Scalable Human-based Computation

Djellel Eddine Difallah
eXascale Infolab
University of Fribourg
Switzerland

Abstract

Human computation is becoming an essential component of many computer systems architectures, resulting in what can be referred to as *Human-Machine Systems*. In order to foster this adoption, the Human computation component should deliver close to predictable performance, both in terms of speed and quality. My dissertation work aims at addressing some of the current shortcomings of paid micro-task crowdsourcing – a popular form of human computation – with the goal of bridging the gap between the inherent differences of computer systems and human computation, for a seamless integration.

Keywords: Scalability, Crowdsourcing, Human-Machine Systems

Research Motivation

Micro-task crowdsourcing is a form of Human Computation where a requester, be it an human operator or an automated computer system, publishes a set of short tasks to a crowdsourcing platform to be completed by a set of workers in exchange of a micro monetary reward for each task. Popular crowdsourcing platforms include Amazon Mechanical Turk, Crowdflower and Crowdsource.com.

Since the inception of the idea of having an *Human in the loop*, a plethora of systems have been proposed to leverage the unique abilities of the human brain when machines and algorithms fail short at solving a given problem. Nowadays, crowdsourcing is increasingly used in order to obtain large-scale human input for a wide variety of information management tasks, common examples include relevance judgements (Carvalho, Lease, and Yilmaz 2011; Hosseini et al. 2012), image search (Yan, Kumar, and Ganesan 2010) or entity linking (Demartini, Difallah, and Cudré-Mauroux 2012). Moreover, we are witnessing a new breed of "System" softwares, such a data management systems (DBMSs) that are integrating crowdsourcing add-ons to their core architecture and expose Human Computation power to their end-users, either through an explicit interface such as *CrowdDB* (Franklin et al. 2011) or implicit such as *Arnold* (Jeffery et al. 2013).

This success is somewhat tempered by i) the actual state of affairs of crowdsourcing platforms that offer no Service Level Agreements and little guarantees to their users, and ii) the inherent properties of the crowd that differ significantly from the high and scalable performance of a computer system.

My dissertation work aims at addressing the scalability issues in micro task crowdsourcing that hinder the developments of further large-scale data management systems, especially in the context of ever pressing big-data challenges. The key research directions I am tackling are:

- *Speed & Latency*: In the absence of SLAs on current crowdsourcing platforms, requesters are often limited to increase the price of their tasks for a better execution time. How does worker retention and task load balancing improve the execution time on a crowdsourcing platform?
- *Quality vs Cost*: Paid micro-task crowdsourcing comes with a monetary cost, and the base budget is often not enough since ensuring quality would require duplicate executions and even verification steps. How does worker profiling help reducing the monetary cost of a crowdsourcing campaign?
- *Human-Machine Integration*: How to integrate and use parsimoniously, a crowdsourcing component in a computer system?

Related Work

The initial uses of crowdsourcing techniques included the creation of test collections for repeatable relevance assessment, machine learning training sets or active learning (Alonso and Baeza-Yates 2011; Kazai 2011; Kazai et al. 2011). Nowadays, hybrid human-machine systems employ crowdsourcing in order to provide better solutions as compared to purely machine-based systems. For example, CrowdDB (Franklin et al. 2011) is a crowd-powered database system that is able to answer SQL queries including special operators that are crowdsourced. Such a system is for instance able to find pictures to be used in motivational slides by asking the crowd to rate pictures that are stored in the database. Another example is the use of crowdsourcing to answer tail queries in web search engines (Bernstein et al. 2012b). Here the goal is to ask the crowd to find the answer to an unpopular search engine query within a set of machine-selected candidate Web pages. Similarly, (Demartini et al. 2013) propose to leverage the crowd ability

to understand a query expressed in natural language and map it into relational operators, CrowdQ mixes natural language processing and human intelligence to create templates that can be applied for future queries. (Marcus et al. 2012; 2011) optimize the count, sort and join operators of a DBMS to the case of crowdsourcing.

More examples of hybrid human-machine systems include: entity resolution (Wang et al. 2012; Whang, Lofgren, and Garcia-Molina 2013), entity linking (Demartini, Difallah, and Cudré-Mauroux 2012), schema matching (Zhang et al. 2013), association rule mining (Amsterdamer et al. 2013), word sense disambiguation (Seemakurty et al. 2010), and query answering (Park et al. 2013; Selke, Lofi, and Balke 2012).

In nearly all previous scenarios, achieving low latency is key. In a multi-tenant setting, either concurrent requesters on a crowdsourcing platform or users of a Crowd-powered DBMS, it is important that users who post critical queries do not have to wait long before getting back an answer from the system. Recent work started to tackle such issues where near real time is a necessity, e.g.: speech captioning or live help of visually impaired persons(Bigham et al. 2010; Bernstein et al. 2012a), such approaches try to have and maintain a pool of workers of a minimum size always available to answer a given request.

Another more generally accepted approach to improve latency is to increase the price. To better understand this phenomena, a number of recent contributions studied the effect of monetary incentives on crowdsourcing platforms. In (Mao et al. 2013), compared crowdsourcing results obtained using both volunteers and paid workers. Their findings show that the quality of the work performed by both populations is comparable, while the results are obtained faster when the crowd is financially rewarded. Wang *et al.* (Wang, Ipeirotis, and Provost 2013) looked at pricing schemes for crowdsourcing platforms focusing on the quality dimension: The authors proposed methods to estimate the quality of the workers and introduce new pricing schemes based on the expected contribution of the workers.. Chandler and Horton (Chandler and Horton 2011) analyzed (among others) the effect of financial bonuses for crowdsourcing tasks that would be ignored otherwise. Their results show that monetary incentives worked better than non-monetary ones given that they are directly noticeable by the workers. Recently also, Singer *et al.* (Singer and Mittal 2013) studied the problem of pricing micro-tasks in a crowdsourcing marketplace under budget and deadline constraints. Faradani *et al.* (Faradani, Hartmann, and Ipeirotis 2011) studied the problem of predicting the completion of a batch of HITs and at its pricing given the current marketplace situation. They proposed a new model for predicting batch completion times showing that longer batches attract more workers.

Increasing the reward of a micro-task very often comes with the price of lower quality, indeed crowd spammers are motivated by quick gain knowing that the requesters will unlikely verify the submitted answers. The most used mechanisms to control quality is by inserting test questions and or creating task repetitions (multiple workers for the same task). Another mechanism to control quality and execution time of on crowdsourcing platforms is by means of active rules to plan the assignment of tasks based on the system performance (Bozzon et al. 2013a). Such an approach is also related to scheduling approaches as it aims at improving crowdsourcing efficiency and effectiveness. Also related to crowdsourcing effectiveness is (Bozzon et al. 2013b) where the goal is to find experts in online social networks based on their activities as candidates for crowdsourced tasks.

Research Questions and Methodology

Workforce Scalability

The timely completion of a crowdsourcing campaign is, as of today, hardly guaranteed and many factors influence its progression pace, including: the crowd availability and demographics, time-of-day, the amount of the micropayments, the number of remaining tasks in a given batch, concurrent campaigns, or the reputation of the publisher and concurrent publishers etc. I propose to research the application of computer system scalability techniques to paid micro-task crowdsourcing as follows:

Scaling-out the crowd: In this model, a large number of workers complete the tasks in parallel and compete for the next tasks. Under the assumption that a large crowd of workers is always available to handle the different tasks at hand, this model can minimize the batch execution time by increasing the competition among workers.
Q1: What are the task scheduling schemes that can be applied if we were to serve tasks in a push fashion?

Scaling-up the crowd: A different way of scaling a crowdsourcing campaign is to focus on attaining higher worker retention rates such that they keep working longer on a given batch. This model potentially presents two advantages: It minimizes the down times incurred when waiting for new workers, and yields potentially better workers having more experience handling a given task.
Q2: Under which circumstances does worker retention enhances the response time in crowdsourcing?

Worker Profiling for Cost Minimization

The crowd is a large mass of anonymous participants with varying skills, objectives and intentions. Low quality submissions can be rooted to the presence of malicious, or unqualified workers. To remediate to this situation i) qualification tests are sometimes required from the workers, ii) tasks are run with multiple repetitions (e.g.: the same task is done by 3 different workers), or iii) further verification steps are required. These methods increase the cost of crowdsourcing by many folds and thus lower the scalability of a solution.

Crowd Expert Finding: If we consider that some tasks can leverage the knowledge of a given set of workers, then we can map the setting to an expert finding – among the crowd – problem; one way this could be achieved is by leveraging the social profile of the participants.

Q3: How to identify expertise from social profiles and associate it with a given crowdsourcing batch?

Reputation Score: Tracking the progress of a given worker and assigning a reputation score to him is another approach that can be used on anonymous workers.
Q4: How to limit the number of task repetitions using probabilistic reasoning to identify good and bad workers?

Research Methodology

The research approach of my thesis is empirical and the applied methodologies are qualitative, quantitative and experimental where i) proceed to literature review of a specific problem and identify potential new approaches ii) If necessary, run preliminary surveys on the crowd participants to refine the hypothesis iii) build a prototype and run real world experiments to test the effectiveness of a given solution iv) gather participants feedback and refine the prototype v) create a set of tools to address the research questions.

Completed Work

Pick-A-Crowd – Personalized Task Assignment We proposed Pick-A-Crowd(Difallah, Demartini, and Cudré-Mauroux 2013), a system exploiting a novel crowdsourcing scheme focusing on pushing tasks to the right worker rather than letting the workers pull the tasks they wished to work on. We proposed a novel crowdsourcing architecture that builds worker profiles based on their online social network activities and tries to understand the skills and interests of each worker. Thanks to such profiles, Pick-A-Crowd is able to assign each task to the right worker dynamically. Experimental results over the system user-base show that all of the proposed models outperform the classic first-come-first-served approach used by standard crowdsourcing platforms such as Amazon Mechanical Turk. Our best approach provides on average 29% better results than the Amazon MTurk model.

ZenCrowd – Data Integration with Probabilistic Reasoning ZenCrowd (Demartini, Difallah, and Cudré-Mauroux 2012) is a system for automatic entity extraction and linking. It is based on a probabilistic framework leveraging both automatic techniques and punctual human intelligence feedback. ZenCrowd can be used in combination with automatic entity extraction, ranking, and matching techniques to improve the overall linking accuracy. The novelty in the crowdsourcing component is the probabilistic framework that helps identifying reliable workers. More than a simple majority vote, this technique assigns a weight to workers providing correct answers to hidden test questions, their subsequent votes will be more valued while reinforcing the system knowledge about new workers not yet assessed.

Scaling-up the Crowd – Latency Improvement through Worker retention

In this work(Difallah et al. 2014) we proposed series of bonus schemes that helps retaining workers on a given batch. One of the best performing schemes that we studied is based on milestones bonuses, where a worker will receive a extra monetary reward after each N tasks he submits. Among the main findings of this work is that by using a milestone bonus we could finish a crowdsourcing batch with less workforce and comparable speed as opposed to a similar batch proposing a higher base reward.

Ongoing and Future Work

Scheduling HIT – Task Load Balancing in a Multi-tenant Setting Scheduling is the traditional way of tackling latency problems in computer science by prioritizing access to shared resources to achieve some quality of service. In that sense, we can consider a crowdsourcing platform as a pool of running tasks which have access to the same resources i.e., the online crowd. I built a prototype and did experimental comparisons of several scheduling techniques to optimize job completion time in a crowd-powered DBMS setting which publishes task to Mturk on behalf of multiple users. The aim of the study is at i) assessing the efficiency of the crowd in settings where multiple types of requests are run concurrently, and b) understanding the tradeoffs of tasks scheduling over the crowd. Ultimately, come up with a set of scheduling algorithms that can be applied in crowdsourcing platforms.

Analysis of Batch lifetime in the realm of a crowdsourcing Platform Batches of tasks running in a crowdsourcing platform are subject to many factors that affect their performance, both in terms of speed and quality. For example, a common result is that increasing the price of a HIT would lead to faster results, however, we still do not know the exact impact of changing the price in the presence of many reputable requesters on the platform. Or, what is the effect of posting a link to a batch on a popular crowdsourcing forum etc. The goal of this work is to study the different variables, and their interactions, that contribute to the progression of a given batch. With that regard, I am planning on using a data driven approach where I integrate logs from different sources (e.g.: Turkopticon, specialised forums, mturk-tracker etc). The findings of such work will contribute to a better understanding of the dynamics of a crowdsourcing platform – here we take Mturk as the reference use case.

Discussion

Graceful *Scalability* is an essential property of IT systems aiming at minimizing their latency while answering increasing numbers of requests concurrently. Integrating the crowd in such systems presents unique challenges and opportunities, but it only makes sense if these systems maintain their scalable property. With that regard, my early results show that retention can help lowering execution time when there are limited number of workers. While my current investigation early results suggest that: given a continuous influx of workers, tasks can be scheduled to achieve prioritization and meet deadlines without bothering the crowd.

References

Alonso, O., and Baeza-Yates, R. A. 2011. Design and Implementation of Relevance Assessments Using Crowdsourcing. In *ECIR*, 153–164.

Amsterdamer, Y.; Grossman, Y.; Milo, T.; and Senellart, P. 2013. Crowdminer: Mining association rules from the crowd. *Proc. VLDB Endow.* 6(12):1250–1253.

Bernstein, M. S.; Karger, D. R.; Miller, R. C.; and Brandt, J. 2012a. Analytic methods for optimizing realtime crowdsourcing. *arXiv preprint arXiv:1204.2995*.

Bernstein, M. S.; Teevan, J.; Dumais, S.; Liebling, D.; and Horvitz, E. 2012b. Direct answers for search queries in the long tail. In *CHI '12*, 237–246. ACM.

Bigham, J. P.; Jayant, C.; Ji, H.; Little, G.; Miller, A.; Miller, R. C.; Miller, R.; Tatarowicz, A.; White, B.; White, S.; et al. 2010. Vizwiz: nearly real-time answers to visual questions. In *UIST*, 333–342. ACM.

Bozzon, A.; Brambilla, M.; Ceri, S.; and Mauri, A. 2013a. Reactive crowdsourcing. In *Proceedings of the 22nd international conference on World Wide Web*, WWW '13, 153–164. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee.

Bozzon, A.; Brambilla, M.; Ceri, S.; Silvestri, M.; and Vesci, G. 2013b. Choosing the right crowd: expert finding in social networks. In *EDBT '13*, 637–648. ACM.

Carvalho, V. R.; Lease, M.; and Yilmaz, E. 2011. Crowdsourcing for search evaluation. In *ACM Sigir forum*, volume 44, 17–22. ACM.

Chandler, D., and Horton, J. J. 2011. Labor Allocation in Paid Crowdsourcing: Experimental Evidence on Positioning, Nudges and Prices. In *Human Computation*.

Demartini, G.; Trushkowsky, B.; Kraska, T.; Franklin, M. J.; and Berkeley, U. 2013. Crowdq: Crowdsourced query understanding. In *CIDR*.

Demartini, G.; Difallah, D. E.; and Cudré-Mauroux, P. 2012. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, 469–478. ACM.

Difallah, D. E.; Catasta, M.; Demartini, G.; and Cudré-Mauroux, P. 2014. Scaling-up the crowd: Micro-task pricing schemes forworker retention and latency improvement. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

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

Faradani, S.; Hartmann, B.; and Ipeirotis, P. G. 2011. What's the Right Price? Pricing Tasks for Finishing on Time. In *Human Computation*.

Franklin, M. J.; Kossmann, D.; Kraska, T.; Ramesh, S.; and Xin, R. 2011. CrowdDB: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, SIGMOD '11, 61–72. New York, NY, USA: ACM.

Hosseini, M.; Cox, I. J.; Milić-Frayling, N.; Kazai, G.; and Vinay, V. 2012. On aggregating labels from multiple crowd workers to infer relevance of documents. In *Advances in information retrieval*. Springer. 182–194.

Jeffery, S. R.; Sun, L.; DeLand, M.; Pendar, N.; Barber, R.; and Galdi, A. 2013. Arnold: Declarative crowd-machine data integration. In *CIDR*.

Kazai, G.; Kamps, J.; Koolen, M.; and Milic-Frayling, N. 2011. Crowdsourcing for book search evaluation: impact of hit design on comparative system ranking. In *SIGIR*, 205–214.

Kazai, G. 2011. In Search of Quality in Crowdsourcing for Search Engine Evaluation. In *ECIR*, 165–176.

Lasecki, W. S.; Marcus, A.; Tzeszotarski, J. M.; and Bigham, J. P. 2014. Using Microtask Continuity to Improve Crowdsourcing. In *Carnegie Mellon University Human-Computer Interaction Institute - Technical Reports - CMU-HCII-14-100*.

Mao, A.; Kamar, E.; Chen, Y.; Horvitz, E.; Schwamb, M. E.; Lintott, C. J.; and Smith, A. M. 2013. Volunteering Versus Work for Pay: Incentives and Tradeoffs in Crowdsourcing. In *HCOMP*.

Mao, A.; Kamar, E.; and Horvitz, E. 2013. Why Stop Now? Predicting Worker Engagement in Online Crowdsourcing. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Marcus, A.; Wu, E.; Karger, D.; Madden, S.; and Miller, R. 2011. Human-powered sorts and joins. *Proceedings of the VLDB Endowment* 5(1):13–24.

Marcus, A.; Karger, D.; Madden, S.; Miller, R.; and Oh, S. 2012. Counting with the crowd. *Proceedings of the VLDB Endowment* 6(2):109–120.

Park, H.; Pang, R.; Parameswaran, A.; Garcia-Molina, H.; Polyzotis, N.; and Widom, J. 2013. An overview of the deco system: Data model and query language; query processing and optimization. *SIGMOD Rec.* 41(4):22–27.

Seemakurty, N.; Chu, J.; von Ahn, L.; and Tomasic, A. 2010. Word sense disambiguation via human computation. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, 60–63. ACM.

Selke, J.; Lofi, C.; and Balke, W.-T. 2012. Pushing the boundaries of crowd-enabled databases with query-driven schema expansion. *Proc. VLDB Endow.* 5(6):538–549.

Singer, Y., and Mittal, M. 2013. Pricing Mechanisms for Crowdsourcing Markets. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, 1157–1166. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee.

Wang, J.; Kraska, T.; Franklin, M. J.; and Feng, J. 2012. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment* 5(11):1483–1494.

Wang, J.; Ipeirotis, P. G.; and Provost, F. 2013. Quality-Based Pricing for Crowdsourced Workers. In *NYU Stern Research Working Paper - CBA-13-06*.

Whang, S. E.; Lofgren, P.; and Garcia-Molina, H. 2013. Question Selection for Crowd Entity Resolution. *Proc. VLDB Endow.* 6(6):349–360.

Yan, T.; Kumar, V.; and Ganesan, D. 2010. Crowdsearch: Exploiting crowds for accurate real-time image search on mobile phones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, 77–90. New York, NY, USA: ACM.

Zhang, C. J.; Chen, L.; Jagadish, H. V.; and Cao, C. C. 2013. Reducing uncertainty of schema matching via crowdsourcing. *Proc. VLDB Endow.* 6(9):757–768.