Towards a Distributed Worker-Job Matching Architecture for Crowdsourcing

Julian Jarrett
College of Computing and Informatics
Drexel University
Philadelphia, Pennsylvania, USA
julian.j.jarrett@drexel.edu

M. Brian Blake
College of Computing and Informatics
Drexel University
Philadelphia, Pennsylvania, USA
mbrian.blake@drexel.edu

Abstract— **While the crowdsourcing paradigm facilitates the use of human-enacted resources from large groups of individuals, matching workers with jobs is limited by the need for these potential workers to proactively subscribe to various networks. This subscription phase is part of an "open call model" that reduces the ability for crowdsourcing platforms to scale or retain crowd-oriented workers. Leveraging collaborative filtering techniques, in this paper, we propose an alternative model that seeks to address the issue through a recommendation technique and system that exploits a push-pull model.**

Keywords- crowdsourcing; recommender systems; human computation; labor markets; recruitment; labor force

I. INTRODUCTION

Crowdsourcing [1], through the advent of the Internet and Web 2.0 technologies, has provided a new paradigm for employment, to harness mass human computation and has given new avenues for businesses and researchers to quickly distribute work across a global cross-section of potential workers [1][2]. As defined by Howe [3], the paradigm entails an *open call model* via the Internet to anonymous individuals to solicit services for work, usually at a much cheaper cost than traditional outsourcing [4]. Labor markets [5] such as Amazon Mechanical Turk, Microworkers and UpWork (formerly ODesk) exhaustively use this model. The model however has a significant deficiency [6]; that is the challenge of attracting and maintaining a crowd [7][8]. Via the open call model, tasks requiring human intelligence or HITs are posted for workers to accept relevant task offerings. Those being exposed to the offering are typically members or subscribers of a labor market's community pool or workers [6]. However, there exist massive crowds of potential workers **outside** of the *subscribed* labor market pools and currently the open call model is not capable of leveraging this untapped pool of workers [6][9].

In this paper, augmented by collaborative filtering, we propose a service-oriented architecture based on an open push-pull worker-job matching model capable of harnessing the wisdom and labor potential of diverse communities external to current labor markets. The architecture incorporates transactional web services that implement services pertinent to crowdsourcing including recruitment, job allocation and compensation. We continue by reviewing the open call model including current recruitment strategies and techniques, and worker-job recommender strategies within and external to the paradigm of crowdsourcing. We follow by presenting our proposed collaborative filtering augmented architecture for open push-pull worker-job matching.

II. RELATED LITERATURE

A. Passive Recruitment – The Open Call Model

The task selection process outlined by Schulze, Krug and Schader describes the open call model [6]. It consists of 4 primary steps; platform selection, task details selection, task choice and worker decision. The first of these phases is seen as the kryptonite of the entire paradigm [6]. It requires that users opt to subscribe to a given labor market, access the platform and retrieve a sub list of tasks with high level descriptions. This list is subject to filters enabled on the platform that either includes or excludes tasks by worker characteristics that cannot be changed such as place of birth (hard conditions) and/or by those that can be changed through acquisition such as skillsets (soft conditions). On selecting a task from the list, the worker is then presented with all the low level details of the task. This includes the allotted time, criteria for successful completion with high standards in quality, compensation details amongst others. After perusing such detail, the worker can opt into working on available task instances. From this germinal participation in crowdsourcing, the worker opts to proceed to working on more available instances, search and opt into other available tasks or resign totally from the platform [6].

B. Recruitment Strategies and Techniques

In response to the many challenges faced in an attempt attracting and maintaining a crowd [7][8], diverse stimuli are used to incentivize workers [5]. These come in varying forms including but not limited to entertainment, access to information, volunteerism, altruism, attention from others in the community and the more frequently seen financial compensation [5].

C. Worker-Job Recommender Strategies

Within the realm of crowdsourcing, we find multiple approaches to recommend workers for jobs. Stakesource [10] is a recommender system used in the requirements

978-1-5090-1663-1/16 $31.00
DOI 10.1109/WETICE.2016.12

specification phase of software engineering to identify all stakeholders in a system with the objective to reduce the possibility of neglecting requirements. It works through the aggregation of social network data primarily through the associations of friend of a friend. Stakeholders are used to identify and recommend other stakeholders who are believed to be crucial to the elicitation of requirements and ought to be involved.

Pick-a-Crowd [11] is another job recommender system within the realm of crowdsourcing that seeks to evolve past the first come first serve open call model [3] and propose some form of push method. Workers are assigned tasks based on their interests as it is assumed that workers will perform these tasks given their existing interests. The profiles of the workers are built through garnered information from their social networking profiles with terminologies compared and matched with the Linked Open Data Cloud. Upon matching worker profiles with task descriptions, category, text and graph based approaches are used to assign tasks [11].

The authors in [12] asserted that is difficult to model a workflow with fundamentally different types of computing components. In their work and likewise in ours [13], we focus on the inclusion of human and machine oriented computing units for crowdsourced tasks. For this work, we provide a task that can be inputted to both types of computing units each however through a different mode of input as outlined in our methodology.

Our work evaluates workers' performance and then recommends jobs that are similar in nature to those that have been previously performed by the worker. Similarity characteristics extend to task difficulty, skillsets, other employer-defined metrics and the worker's previous performance. Metric values can be continuous or discrete as their similarities are computed via a distance function. Our recommender works under the assumption that workers will perform jobs well given that they satisfactorily performed jobs similar in nature in their work history. It also assumes that the worker posses the requisite skills for performing the task and others of the like given their previous satisfactory performance. In [11], a push methodology is used similar to our work to assign jobs to workers however differs where jobs are assigned based on worker interests. It also differs where our push mechanism for worker job assignment is preceded by a pull which first collects and analyzes potential workers from diverse communities and jobs from multiple labor markets using collaborative filtering.

III. Proposed Architecture

Core crowd management services include the recruitment of the crowd, compensation models, managing costs and tradeoffs and finally the optimization of skillsets and expertise [9]. We propose a comprehensive platform (Fig. 1) to support the evolution of crowdsourcing from the current open call model to an open push-pull model while maintaining the notion of these core services in our architecture. To support the new open push-pull model, we propose a Service Synchronization and Coordination Middleware (SSCM)[9][14].

A. Diverse Respositories

For our proposed architecture, we propose finding potential workers across several prospective communities. These communities cater to different foci and as such will aid us to diversify our worker pool. We seek workers from professional networks; these are repositories (e.g. LinkedIn, Monster.com, Indeed.com) that maintain worker's CV's, resumes and other professional oriented characteristics. On the grounds that workers may be associated through social circles, we turn to social media networks (e.g. Google+, Facebook). We assume that such workers and their friends will not only share similar interests but also have the high probability to bear similar credentials by association. We also turn to existing labor markets (e.g. Amazon Mechanical Turk) that have already built a community specialized as a labor pool specialized and ready for work [9][14].

B. Open Pull through Crowd and Provider Interfaces

To address the issue of crowd recruitment [7][8], we propose an open pull mechanism. Most e-platforms falling into one of the repository categories mentioned above already provide some type of web service or API that can be used to poll and query users matching a specific criteria. As such we can tailor queries with varying hard and soft conditions to obtain a list of users that can satisfy the requirements for worker candidates for a specific task or category of tasks [9][14].

C. Open Push through the SSCM

Likewise for retention of our crowd, we propose a mechanism as we did for recruitment; an open push. Using recommendation techniques such as machine learning, case-based reasoning and collaborative filtering, we are able to push jobs to our labor force via channels in our SSCM and web API's of the workers' communities. Tasks are recommended to workers based on varying criteria as defined by the owners of the platform; these can include but is not limited to skill qualification requirements for the task, similar task difficulty and similar job description. The SSCM consists of service modules outlined in the introduction of section 3 to support the recruitment of workers to self generate a crowd [14]. This is coordinated through recruitment, job allocation and contract modules, the compensation of workers through the compensation module, the optimization of expertise and skillsets and managing of costs and tradeoffs through the solution resolution module.

IV. Recomennder Engine

Our recommendation engine incorporates a collaborative filtering technique. The principle behind our recommender is grounded in gradient descent using the least squared error algorithm where the path of steepest descent to convergence is taken towards a global maximum based on learned weights. In addition to learning weights, we use the more specialized low ranked matrix approximation algorithm to learn to choose useful features over time for more accurate approximation.

Using the open pull, data is pulled into our system. Typical compatible data sets for our system include several matrices. These include a job matrix with unique IDs, a job feature set matrix with binary or continuous percentage values for each feature, a user matrix with optional skill or expertise level scores based on user profiles, and finally a score matrix with employer feedback for jobs performed by workers. After analysis of our data matrices, our recommender predicts an estimated score for each available job per user. We then take top $N \leq 10$ predictions that are at least an configurable X% similar to jobs in the worker's uptake history recommend to the user. Similarity of jobs is calculated by applying a distance function between the feature set matrices for the recommended job and a job in the worker's uptake history.

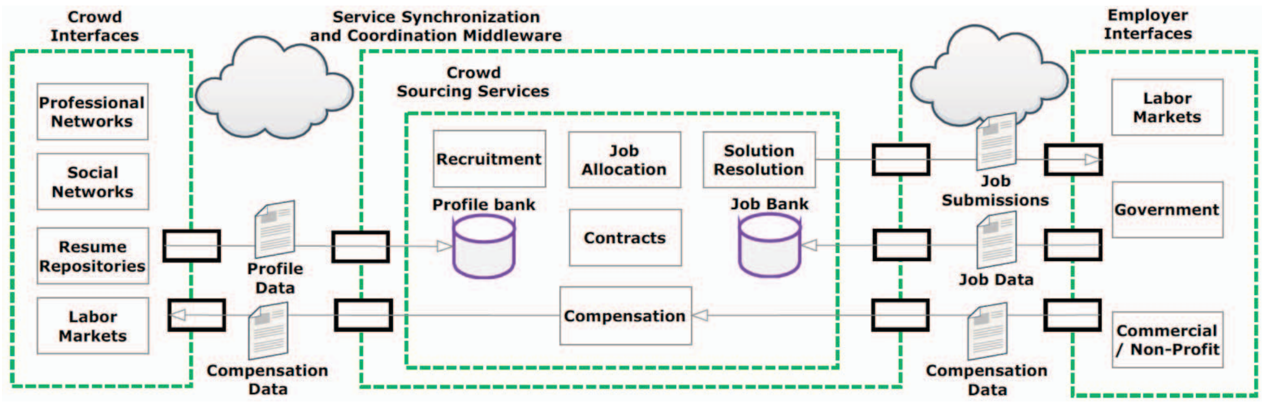

Fig. 1. Open push-pull paradigm support through the service synchronization and coordination middleware.
For completeness we adapted this figure from [15]

V. Conclusion and Outlook

We have proposed a path for crowdsourcing expansion and scaling. From literature, the community acknowledges the limits of the open call model. Through the use of recommender systems, more specifically a collaborative filtering approach, we propose an open push-pull model as an alternative to the open call model. This new model also seeks to mitigate against the issue of attracting, growing and retaining a crowd. We also outline an architecture that can be considered to support the evolution of data including changing values and increasing volumes. We foresee in the future, other approaches that support an open pull /push model emerging from the research community and industry as well as the gradual adaptation of the model as systems migrate from the open call model.

References

[1] M. Yuen, I. King, and K. Leung. 2011. "A survey of crowdsourcing systems." *Privacy, Security, Risk and Trust (PASSAT), 2011.*

[2] A. Kittur, E. H. Chi and B. Suh. 2008. "Crowdsourcing user studies with Mechanical Turk." *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2008.

[3] J. Howe. 2006. "The rise of crowdsourcing." *Wired Magazine,* 14.6 (2006): 1-4.

[4] F. Kleemann, G. Voß, and K. Rieder. 2008. "Un (der) paid innovators: The commercial utiliza-tion of consumer work through crowdsourcing." *Science, Technology & Innovation Studies* 4.1 (2008): PP-5.

[5] J. J. Horton and L. B. Chilton. 2010. "The labor economics of paid crowdsourcing." *Proceedings of the 11th ACM conference on Electronic commerce*. ACM, 2010.

[6] T. Schulze, S. Krug, and M. Schader. 2012. Workers' task choice in crowdsourcing and human computation markets.

[7] A. Doan, R. Ramakrishnan and A. Y. Halevy 2011. "Crowdsourcing systems on the world-wide web." *Comm. of the ACM* 54.4 (2011): 86-96.

[8] A. Quinn and B. Bederson. 2011. "Human computation: a survey and taxonomy of a growing field." 2009. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1403-1412. ACM, 2011.

[9] J. Jarrett, and M. B. Blake. 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.

[10] S. L. Lim, D. Quercia, A. and Finkelstein. "StakeSource: harnessing the power of crowdsourcing and social networks in stakeholder analysis." 2010. *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2*. ACM, 2010.

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

[12] M. Z. Candra, R. Zabolotnyi, H. L. Truong and S. Dustdar. 2014. Virtualizing Software and Human for Elastic Hybrid Services. In *Advanced Web Services*(pp. 431-453). Springer New York.

[13] J. Jarrett, I. Saleh, M. B. Blake, R. Malcolm, S. Thorpe and T. Grandison. 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.

[14] J. Jarrett, M. B. Blake, L. Ferrerira da Silva, L. Mello, S. Andere, G. Oliveira da Cruz. 2015. Self-Generating a Labor Force for Crowdsourcing – "Is Worker Confidence a Predictor of Quality?". *3rd IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb 2015)*.

[15] J. Jarrett, M. B. Blake. "*Using Collaborative Filtering to Automate Worker-Job Recommendations for Crowdsourcing Services*". IEEE 23rd International Conference on Web Services, June 27 - July 2, 2016, San Francisco, USA. - pending