended phrases are at least 70% similar in difficulty to phrases already translated by the worker. The recommender also made predictions on possible scores for the newly recommended translation tasks. Using the actual performance, their self-professed level of competency and predictions, the evaluative metrics P, SPI, CPI and CCI (outlined in section 3.8 equations 9 through 12) were applied to evaluate the self-awareness and competency of the labor pool.

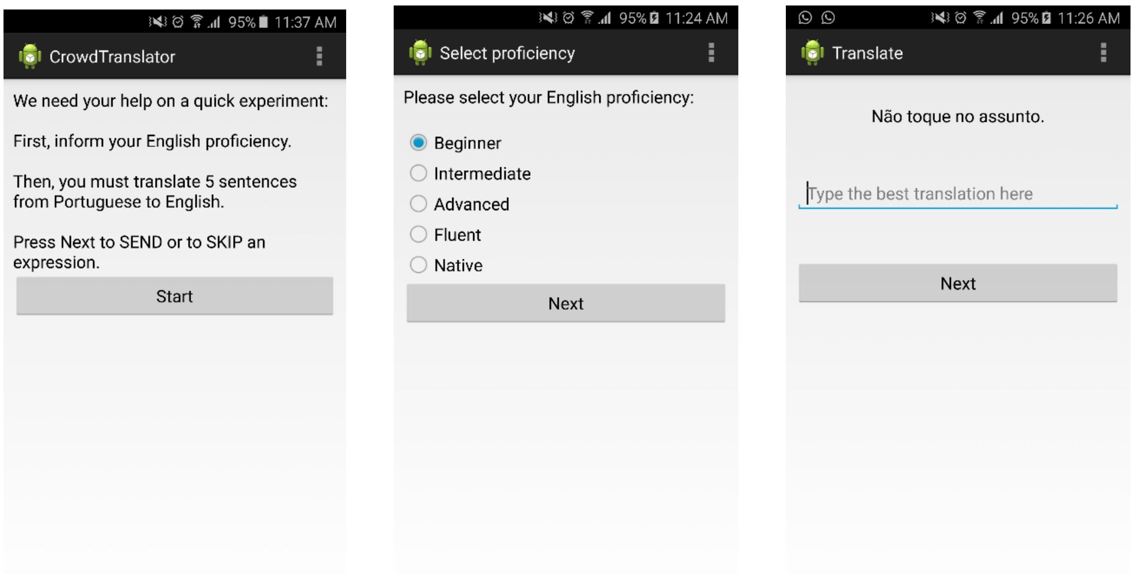

Figure 25. Screenshots of worker crowdsourcing mobile interface.

4.3.1 Data and Sources. All 52 respondents were native Portuguese speakers of Brazilian decent, some residing in Brazil and others in the United States; the respondents had varying levels of competency in English. The machine worker was invoked through Google Service API's for translations via Google Translate.

4.3.2 Threats to validity. The background of human participants can potentially affect the outcome of human success for this task. Likewise, the algorithms, techniques and approaches used in the language translator, can potentially affect machine success. Evaluations by the Brazilian evaluators were also subjective.

4.3.3 Results. In Study 3, patterned and temporal results were observed. Patterned results are observed on collectively on all 52 respondents and the machine worker. Temporal results are results observed as respondents are added to the workforce over 5 iterations in increments of 10.

The first iteration begins with 10 human workers and 1 machine worker. The final iteration sees the addition of 12 human workers; the standard increment and the 2 remaining. Despite the size of the workforce being small, even in preliminary form, this case study supports the contributions. In this section, the patterned results are presented first followed by the temporal.

4.3.4 Patterned results. Several observations were made across our 52 human workers and the single machine worker. System parameters were configured to produce individual recommendations with at least 70% similarity in difficulty to jobs a worker successfully completed. Using collaborative filtering, it also took into consideration the job difficulties and the employers' evaluations of workers on those completed jobs. The difficulty of the jobs recommended (Figure 26) and the jobs selected (Figure 27) were on average 2 levels less than the workers' respective self-evaluated levels of competence in English.

Difficulty of the jobs recommended were a direct reflection of jobs the workers selected. Given the nature of workers' job selections, the recommended job difficulty and job difficulty of tasks completed are very similar in shape; this is evident when the two are graphed together (Figure 28).

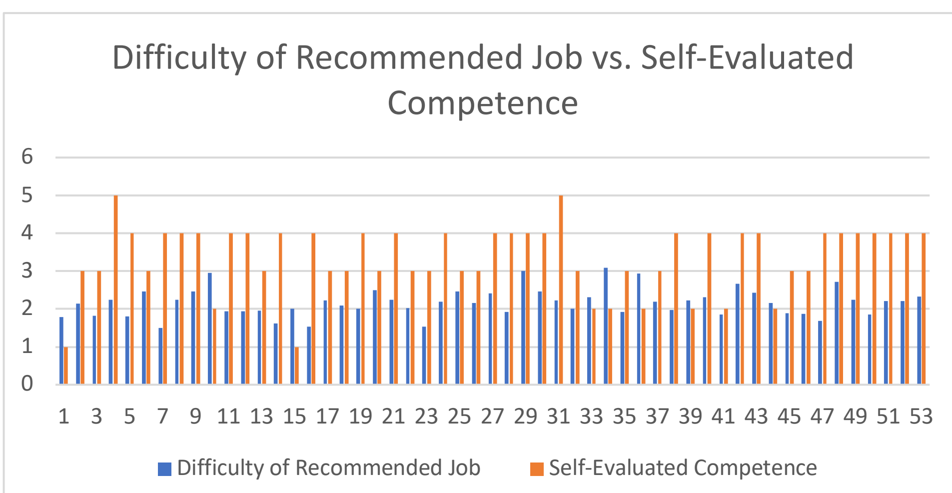

Fig. 26. Bar graph comparing the difficulty of recommended jobs to the workers' self-evaluated competence.

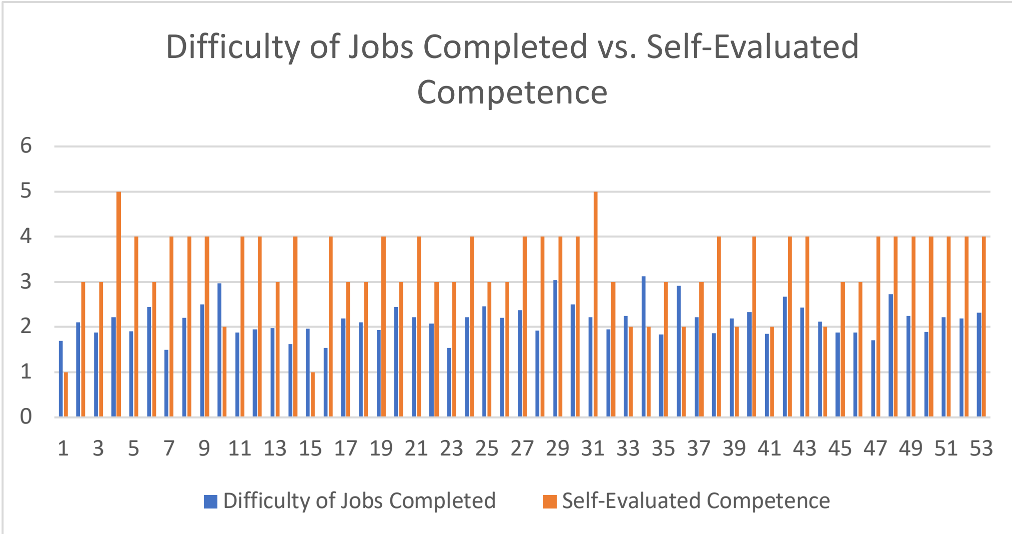

Fig. 27. Bar graph contrasting worker's self-evaluated competence and the difficulty of jobs already completed.

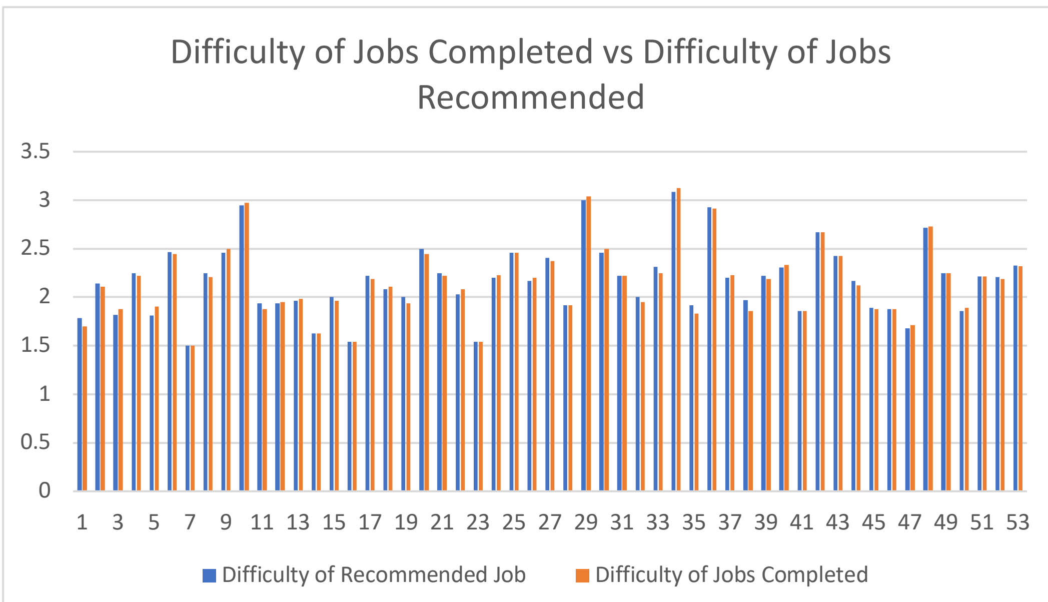

Fig. 28. Bar graph contrasting diffiuclty of jobs already performed by worker vs the difficulty of jobs recommended to worker.

Collectively, the workers' average competencies (CCI) were higher when compared to the average level of difficulty of the jobs they selected (Figure 29). As such, the community's overall level of competency (CCI) was higher than the level of difficulty of jobs they completed. For 66% of the workers, the average level of competence calculated by the system given individual performance (P), was higher than the self-evaluated level (SPI) indicated by the worker (Figure 30).

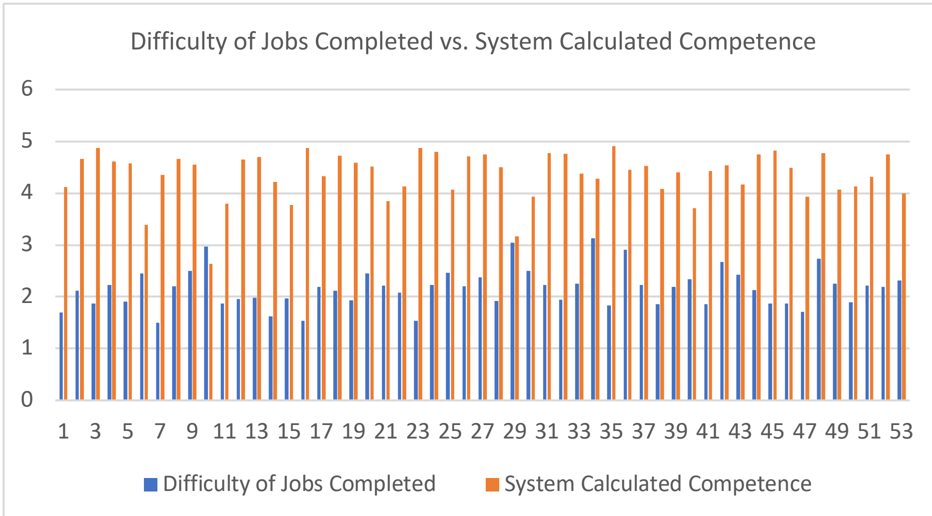

Fig. 29. Bar graph contrasting system calculated competence and the difficulty of jobs already completed.

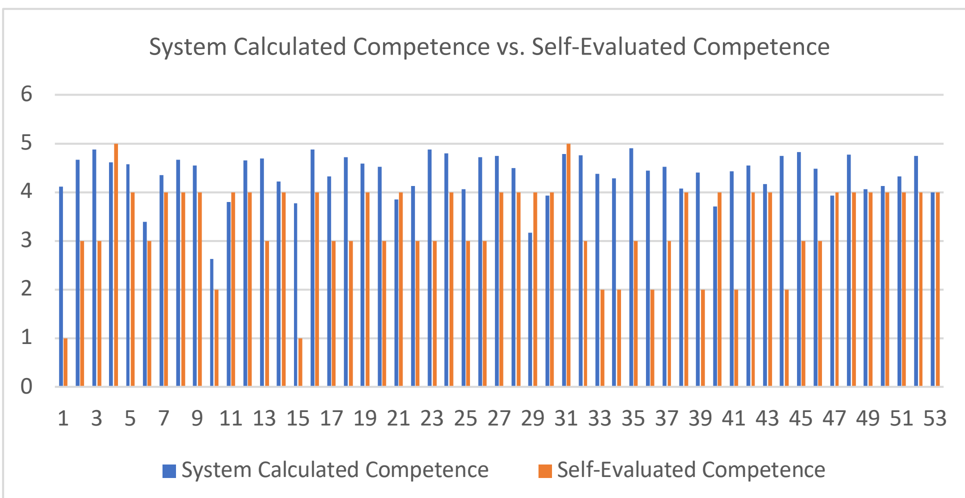

Fig. 30. Bar graph contrasting worker's system calculated competence and the worker's self evaluated competence.

Using the assessment metrics, the labor force can be assessed from an individual and collective perspective. From the calculated SPI's, 35 of 53 workers underestimated their true potential, 17 had a fair assessment of their skills and a single worker over estimating (Figure 31).

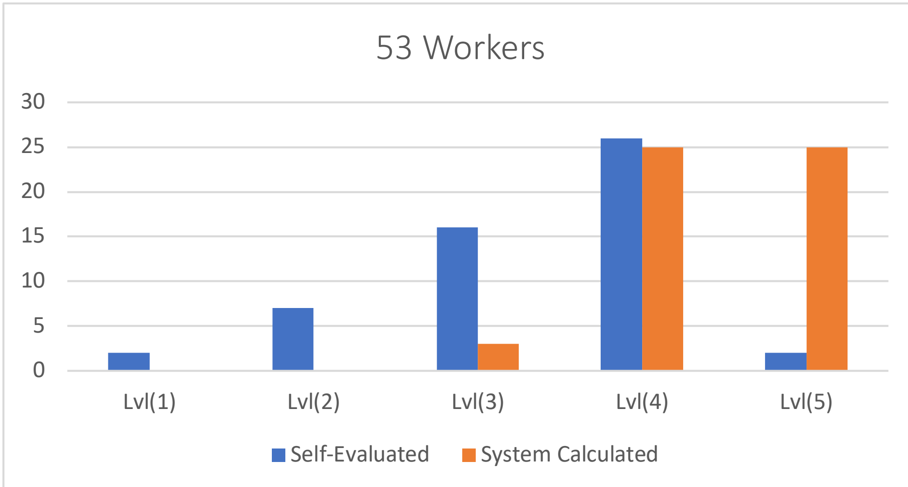

Fig. 31. Bar graph contrasting community's self evaluation and calculated ratings.

A fine-grained analysis reveals the 2 workers with SPI 1 and the 7 with SPI 2; all had higher levels of performance. For workers with SPI 3, 15 of 16 had higher levels of performance, with the other worker giving a correct assessment. For those bearing SPI of 4, there exists a single over estimation with 11 giving consistent self-assessments and another 14 with higher levels of performance being assigned to SPI 5. There were 100% consistent self-evaluations with 2 workers with SPI 5 (Table 5). In summary, the average SPI of all workers 3.36 in contrast to an average competence level of 4.35 (Figure 32).

Table 5
Self vs. Calculated Competences for Users

	Self	Calculated
Level 1	2	0
Level 2	7	0
Level 3	16	3
Level 4	26	25
Level 5	2	25

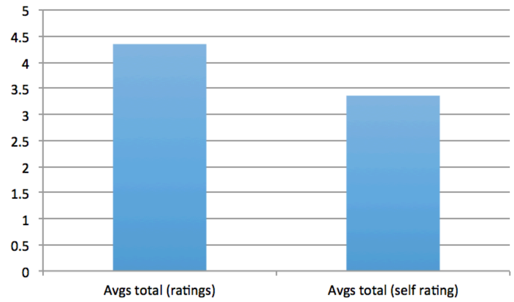

Fig. 32. Average actual rating vs. average self-evaluated rating.

4.3.5 Temporal results. Temporal results were studied in a simulation where human workers were introduced to the work force over 5 simulation steps in increments of 10; the final step accommodated the remaining 2 human workers. The machine worker was also included from step 1. Figures 33 to 37 show the number of workers by their self-evaluated competence (derived from the SPI) against their actual performance (derived from P) as the workers completed tasks over 5 simulation steps. In each simulation, it was found that workers had tendencies to evaluate themselves at lower levels of competencies in contrast to the system calculated competencies based on their actual performances.

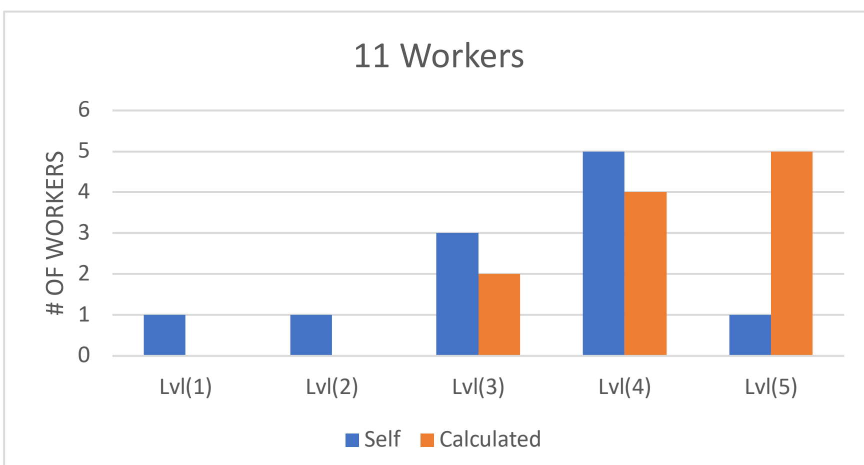

Fig. 33. Bar graph showing worker's self-evaluated competence vs system calculated competence with 11 workers.

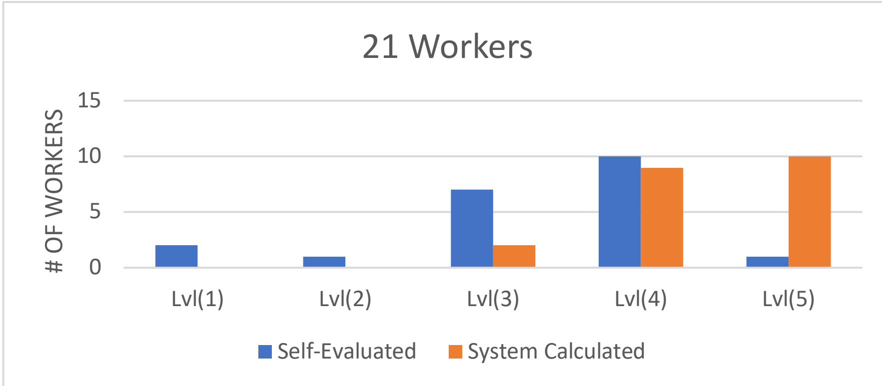

Fig. 34. Bar graph showing worker's self-evaluated competence vs system calculated competence with 21 workers.

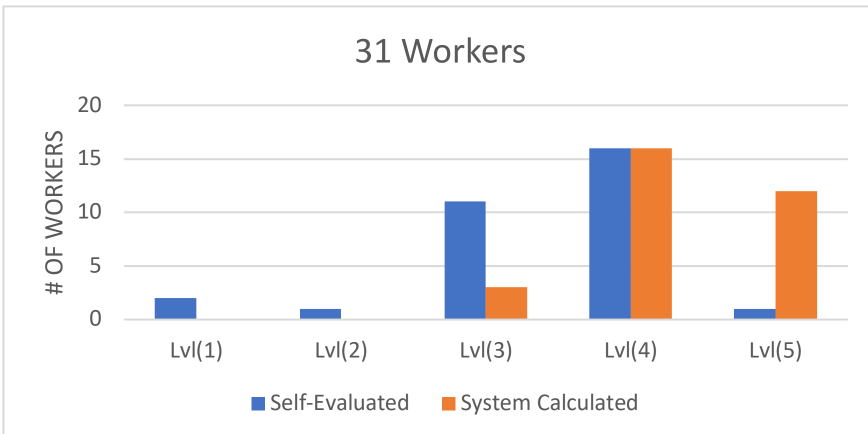

Fig. 35. Bar graph showing worker's self-evaluated competence vs system calculated competence with 31 workers.

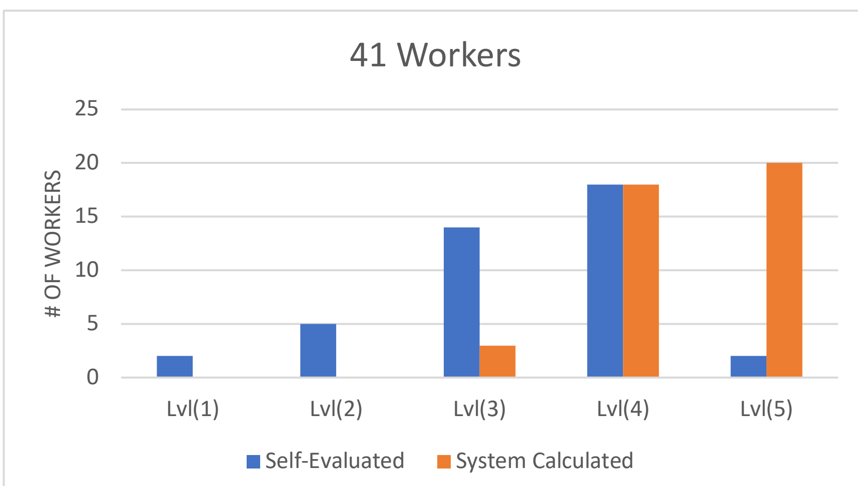

Fig. 36. Bar graph showing worker's self-evaluated competence vs system calculated competence with 41 workers.

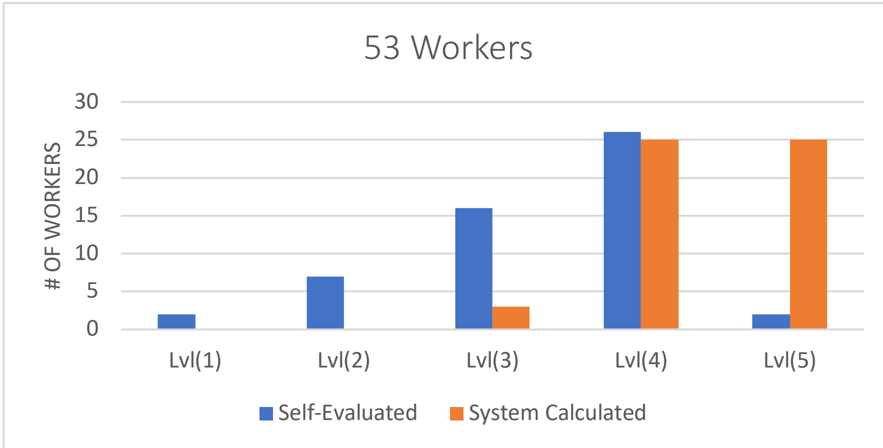

Fig. 37. Bar graph showing worker's self-evaluated competence vs system calculated competence with 53 workers.

Figures 38 to 42 shows the number of workers within a particular level for each simulation step comparing compares the self evaluated competency (derived from the SPI) against the system calculated competency (derived from P). There is a noticeable absence of workers in system calculated competency levels 1 and 2 the across simulation steps (Figures 38 and 39); despite these self-evaluated levels being overwhelmingly selected. At level 3, most workers evaluated themselves inconsistently with the system calculations across steps (Figure 40) as the majority were promoted to higher levels of competency. Workers self-evaluating as level 4 (Figure 41) had the most consistent competencies across all steps, almost mirroring the system calculated levels; this makes this group the most self-aware. Not many workers evaluated themselves with a level 5 competency, however system calculations showed the greatest disparity in this group when comparing results making this group the most unaware of their true capabilities (Figure 42).

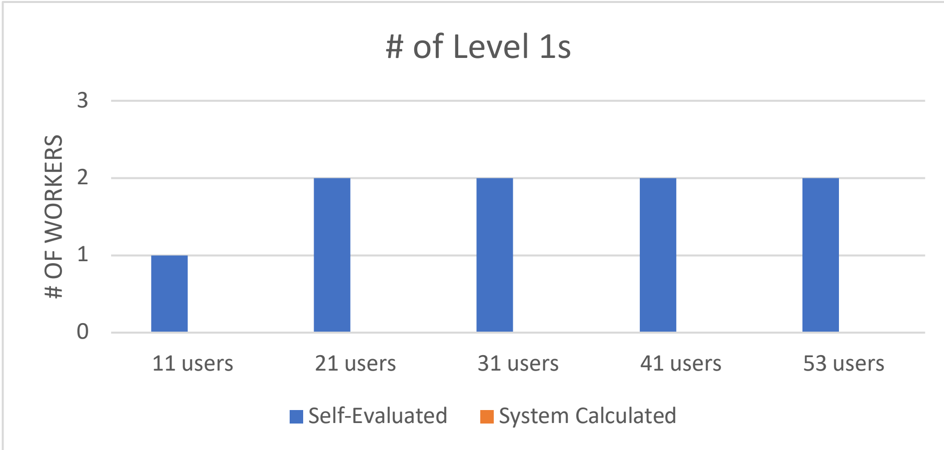

Fig. 38. Bar graph showing worker's self-evaluated competence vs system calculated competence at competence level 1.

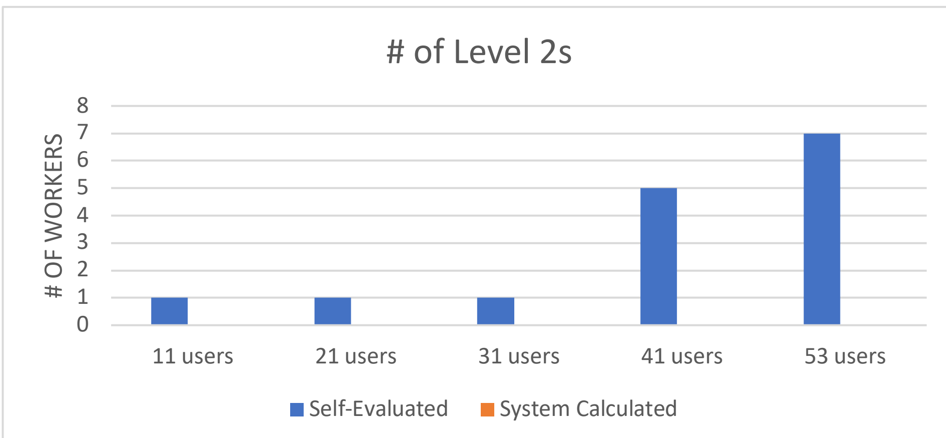

Fig. 39. Bar graph showing worker's self-evaluated competence vs system calculated competence at competence level 2.

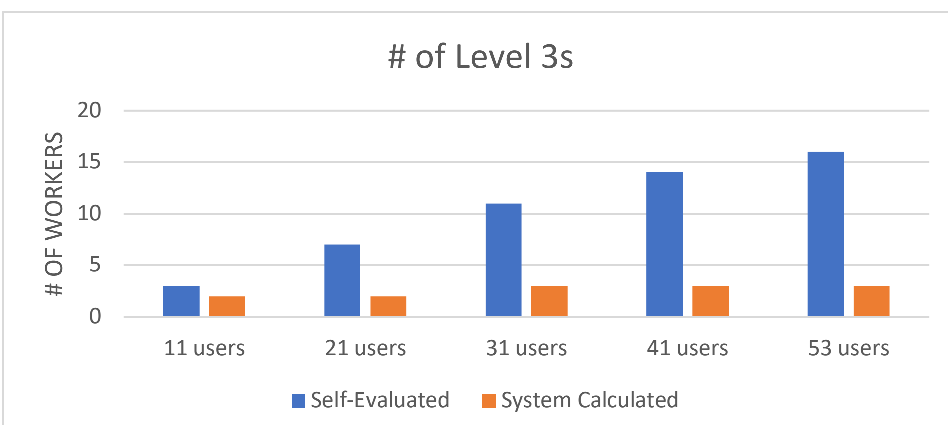

Fig. 40. Bar graph showing worker's self-evaluated competence vs system calculated competence at competence level 3.

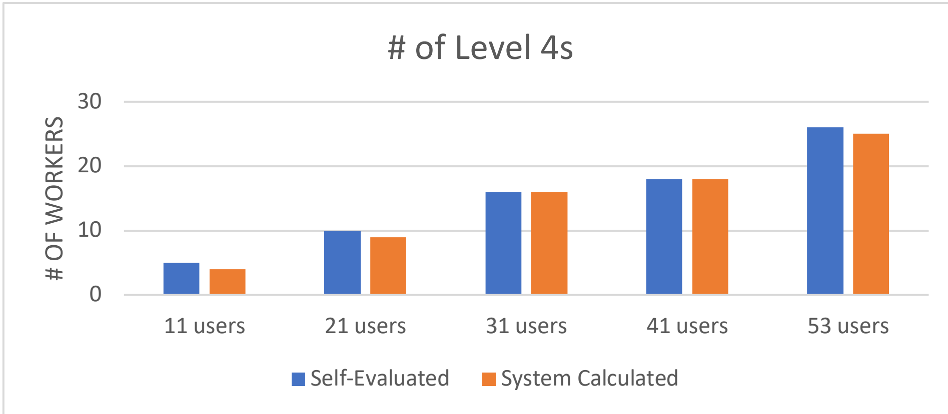

Fig. 41. Bar graph showing worker's self-evaluated competence vs system calculated competence at competence level 4.

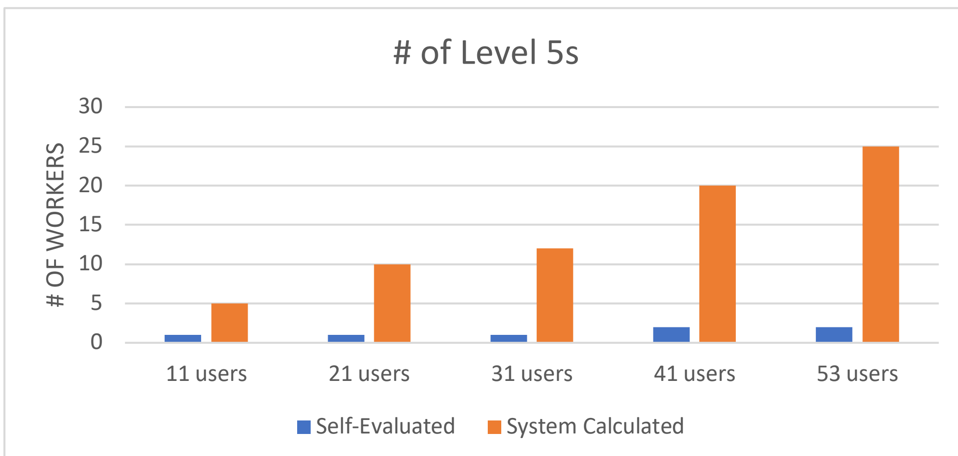

Fig. 42. Bar graph showing worker's self-evaluated competence vs system calculated competence at competence level 5.

A conservative community evaluation was observed in a performance (P) to self-evaluation ratio. This community is perceived as being *overly critical* of its own capability and has negatively represented itself by a mean CCI of -21.51% (Figure 43).

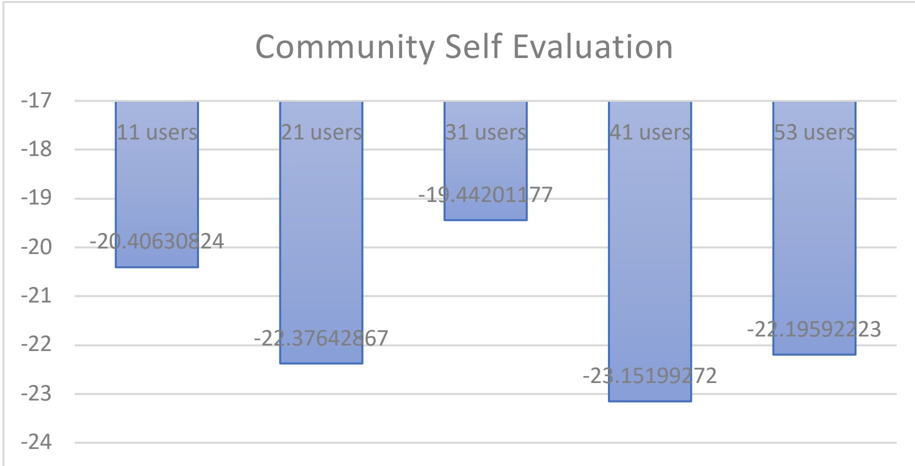

Fig. 43. Bar graph showing community's self evaluation as workers complete jobs over 5 iterations.

For each simulation step, the recommender's ability to adapt to changing data was evaluated. Changing aspects of data included the increasing worker pool, the number of completed jobs and the corresponding feedback from the employers for completed jobs. The collaborative filtering algorithm iteratively updated weights for job features and improved with increments of new data. The recommender consistently recommended jobs to workers with an average of 86.175% similarity to jobs previously completed (Figure 44).

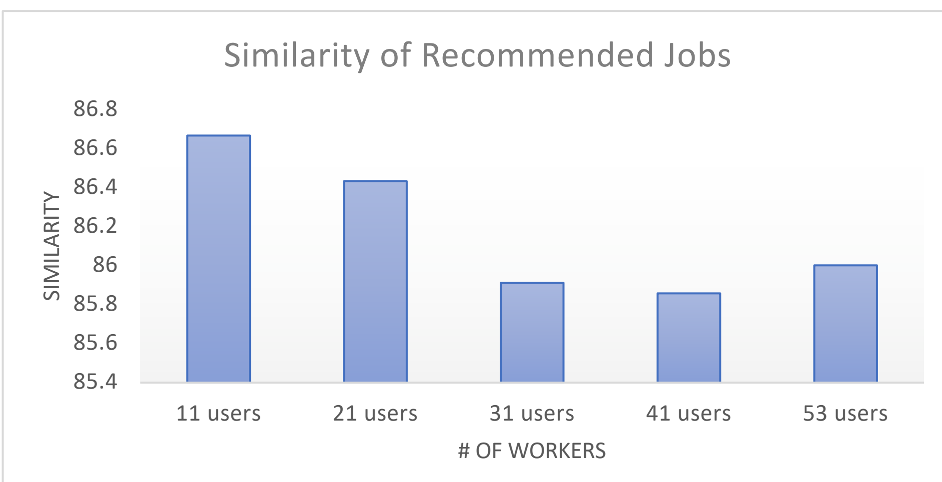

Fig. 44. Bar graph showing recommended job similarity as workers complete jobs over time.

Figure 45 illustrates score predictions for random jobs for 10 select human workers and the machine worker. Each line represents the recommendation score for a job for a specific user, over the 5 iterations; the jobs are different across users. On each iteration, the training set increased as workers are introduced to the worker pool and they complete jobs. Recommendations show continuous improvements on each iteration. The improvements are a reflection of a growing training set for the recommender to make its evaluations and make better predictions against the background of more data.

4.3.6 Conclusion. Using metrics such as the CCI and CPI, facilitates employers seeking crowdsourced labor to collectively understand the nature of a given labor force. In addition, platform owners can use these metrics for advertising the labor capacity of the workforce. The CCI is a collective indication of the work force's performance capability based on actual performance, the CPI is the general community's consensus of themselves based on their individual self-evaluations. With this type of information, employers can understand the true nature of the capabilities of workers in a labor force relative to another. This approach produces a more informed cross-sectional view into the capabilities and worker perception of their own skills in a labor force, employers can adjust their levels of worker confidence and opt whether to crowdsource their tasks through the platform or seek another with a higher CCI index. Negative CCI ratings result in lack in confidence and under representation of worker capabilities. In contrast, positive CCI ratings result in over confidence and over representation of worker capabilities.

To directly address the research question, evaluative metrics can allow for the evaluation of collective capabilities of a worker pool. Self-evaluation of competence is not a valid measure for employers to use to determine the actual performance of workers. Self-evaluated competence can be misrepresented if the worker actually does possess some competence or may be completely false. In the case of this community, their potential was under-represented when self-evaluation was used as the measure for competence.

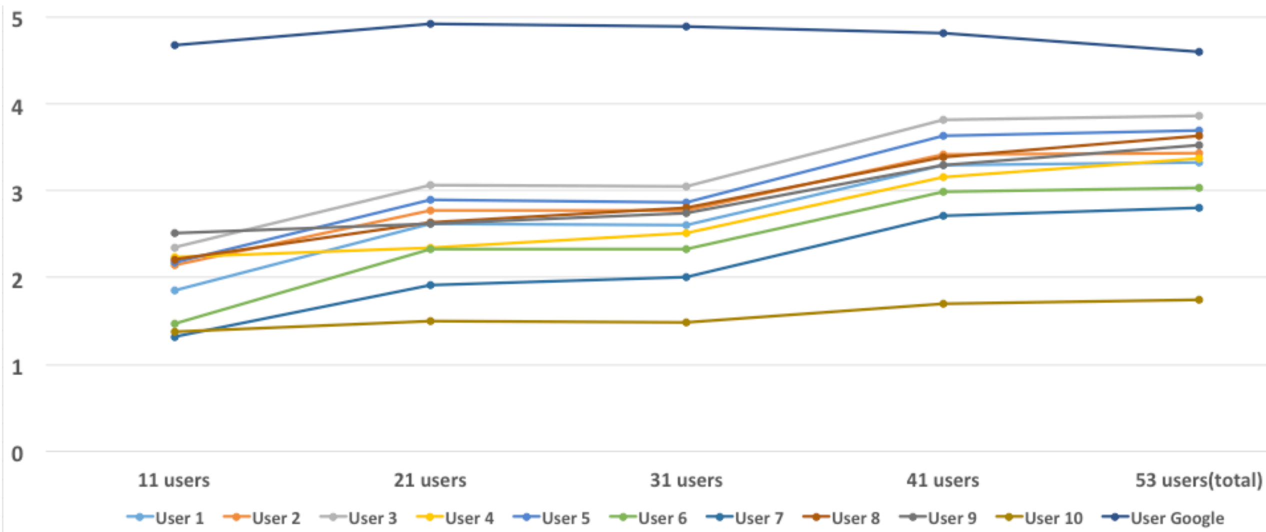

Fig. 45. Line graph showing change in recommender prediction accuracy for a selection of workers as jobs are completed over time.

4.4 Study 4 – Impact of Increasing Data (Platform Growth) on Recommendation Times

Simulated data was used to assess the impact of increasing volumes of data on the recommendation times of the collaborative filtering recommender engine. This study seeks to answer **part 3 of Research Question 2**:

Is there an operational approach that facilitates **recommendations as a function of changing environment data** (*i.e. jobs, labor pool, and workers' performance*)?

To further scope this research question, this study was designed to assess the impact of platform growth on making recommendations. In this experiment, platform growth will be scoped as increasing volumes of worker pool and job offering data. The research question was further scoped with the following objective:

- *O1 – What impact does increasing volumes of data have on the collaborative filtering based worker-job recommender respond? This behavior is measured with respect to time to yield recommendations vs. magnitude of the data.*

To assess the impact an increasing worker pool and entry rates of new jobs to the system has on recommendations, the completion times for each simulation were recorded and graphed. The simulations were sandboxed to standalone computer with 16GB of RAM and a 2.9 GHz Intel Core i5 processor with two cores.

4.4.1 Data and sources. This experiment consists of two simulations. In both simulations, the 300 job instances from Study 2 were reused. The results from Study 2 produced a 68-characteristic binary feature set across all instances of jobs. Worker data however was synthesized. This study consisted of two simulations, the first maintaining a worker pool of 1000 workers, and an initial job bank of 30 jobs increasing in increments of 30 through to 300 jobs in each simulation step. The second simulation maintained a bank of 300 jobs with an initial worker pool of 100 workers increasing in increments of 100 through 1000 to workers in each simulation step. For both simulations, workers were randomly assigned to jobs and their scores were also synthesized.

4.4.2 Results. The results in this section shows the impact of increasing volumes of data on the time it takes to make recommendations. It shows the impact of a fixed worker pool and an increasing availability in jobs; it also shows the impact of with a fixed number of jobs and an increasing worker pool. The results demonstrate that the system, in context of a somewhat limited simulation environment, responds favorably to increasing volumes of data.

Each simulation consists of increasing volumes of data and the completion times for the engine to make recommendations. Scatterplots illustrate the engine’s completion times in seconds, in calculating predictions and making recommendations as the volume of data increases. The scatterplots show that completion times increase linearly and proportionately to the magnitude of the data as simulations progress.

Table 6 shows the data for the job study. Each simulation step reflects an increasing number of jobs by an interval of 30, versus the time the recommender took to calculate predictions for 1000 workers. Figure 46 illustrates a scatterplot with the engine's completion times in seconds, in calculating predictions and recommendations as the volume of data increases; it shows that processing time increases proportionately to increasing magnitudes of the data.

Table 6

Simulation Data for Impact of Increasing Job Offerings – Job Simulation

# of Jobs	**Completion Time (secs)**
30	550
60	599
90	770
120	914
150	1530
180	1703
210	2011
240	2274
270	2638
300	3034

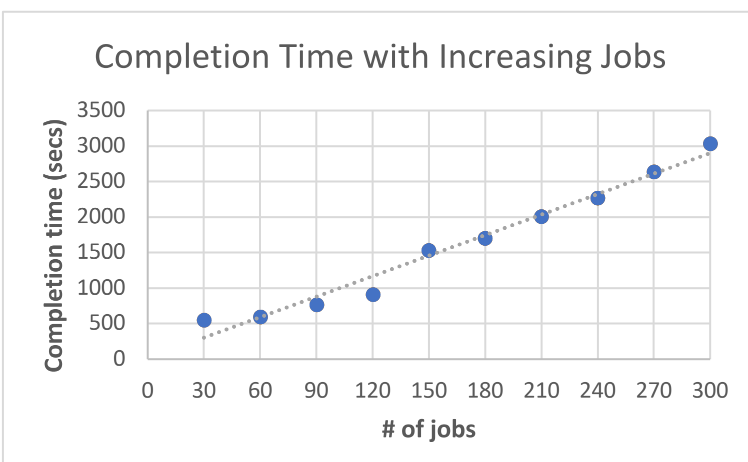

Fig 46. Scatterplot showing completion times vs. increase in jobs and fixed worker pool.

Table 7 shows the data for the worker study. Each simulation step reflects an increasing number of workers by an interval of 100, versus the time the recommender took to calculate predictions for 300 jobs. Figure 47 illustrates a scatterplot with the engine's completion times in seconds, in calculating predictions and recommendations as the volume of data increases; it shows that processing time increases linearly and proportionately to increasing magnitudes of the data.

Table 7

Simulation Data for Impact of Increasing Worker Pool – Worker Simulation

# of Workers	Completion Time (secs)
100	31
200	82
300	170
400	308
500	644
600	847
700	1154
800	1474
900	1944
1000	2420

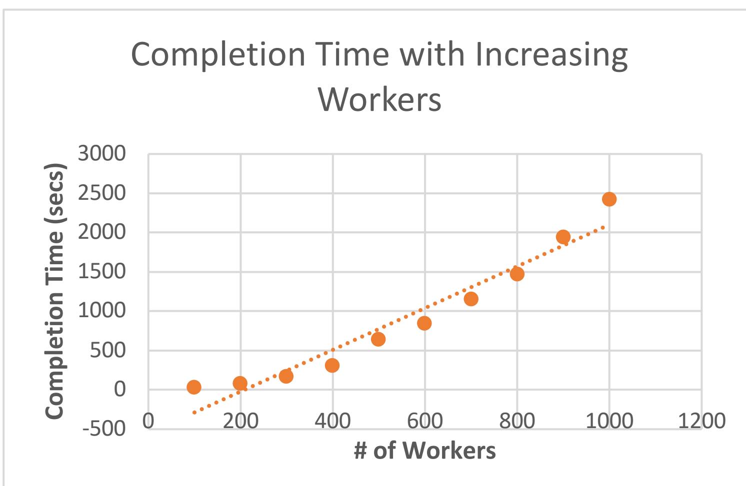

Fig 47. Scatterplot showing completion times vs. increase in worker pool and fixed job catalog.

4.4.3 Conclusion. Figure 48 shows a comparison of both scalability studies superimposing the curves from the 2 scatterplots. The graph shows that a larger worker pool has more of an impact on the time it takes to make predictions as opposed to a smaller worker pool with a higher availability of jobs. Future work will include an experiment designed to assess the performance and scalability of the recommender engine through the analysis of the complexity of its underlying algorithms and processes.

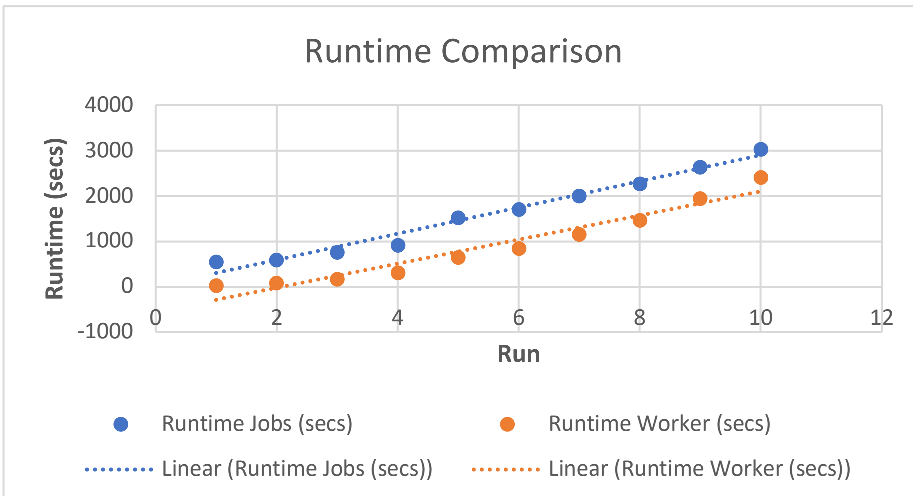

Fig 48. Scatterplot showing performance times vs. increase in volumes of data.

4.5 Study 5 – Recommendations as a Function of Changing Data

This experiment observed the calculated recommendations as a direct result of the recommender engine's ability to make recommendations as a function of rapidly changing data. This study seeks to answer **part 3 of Research Question 2**:

Is there an operational approach facilitates **recommendations as a function of changing environment data** (*i.e. jobs, labor pool, and performance*)?

The study is further scoped and guided by the following objective:

O1 – In making its recommendations, is the approach sensitive to responding to changes and evolving data? Evolving data within this context is defined as an increase in job catalog and an increasing workforce.

Recommendations were observed from two perspectives, first from a fixed sized workforce and an increasing availability in jobs; next, from a fixed number of jobs and workforce increasing in size. To further understand the effect of worker performance on a recommended job, predicted scores and the similarity in characteristics of jobs completed to recommended jobs, several case simulations setup as outlined in sections 4.5.3 through 4.5.4.

4.5.1 Data and sources. The 300 instances of jobs from study 2 along with the resulting 68-characteristic feature set was retained for this experiment. Worker data however was randomly and systematically synthesized in a pseudo-random manner. Job and worker data were formatted in like manner to the matrices illustrated in Listing 3.

4.5.1.1 Job simulation. This configuration consisted of a worker pool of 1000 workers with an incremental number of jobs, 30 through 300 in increments of 30.

4.5.1.2 Worker simulation. This configuration maintained a job bank of 300 instances with incremental number of workers, 100 through 1000 in increments of 100.

4.5.2 Case simulations. To illustrate the recommender's response to increasing volumes of data, continuous evaluation of the worker's performance relative to others in the workforce completing similar jobs, cases are presented that measure the responsiveness of the collaborative filtering based approach as a function of changing data. Data are introduced incrementally in an increasing fashion. The cases show how evolving data affects recommendations of a job, over 10 simulation steps for a selection of 10 random workers. Job performance is scored using a 10-point scale, with 10 being the highest achievable score.

4.5.3 Individual case simulations. To illustrate the effect the recommender can have on an individual's recommendation, a case is presented illustrating how recommendations for a job evolves for a worker in contrast to actual performance. The cumulative evaluations from all individual workers are used in system wide calculations.

4.5.4 System-wide simulations. Each simulation consists of the system wide average similarity of recommended jobs followed by the system wide average predictions in contrast to the average performance in similar jobs that were already completed. This gives insight into the overlap in similarity of the jobs already completed to the jobs being recommended.

The study closes with a comparison of the system wide predictions and performance in both simulations. More specifically, it compares the effect a growing work force with full job catalog has on recommendations to that of a full workforce compliment to a growing job catalog.

4.5.5 Evolving recommendations. Figure 49 illustrates a case from the job simulation. The case observes the recommendation of a job over 10 simulation steps; this job will be referred to as "testjob1". Given consistent performance and selection of similar jobs by six workers, testjob1 was recommended over the entire simulation. The recommender's prediction for testjob1 for the six workers sustained a score of 8 or higher; this suggests that the six workers would have a higher possibility of scoring well in the job given performance in their work history.

The remaining four users had lower frequencies of recommendations for this job. It was only recommended once (10% of the time) for worker D1; this worker engaged and scored well other types of jobs which altered the testjob1 being included in the recommended list despite high predictions (Figure 50). The job was recommended to workers M19 and X4 with frequency of 20% of time, notably the first two steps of the simulation. The testjob1 was subsequently removed from the list as the workers engaged in other types of tasks with little overlapping characteristics.

Average performance in similar jobs also played a role in testjob1 appearing in the recommended list; it was replaced by other jobs where their predicted chances of performance were higher.

Testjob1 was recommended to worker A1 40% of the times (Figure 50). It was initially recommended for the first three iterations and then subsequently removed from the list while the predicted chances of other jobs were higher despite consistently performance for the current job. As the chances for newly recommended jobs decreased, testjob1 returned to the recommended list for simulation step 7 then was subsequently removed (Figure 49). Testjob1 was recommended to the workers for an average of 69% of the time with an average prediction of 9.18.

4.5.6 Individual evaluations. This individual evaluation shows assessments done for worker A1 across all 10 simulation steps, Figure 51 shows a higher predicted score for recommended jobs versus the worker's average performance score in similar jobs. The worker's average performance score in similar jobs was 1.29 mean points lower than the predicted average score for recommended jobs (Figure 52). The worker had a higher chance of performing recommended jobs and others like them that were consistently engaged. Despite the slight degradation in the average performance in similar jobs (Figure 51), the prediction scores for recommended jobs remained relatively consistent due to the community-wide evaluative nature of collaborative filtering. When compared to the performance of other workers in the community engaging in jobs with overlapping characteristics, their scores were on average 8 and higher.

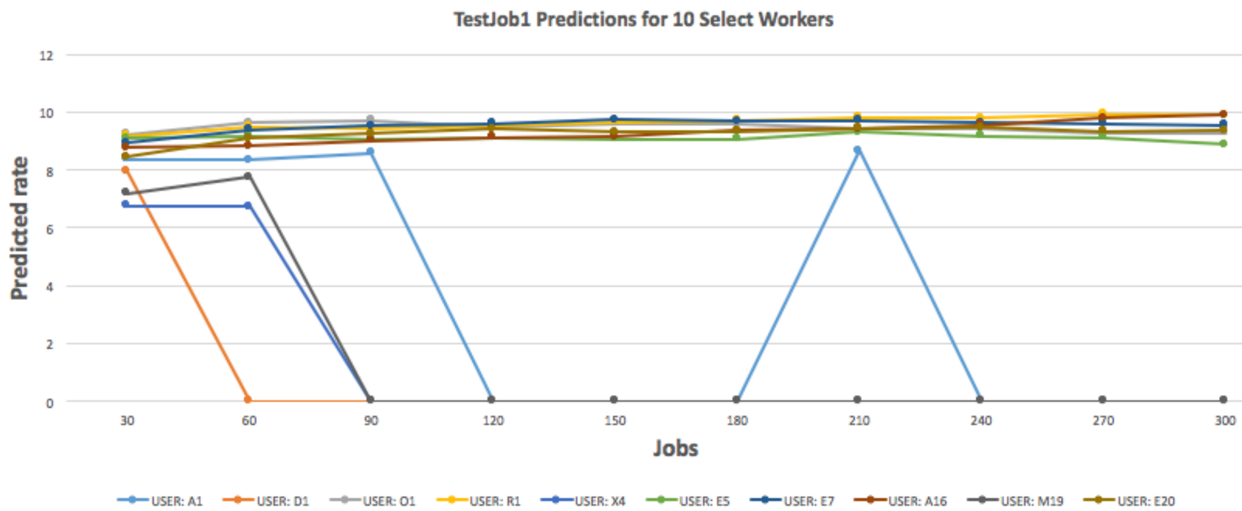

Figure 49. Line graph showing recommended scores for testjob1 over 10 job simulation steps.

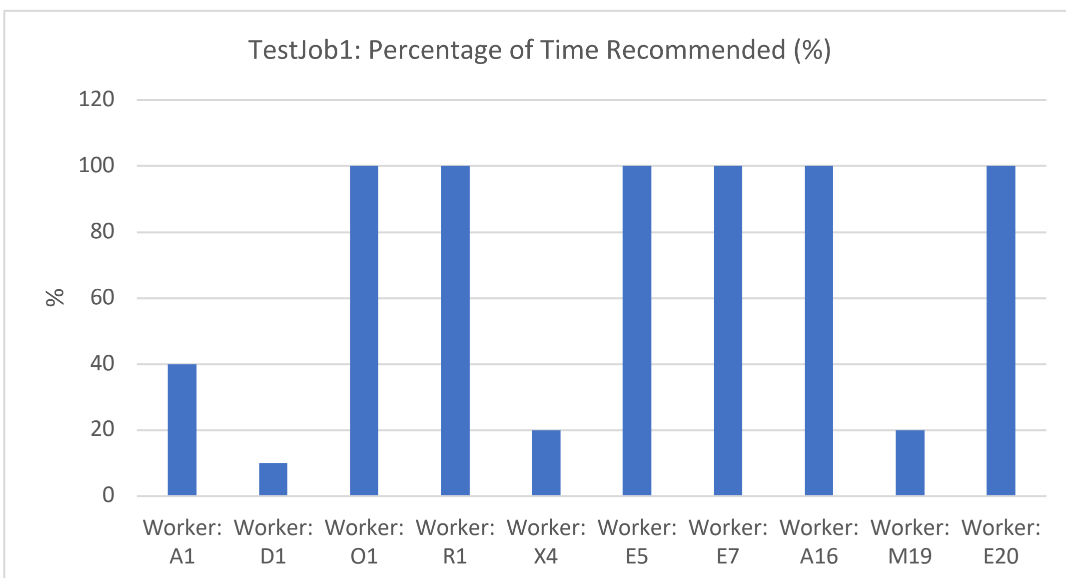

Figure 50. Bar graph showing the number of times testjob1 was recommend to each worker.

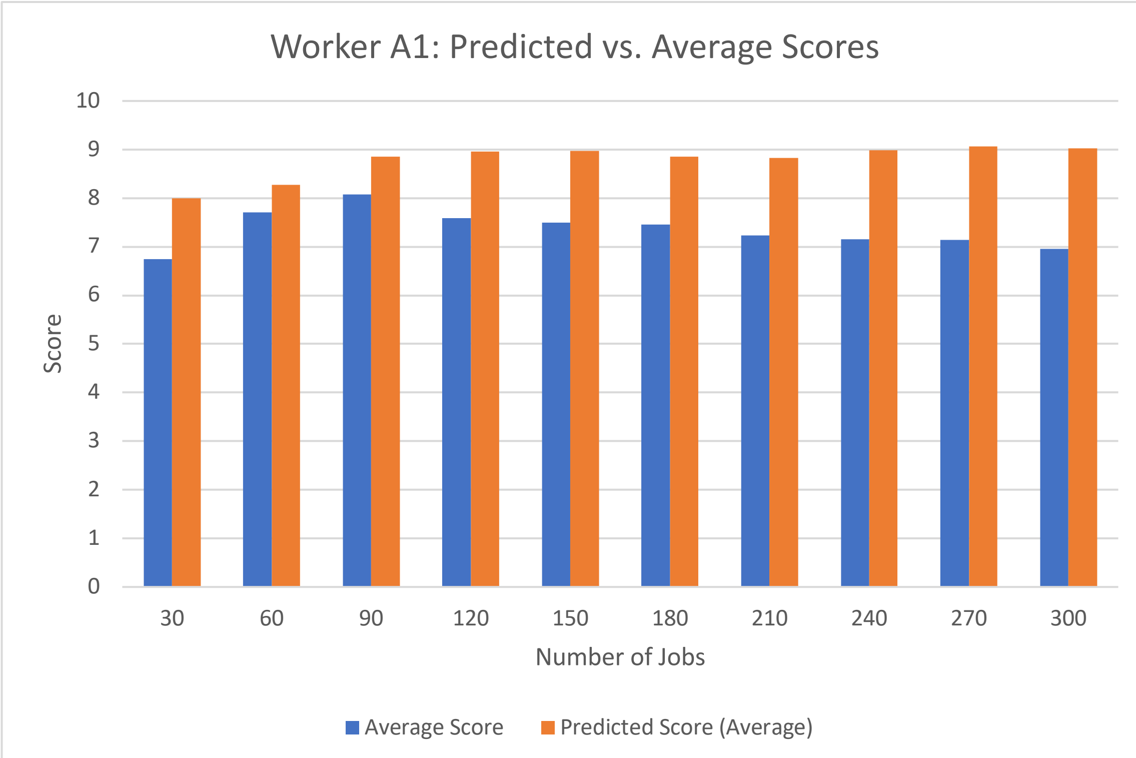

Figure 51. Bar graph showing predicated and average scores for worker A1.

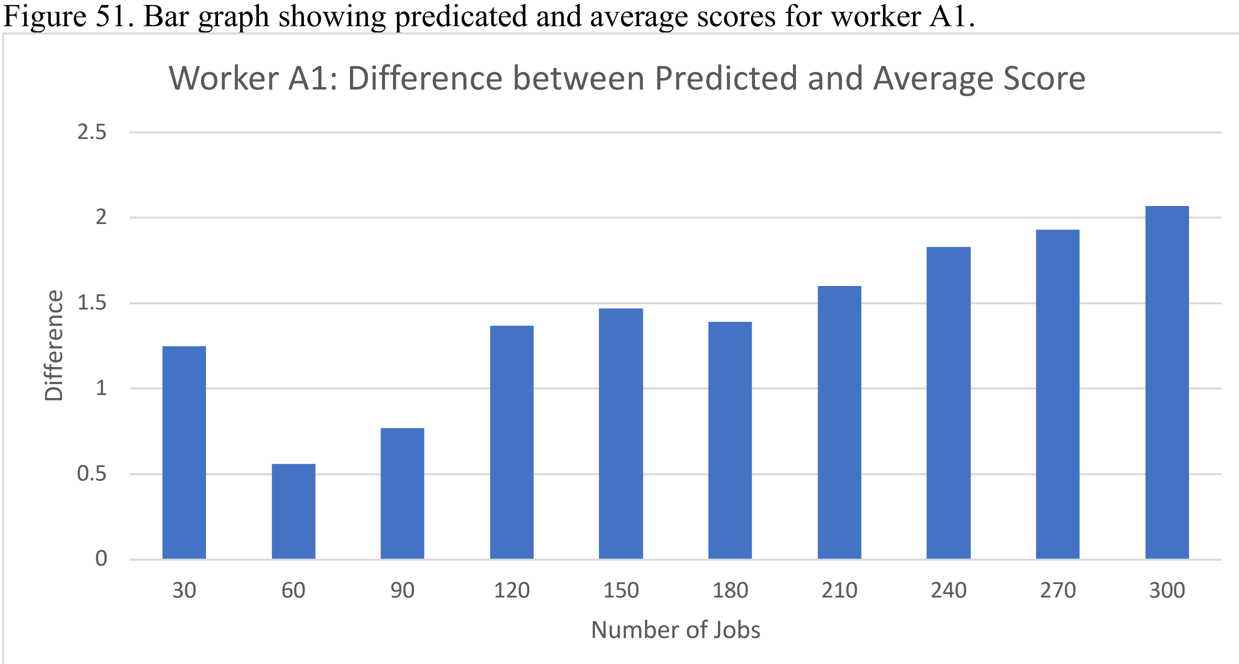

Figure 52. Bar graph showing the difference between predicted and average scores for worker A1.

4.5.7 Scores and predictions. Figure 53 contrasts the community wide average scores for jobs completed against the average predictions for recommended jobs, over 10 recommendation simulation steps in the job simulation. As more jobs became available introducing more options for worker engagement and for recommendation, the general performance of the workforce slightly increased. General performance was also higher than predictions when the job catalog was less than half of its compliment. Predictions however saw more sharp increases as more jobs were completed, including those with and without overlapping features. Upon the job catalog reaching half of its compliment, the predictions surpassed and remained slightly above the average performance.

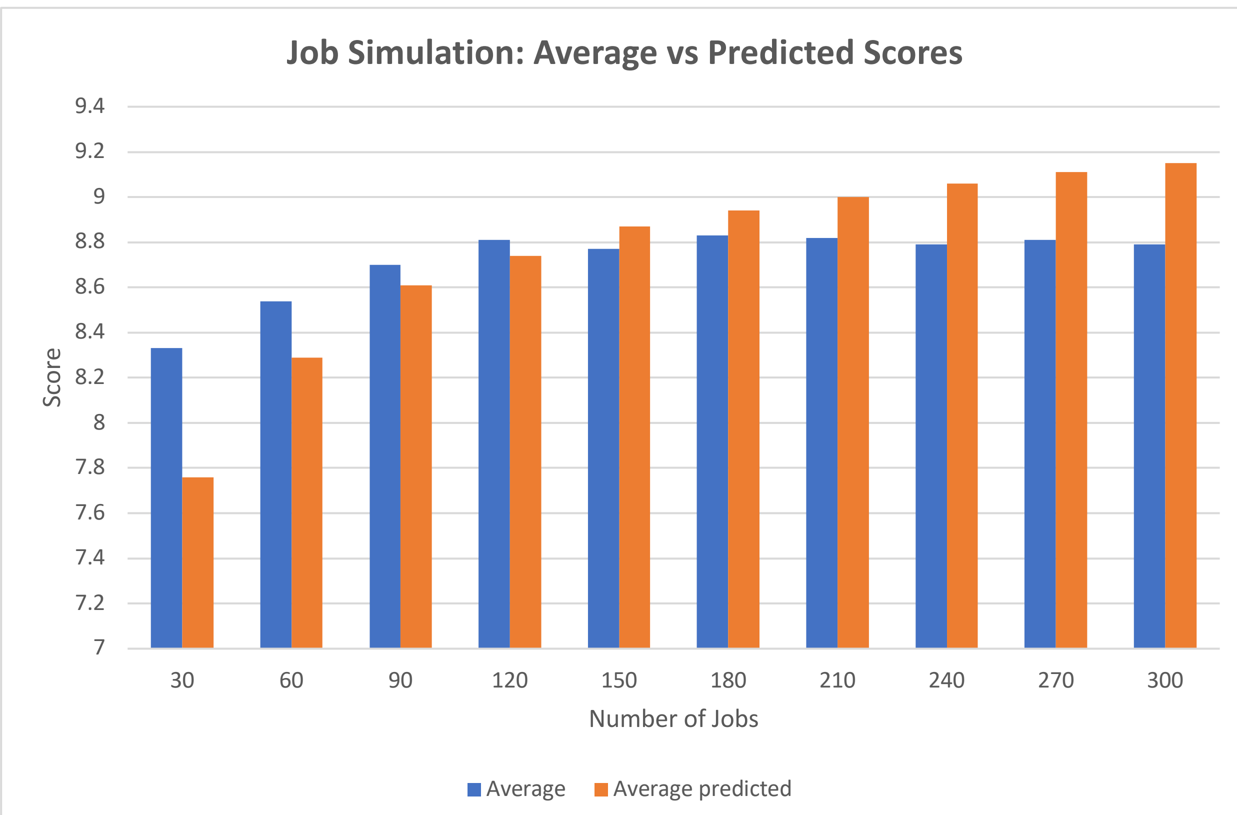

Figure 53. Bar graph showing average and predicted scores for the job simulation.

Figure 54 compares the community wide average performance score for jobs performed to the average predictions for recommended jobs in the worker simulation. Both the ratings and predictions had graceful increases throughout the iterations; an indication that improving performance impacts and increases predictions. The predictions for recommended jobs were on average 0.04 points, marginally higher than the performance in jobs already completed. This difference can be attributed to variety of circumstances. Workers are not bound to do jobs that were recommended and have the freedom to complete others not in the list. In addition to this, worker predictions are estimates based on previous performance. Subjectivity in employer evaluations, worker effort with respect to precision, time spent on job and other constraints can potentially affect actual outcomes and do not translate exactly to predictions.

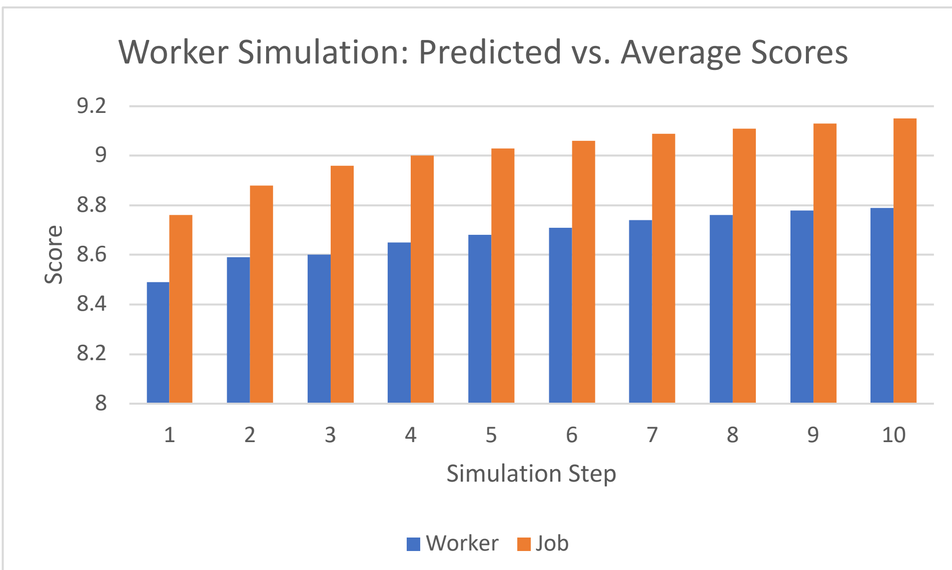

Figure 54. Bar graph showing average and predicted scores for the worker simulation.

In comparing the average predictions for both simulations, Figure 55 illustrates that predictions for the job simulation experiences higher levels of predictions in earlier steps with small increments of improvement as it progresses to the latter steps. In the case of the worker simulation, the initial prediction was measured 1 unit below the initial prediction for the job simulation. Earlier worker simulation steps show predictions improvements as the workforce grows; this is a natural result of more data to drive community-based evaluations of the collaborative based recommender. Convergence is observed as both simulations progress; however, is most prevalent in the final 5 steps as both scenarios gain identical characteristics with respect to worker and job datasets.

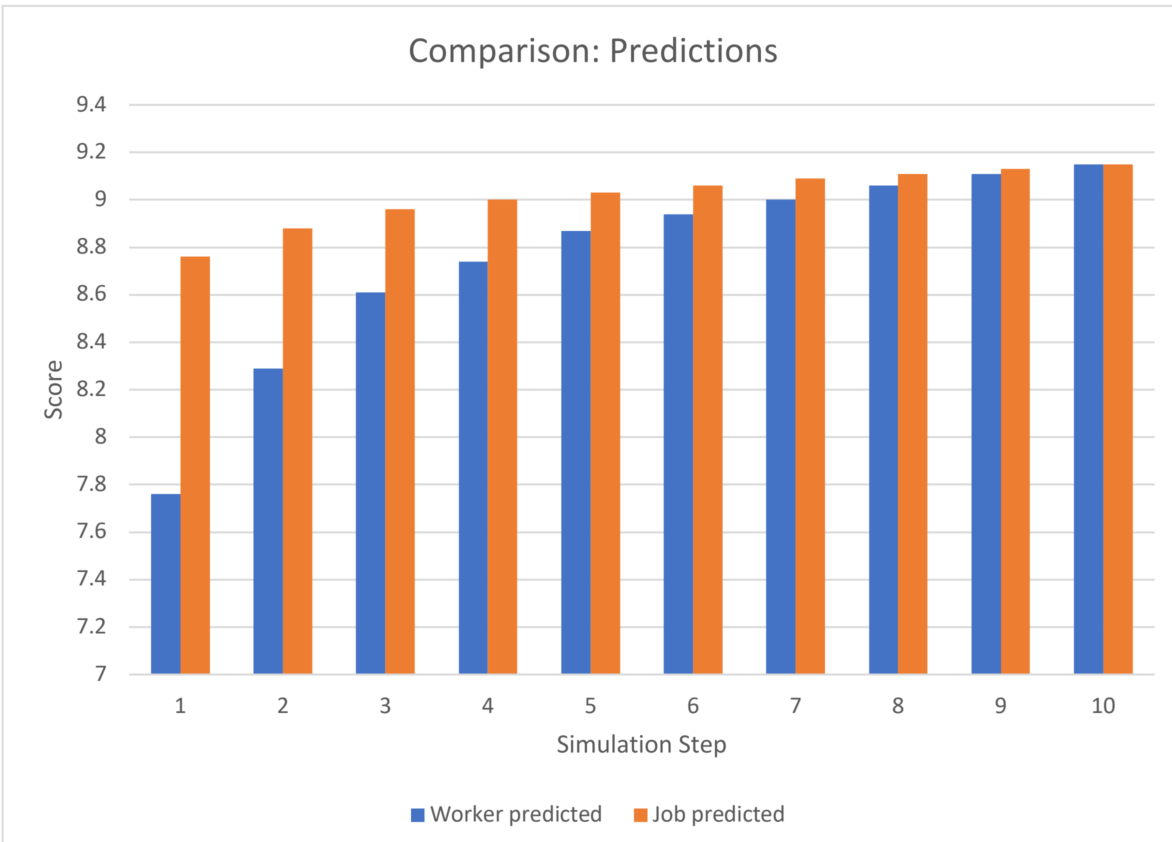

Figure 55. Bar graph showing predictions for the both simulations.

4.5.8 Similarities. Figure 56 shows the average total similarity for jobs for all 1000 workers in the job simulation. With smaller amounts of available jobs (≤120), the similarity index consistently remained at 100%. As the number of jobs increased, with each introducing new characteristics and with increasing uptake in jobs by workers, the similarity index trended downwards in a stepped, slow graceful manner only changing by an average rate of 0.016% per iteration.

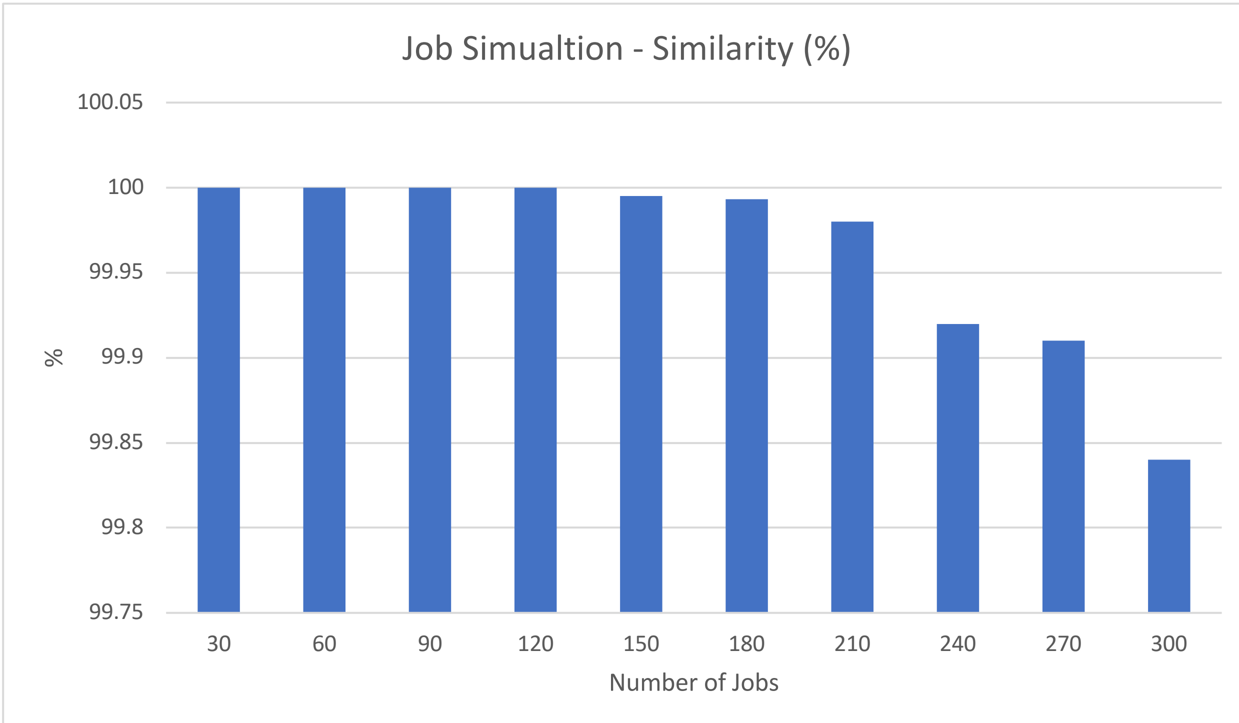

Figure 56. Bar graph showing similarity in completed and recommended jobs in the job simulation.

Figure 57 shows the average total similarity for recommended jobs in the worker simulation. With a smaller labor force, the similarity index consistently remained closer to 100%. As the labor force increased to 300 workers with an uptake in jobs, the similarity index sharply decreased however trended relatively stable by +- 0.02% for the remaining iterations. The recommendations changed by an average rate of 0.017% per iteration.

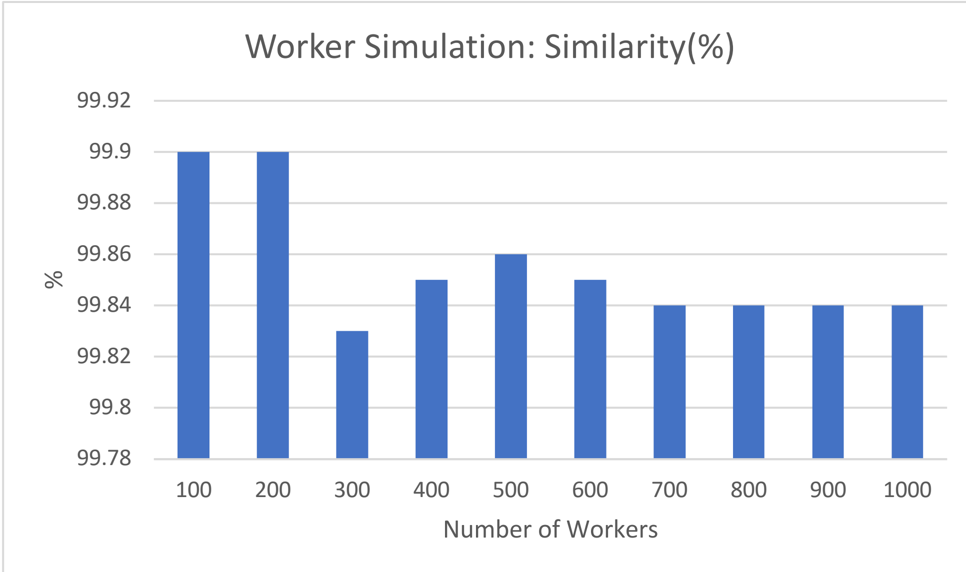

Figure 57. Bar graph showing similarity in completed and recommended jobs in the worker simulation.

In comparing the similarity in characteristics of jobs completed to the top 10 recommended jobs across all workers, the job simulation lead to recommendations with 100% similarity in early simulation steps (Figure 58). This is due primarily to the small job catalog available to the larger workforce; recommendations are fewer and limited to the small offerings in jobs where in some instances bear no similarities to other jobs. As the job catalog increases, the similarity decreases gracefully which allows for jobs bearing some similarity but also new characteristics.

The worker simulation had a larger initial selection of jobs allowing for earlier, more diverse selections in recommendations. Workers are able to better align themselves with jobs best suited to their expertise. As the workforce grows, there is an increase in jobs completed resulting in more data for the recommender to perform more comprehensive community evaluations. This also influences the earlier introduction of new types of jobs bearing some characteristics to those being performed.

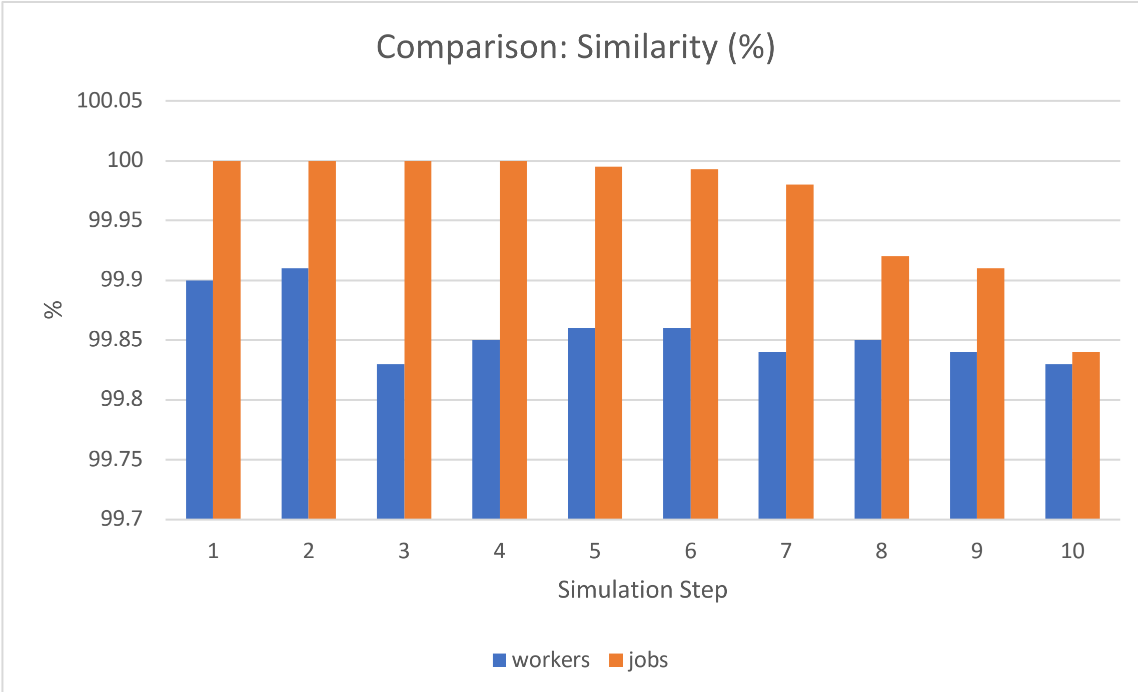

Figure 58. Bar graph showing similarity in completed and recommended jobs for both simulations.

Figure 59 shows that after 10 simulation steps, the performance of the labor force in both studies converge where the configurations of the labor force and catalogs become identical; however, there was little difference, on average +- 0.04 points in general performance over both simulations.

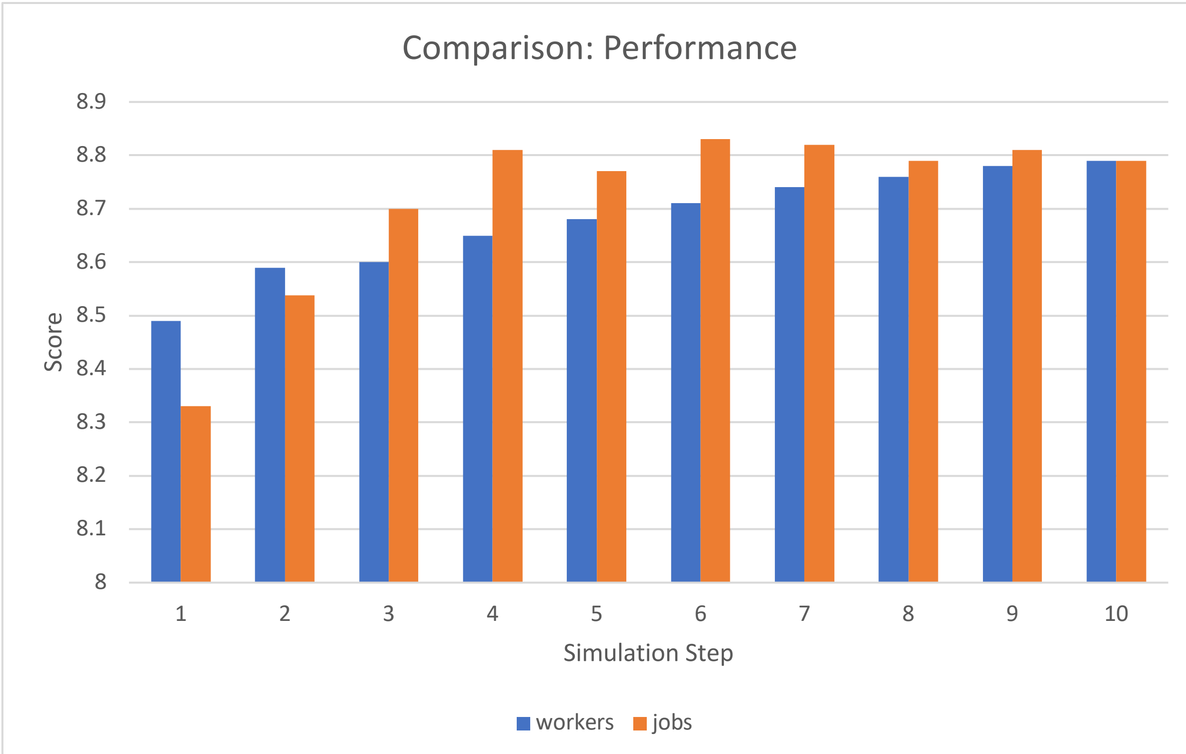

Figure 59. Bar graph showing performance for both simulations.

4.5.7 Conclusion and Comparison of Studies. In comparing the trends from both studies, having a full catalog of jobs available for a developing labor force yielded better overall performance in very early iterations. A larger labor force with fewer selections in jobs resulted in lower performance with workers engaging in jobs available with less opportunity to being aligned with their expertise. Inversely, workers experience increased flexibility to engage in jobs more aligned with their expertise from a wider initial selection in jobs; resulting in better performance (Figure 59).

A developing labor force with a full catalog of jobs by the third iteration, shows a more favorable overall trend when compared to having a full labor force with a developing catalog of jobs. As more jobs become available increasing the catalog, workers gain the opportunity to engage in a more diverse selection of jobs including those more aligned with on their expertise; this leads to sharp increases in performance (Figure 59).

5.0 Conclusion, Threats, Future Work, Outlook

With the advent of human centered computing, proliferation of smart mobile devices, pervasive wireless connectivity and Web 2.0 technologies, crowdsourcing is a modern and popular mechanism used to distribute work over the Internet. Using an open call, a passive recruitment strategy, different platforms solicit labor from identified or anonymous individuals. Systems have been constructed with integration of machine computing elements and human computing elements via crowdsourcing, primarily in cooperative configurations. This body of work has identified deficiencies in these types of systems and has demonstrated strategies the address the same. This body of work addressed the following research questions:

Research Question 1

When managing crowd computing resource spanning human and machine workers, what general information models effectively define:

- functional, non-functional and evaluative concerns for both types of workers, and
- a wide cross-section of tasks from diverse problem domains?

Research Question 2

Is there an operational approach that enables systematic and reliable:

- Delegation of work across human and machine work resources
- Metrics that allow for the evaluation of collective capabilities of a worker pool
- Recommendations as a function of changing environment data
    - Jobs
    - Workforce
    - Performance
- On-demand modes of operation within a service driven infrastructure

5.1 Mixed Cooperative, Human Only Competitive and Mixed Competitive Models

Most work in literature currently outlines a human only competitive model. The approach outlined in this work supports a competitive model that is independent of the type of worker, thus supporting human only, machine only and mixed competitive models. While the literature does support mixed workers, the workers are engaged in a cooperative workflow performing distinct tasks at specific milestones. The literature describes mixed co-operative systems focused on load balancing and the scaling of workers at their respective points of labor.

5.2 Open Call vs. Open Push-Pull

Crowdsourcing is currently modeled on the open call model; this has also been cited as a major challenge that directly affects the prospects of recruitment and labor force retention. Tasks are placed online and their owners wait for prospective candidates to take up the offerings subjected to the task selection process; this results in a passive type of recruitment. Job offerings are also limited to a specific platform where the workers in the labor force must all subscribe to gain access to them. This body of work outlines the Open Push-Pull model that addresses the current issues in the current implementation of the crowdsourcing paradigm. Through an open pull, job and worker data are pulled in from diverse organizations and community oriented repositories respectively. Using a worker-job matching approach, jobs are assessed and matched to prospective workers; jobs are pushed through communication channels to workers in their existing online circles and communities. This constitutes an active form of recruitment as opposed to passive recruitment with the current structure of traditional crowdsourcing. Active recruitment increases job awareness to individuals in online circles and communities who may potentially convert to workers, hence increasing the capacity and potential of the labor force.

5.3 Top Down Approach vs. Bottom Up Approach

Most approaches in literature utilize a top down approach to making recommendations. It is typically centered around the professed profile, skillsets and interests of the worker which influences the recommendations made to the worker. Current approaches including machine-learning enhanced, probabilistic and semantic approaches, work under the assumption that information workers' profiles are accurate both in terms of facts and worker's own awareness of their competencies. Pick-a-Crowd uses some form of push architecture however the SOA proposed differs in many ways. Worker-job recommendations are semantically influenced by social media interests, driven by concepts liked by the worker on Facebook (Difallah, Demartini, & Cudré-Mauroux, 2013). Tasks are also limited to those found on Amazon Mechanical Turk as opposed to a cross platform approach supported in the reified object model and supporting SOA presented in this work It also uses a top-down approach using worker's social media profiles to make recommendations given concepts that are professed as liked. This work uses a bottom-up approach making recommendations based on actual competence with provisions such as the worker's elastic index for a specific job.

This body of work utilizes a bottom up approach that makes recommendations based on workers' track record in performance. It is inferred that a worker has ascertained some level of mastery in a skill, given an assessment on historical performance in jobs requiring them. This inference is represented as a score he or she is likely to obtain if assigned a new job requiring the same or overlapping skillsets. This approach gives employers more insight in understanding the competencies available of a labor force for given jobs. It also provides performance metrics that allow for the general competence and performance of individual workers and for the labor pool at large. Using this information, employers unveil self-awareness from the workers versus actual mastery of skills based on the objective evaluation of other employers and task owners. It also shows where individuals and the wider labor pool have under-represented or over-represented their competencies.

5.4 Pre-defined Classes vs. Reified OO Modeling

Approaches exist that combine machine and human workers within a workflow using a cooperative model (Candra, Truong & Dustdar, 2013). They involve design time constructs and metrics and their translation to classes at implementation. The Elastic Profile was used as a metric to assist with scaling of machine and human computing elements at their respective phases in a workflow. The computing elements were defined using Backus-Naur Form (BNF) grammar, then translated to classes at implementation (Candra, Truong & Dustdar, 2013). The approach however does not support the dynamic introduction of new metrics and characteristics for jobs at runtime, rather depends on the definition of those metrics by a domain specific set of rules governing the workflow. This approach however does not provide flexibility for the dynamic and run-time addition of new characteristics and skills to the profiles of the workers. It also does not support competitive mixed worker models.

To address the issues above, this body of work uses a reified model to achieve the needed flexibility for environments with rapidly evolving data. The reified approach enables flexibility through the abstraction of both humans and machines under an ADT which is used for the evaluation of workers. The reified model also extends itself to jobs from diverse sources, supporting editable and extendable characteristics in description of jobs to offer more detailed and granular evaluation. This flexibility in the data model allows for a wide support of data from disparate sources which also enables system interoperability.

5.5 Semantic Approach vs. Collaborative Filtering approach

Semantic based approaches have been used to make recommendations primarily using linked data, relationships, subject-predicate-object triples and drawing inferences. Recommendations work on the assumption that the existence of a predicate between a subject and object, qualifies the subject to know other objects in the same category of the object in existing triple. Semantic recommenders, though may yield instances of success, operate on a flawed premise that the subject has other predicate affiliations with other objects in the class. Semantic recommenders are too general and do not offer fine grained analysis for recommendations based on multi-character set job. Some use a system of voting to aid in ranked recommendations based on information gathered from multiple triples; subjects are ranked by the frequency of associations they have with objects in the class where the association with some objects may carry more weight than others. The recommendations are also based on a top down analysis based on assertions (objects) provided by the worker, who is the subject. As outlined in the previous subsection, flawed professions lead to flawed recommendations, semantic recommenders are also susceptible to the same.

This body of work utilizes a collaborative filtering based recommender that uses a bottom up approach. It infers that workers possess the requisite skillset given their performance on a job requiring the skillset. Using this approach, jobs can be characterized using a large feature set which when combine with the reified object-oriented model, supports flexibility, and the real-time extensibility of new features for granular descriptions. The collaborative filtering approach also allows for a comprehensive community-wide evaluation which does not limit recommendations to hard, existing associations as in approaches using linked data and semantic based triples. The collaborative filtering approach also responds well to rapidly evolving data including but not limited to fluctuating individual and collective worker performance, job uptake, increase in job offerings and an expanding or shrinking labor pool. Semantic based triples tend to be more long term and do not consider user trends in performance when making recommendations; instead, it focuses on professed information embedded in the triples.

5.6 Threats

This body of work is threatened by a community wide cold start in data. To make recommendations, there needs to be at least one completed job in the system. Recommendations are also not as accurate with small amounts of completed tasks; they are quickly refined within a relatively small amount of iterations as workers take up more job offerings. Cold starts can be resolved by using other traditional top-down approaches until a system defined number of jobs are completed. The system however can handle a cold start for an individual worker with no work history, by applying a mean normalization calculation of all jobs performed by all other workers in the system. This then makes recommendations based on the jobs that are the most frequently with the greater possibility of yielding higher scores.

5.7 Future Work

Prospects and iterations of this work include evaluating the system using social computing scrutiny, human constraints, sentiments and influences that may impact further refinements needed for the recommender. Other approaches will be combined for recommendation including but limited to establishing associations amongst job skillsets and the clustering of workers who exhibit similar skillsets and interests in jobs. This will be done with the intent to build specialized groups within the workforce to prioritize job recommendations to these groups of workers.

5.8 Outlook

More approaches will be implemented to migrate systems from the open call model, subscription models, passive crowd recruitment and platform lock-in constraints to more open interoperable systems facilitating integration through interoperable standards. More distributed type architectures will emerge with inclusive frameworks configurable for human and machine workers to work both in cooperative and competitive modes. These architectures will also support the aggregation of jobs and workers from different repositories using innovative standards and protocols for some sort of crowdsourcing definition language.

6.0 References

Allahbakhsh, M., Benatallah, B., Ignjatovic, A., Motahari-Nezhad, H. R., Bertino, E., & Dustdar, S. (2013). Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing*, *17*(2), 76-81.

Ambati, V., Vogel, S., & Carbonell, J. G. (2010). Active learning and crowd-sourcing for machine translation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)* (pp. 2169-2174).

Behrend, T. S., Sharek, D. J., Meade, A. W., & Wiebe, E. N. (2011). The viability of crowdsourcing for survey research. *Behavior research methods*, *43*(3), 800.

Boer, P. M. D., & Bernstein, A. (2016). Pplib: toward the automated generation of crowd computing programs using process recombination and auto-experimentation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *7*(4), 49.

Candra, M. Z., Truong, H. L., & Dustdar, S. (2013). Modeling elasticity trade-offs in adaptive mixed systems. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2013 IEEE 22nd International Workshop on* (pp. 21-26). IEEE.

Candra, M. Z., Zabolotnyi, R., Truong, H. L., & Dustdar, S. (2014). Virtualizing software and human for elastic hybrid services. In *Advanced Web Services* (pp. 431-453). Springer New York.

Chatzimilioudis, G., Konstantinidis, A., Laoudias, C., & Zeinalipour-Yazti, D. (2012). Crowdsourcing with smartphones. *IEEE Internet Computing*, *16*(5), 36-44.

Desell, T., Goehner, K., Andes, A., Eckroad, R., & Ellis-Felege, S. (2015). On the effectiveness of crowd sourcing avian nesting video analysis at Wildlife@ Home. *Procedia Computer Science*, *51*, 384-393.

Difallah, D. E., Demartini, G., & Cudré-Mauroux, P. (2013). Pick-a-crowd: tell me what you like, and i'll tell you what to do. In *Proceedings of the 22nd international conference on World Wide Web* (pp. 367-374). ACM.

Doan, A., Ramakrishnan, R., & Halevy, A. Y. (2011). Crowdsourcing systems on the world-wide web. *Communications of the ACM*, *54*(4), 86-96.

Dustdar, S., & Bhattacharya, K. (2011). The social compute unit. *IEEE Internet Computing*, *15*(3), 64-69.

Dustdar, S., & Truong, H. L. (2012). Virtualizing software and humans for elastic processes in multiple clouds–a service management perspective. *International Journal of Next - Generation Computing (IJNGC)*, *2012*.

Fowler, M. (2003). Patterns in Enterprise Software. In Web, Retrieved March 4, 2017 from https://martinfowler.com/articles/enterprisePatterns.html

Gao, H., Barbier, G., & Goolsby, R. (2011). Harnessing the crowdsourcing power of social media for disaster relief. *IEEE Intelligent Systems*, *26*(3), 10-14.

Gomaa, H. (2001, July). Designing concurrent, distributed, and real-time applications with UML. In *Proceedings of the 23rd international conference on software engineering* (pp. 737-738). IEEE Computer Society.

Goodchild, M. F., & Glennon, J. A. (2010). Crowdsourcing geographic information for disaster response: a research frontier. *International Journal of Digital Earth*, *3*(3), 231-241.

Heer, J., & Bostock, M. (2010). Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 203-212). ACM.

Horton, J. J., & Chilton, L. B. (2010). The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM conference on Electronic commerce* (pp. 209-218). ACM.

Howe, J. (2006). The rise of crowdsourcing. *Wired magazine*, *14*(6), 1-4.

Huberman, B. A., Romero, D. M., & Wu, F. (2009). Crowdsourcing, attention and productivity. *Journal of Information Science*.

Jarrett, J., Saleh, I., Blake, M. B., Malcolm, R., Thorpe, S., & Grandison, T. (2014). Combining human and machine computing elements for analysis via crowdsourcing. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on* (pp. 312-321). IEEE.

Jarrett, J., & Blake, M. B. (2015). Collaborative Infrastructure for On-Demand Crowdsourced Tasks. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2015 IEEE 24th International Conference on* (pp. 9-14). IEEE.

Jarrett, J., da Silva, L. F., Mello, L., Andere, S., Cruz, G., & Blake, M. B. (2015). Self-Generating a Labor Force for Crowdsourcing: Is Worker Confidence a Predictor of Quality?. In *Hot Topics in Web Systems and Technologies (HotWeb), 2015 Third IEEE Workshop on* (pp. 85-90). IEEE.

Jarrett, J., & Blake, M. B. (2016). Reusable Meta-Models for Crowdsourcing Driven Elastic Systems. In *Information Reuse and Integration (IRI), 2016 IEEE 17th International Conference on* (pp. 50-57). IEEE.

Jarrett, J., Blake, B., & Saleh, I. (2017). Crowdsourcing, Mixed Elastic Systems and Human-Enhanced Computing–A Survey. *IEEE Transactions on Services Computing*.

Jarrett, J., & Blake, M. B. (2017). Interoperability and Scalability for Worker-Job

Matching across Crowdsourcing Platforms. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2017 IEEE 26th International Conference.* IEEE.

Kamar, E., Hacker, S., & Horvitz, E. (2012). Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1* (pp. 467-474). International Foundation for Autonomous Agents and Multiagent Systems.

Keim, T. (2007). Extending the applicability of recommender systems: A multilayer framework for matching human resources. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on* (pp. 169-169). IEEE.

Kittur, A., Chi, E. H., & Suh, B. (2008). Crowdsourcing user studies with Mechanical Turk. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 453-456). ACM.

Kittur, A., Smus, B., Khamkar, S., & Kraut, R. E. (2011). Crowdforge: Crowdsourcing complex work. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (pp. 43-52). ACM.

Kleemann, F., Voß, G. G., & Rieder, K. (2008). Un (der) paid innovators: The commercial utiliza-tion of consumer work through crowdsourcing. *Science, technology & innovation studies*, *4*(1), PP-5.

Kulkarni, A., Gutheim, P., Narula, P., Rolnitzky, D., Parikh, T., & Hartmann, B. (2012). Mobileworks: Designing for quality in a managed crowdsourcing architecture. *IEEE Internet Computing*, *16*(5), 28-35.

Kulkarni, C. E., Socher, R., Bernstein, M. S., & Klemmer, S. R. (2014). Scaling short-answer grading by combining peer assessment with algorithmic scoring. In *Proceedings of the first ACM conference on Learning@ scale conference* (pp. 99-108). ACM.

Lim, S. L., Quercia, D., & Finkelstein, A. (2010). StakeSource: harnessing the power of crowdsourcing and social networks in stakeholder analysis. In *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering-Volume 2* (pp. 239-242). ACM.

Lopez, M., Vukovic, M., & Laredo, J. (2010). Peoplecloud service for enterprise crowdsourcing. In *Services Computing (SCC), 2010 IEEE International Conference on* (pp. 538-545). IEEE.

Malinowski, J., Keim, T., Wendt, O., & Weitzel, T. (2006). Matching people and jobs: A bilateral recommendation approach. In *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on* (Vol. 6, pp. 137c-137c). IEEE.

MSDN. (2017). N-Tier Data Applications Overview. In Web, Microsoft, Retrieved March 17, 2017 from https://msdn.microsoft.com/en-us/library/bb384398.aspx

Ng, A. (n.d.). CS299 Lecture Notes. In Web, Stanford University, Computer Science Department, Retrieved February 2, 2019 from http://cs229.stanford.edu/notes/cs229-notes1.pdf

Quinn, A. J., & Bederson, B. B. (2011). Human computation: a survey and taxonomy of a growing field. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 1403-1412). ACM.

Paparrizos, I., Cambazoglu, B. B., & Gionis, A. (2011). Machine learned job recommendation. In *Proceedings of the fifth ACM Conference on Recommender Systems* (pp. 325-328). ACM.

Poetz, M. K., & Schreier, M. (2012). The value of crowdsourcing: can users really compete with professionals in generating new product ideas?. *Journal of Product Innovation Management*, *29*(2), 245-256.

Riveni, M., Truong, H. L., & Dustdar, S. Towards a Unified Model of Dynamic Properties in Mixed Service Oriented Systems.

Satzger, B., Psaier, H., Schall, D., & Dustdar, S. (2013). Auction-based crowdsourcing supporting skill management. *Information Systems*, *38*(4), 547-560.

Schall, D. (2011). A human-centric runtime framework for mixed service-oriented systems. *Distributed and Parallel Databases*, *29*(5), 333-360.

Schall, D., & Skopik, F. (2010). Mining and composition of emergent collectives in mixed service-oriented systems. In *Commerce and Enterprise Computing (CEC), 2010 IEEE 12th Conference on* (pp. 212-219). IEEE.

Scekic, O., Truong, H. L., & Dustdar, S. (2013). Incentives and rewarding in social computing. *Communications of the ACM*, *56*(6), 72-82.

Schenk, E., & Guittard, C. (2009). Crowdsourcing: What can be Outsourced to the Crowd, and Why. In *Workshop on Open Source Innovation, Strasbourg, France* (Vol. 72).

Schulze, T., Krug, S., & Schader, M. (2012). Workers' task choice in crowdsourcing and human computation markets

Shanks, G., Tansley, E., & Weber, R. (2003). Using ontology to validate conceptual models. *Communications of the ACM*, *46*(10), 85-89.

Sullivan, B., Aycrigg, J., Barry, J., Bonney, R., Bruns, N., Cooper, C., Damoulas, T., Dhondt, A., Dietterich, T., Farnsworth, A. and Fink, D., (2014). The eBird enterprise: an integrated approach to development and application of citizen science. *Biological Conservation*, *169*, pp.31-40.

Tan, W., Blake, M. B., Saleh, I., & Dustdar, S. (2013). Social-network-sourced big data analytics. *IEEE Internet Computing*, *17*(5), 62-69.

Vukovic, M. (2009). Crowdsourcing for enterprises. In *Services-I, 2009 World Conference on* (pp. 686-692). IEEE.

Yan, T., Marzilli, M., Holmes, R., Ganesan, D., & Corner, M. (2009). mCrowd: a platform for mobile crowdsourcing. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems* (pp. 347-348). ACM.

Wong, D. "Data is the Next Frontier, Analytics the New Tool." (2012).

Xu, Z., Liu, Y., Yen, N., Mei, L., Luo, X., Wei, X., & Hu, C. (2016). Crowdsourcing based description of urban emergency events using social media big data. *IEEE Transactions on Cloud Computing*.

Yuen, M. C., King, I., & Leung, K. S. (2011). A survey of crowdsourcing systems. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on* (pp. 766-773). IEEE.

Zaidan, O. F., & Callison-Burch, C. (2011). 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* (pp. 1220-1229). Association for Computational Linguistics.

Zhai, Z., Sempolinski, P., Thain, D., Madey, G., Wei, D., & Kareem, A. (2011, December). Expert-Citizen Engineering:" Crowdsourcing" Skilled Citizens. In *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on* (pp. 879-886). IEEE.