Quality control in crowd work

Crowd work — that is, piecework [2] on platforms such as Amazon Mechanical Turk — has become both a crucial source of income for many workers, and a major source of annotations for modern machine learning systems. Tasks on crowd work platforms range from the prosaic (e.g., transcribing audio) to the novel (literally, in some cases, writing novels [11]). Nearly every approach to crowd work requires a strategy for quality control, deciding which workers are allowed to continue working on the task, or at worst, which ones are denied payment [4, 15, 18, 19]. These algorithmic approaches variously seek to test workers' abilities, measure cross-worker agreement, and motivate high–effort responses. Workers' reputations — and by extension, their prospective abilities to find work — are determined by whether these algorithms decide that workers' submissions are high quality.

Unfortunately, these quality control algorithms have — perhaps unintentionally — resulted in wage theft for many workers, because the algorithms deny payment for good-faith work. There are often many correct ways to interpret a task, and often these algorithms only recognize and pay for the most common one [10]. Requesters' right to reject work without compensation is the second–most common discussion topic about the Mechanical Turk participation agreement [14], and mass rejection is a persistent, widespread fear [13]. While only a small minority of work is rejected [7], the fear of getting work algorithmically rejected looms large [14]. To manage the concern, workers use back–channels to share information about which requesters are reliable and don't reject unnecessarily [6, 8].

We can think of the relationship Turkers have with the systems that manage them as analogous in some sense to the foremen who used to manage factory workers. Pieceworkers in railroads, car assembly, and many other industries historically interacted with foremen as their interface with the company: the foreman assigned workers to suitable work given their qualifications, responded to workers when they needed assistance, and provided feedback on the output of the work. This relationship was a crucial boundary between managers and workers that foremen had to navigate carefully — neither management nor worker, but decidedly and intentionally in the middle [20].

The foreman's job was important because even the most standardized work sometimes surfaces unique circumstances. As much as managers attempt to routinize work, scholarship on the routinization of work tells us that improvisation remains a necessary aspect of even the most narrowly prescribed work [1, 9].

> *When performance is difficult to evaluate, imperfect input measures and a manager's subjective judgment are preferable to defective (simple, observable) output measures.* –[3], quoted in [2]

The challenge is that algorithmic review mechanisms are not well–equipped to understand unusual cases. A crowd worker's output is almost never evaluated by humans directly, but algorithmically scored either in comparison to the work of other workers or a known "gold standard" correct response [12]. However, often the most popular answer isn't actually the correct one [16], and a gold standard answer may not be the only correct answer [10]. If the task becomes more complex, for example writing, algorithmic systems fall back to evaluating for syntactic features that, paradoxically, both make it easy to game and frustratingly difficult to succeed [21]. This general characterization of an algorithmic agent — an algorithmic system that essentially seeks agreement in some form — is not designed to evaluate entirely novel work. With all of the mistakes these systems make, and with the additional work that crowd workers have to do to make these systems work [5], it should come as little surprise that crowd workers are hesitant to attempt work from unknown and unreliable requesters [14].

The problem is that these algorithmic foremen can't distinguish novel answers from wrong answers. Rare responses do not necessarily mean that the worker was not paying attention — in fact, we prize experts for unique insights and answers nobody else has produced. However, where the foreman would evaluate unusual work relative to its particular constraints, the algorithm at best can only ask if this work resembles other work. Crowd workers might then receive a mass rejection as a result, harming their reputation on the platform. Again as in other cases we've discussed, gathering more training data is not a feasible path to fix the problem: crowd work is often carried out exactly in situations where such data does not yet exist. Machine learning algorithms that evaluate worker effort also require substantial data for each new task [17]. The street–level algorithm is stuck with a cold–start problem where it does not have enough data to evaluate work accurately.

REFERENCES

[1] Paul S. Adler, Barbara Goldoftas, and David I. Levine. 1999. Flexibility Versus Efficiency? A Case Study of Model Changeovers in the Toyota Production System. *Organization Science* 10, 1 (1999), 43–68. https://doi.org/10.1287/orsc.10.1.43

[2] Ali Alkhatib, Michael S. Bernstein, and Margaret Levi. 2017. Examining Crowd Work and Gig Work Through The Historical Lens of Piecework. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4599–4616. https://doi.org/10.1145/3025453.3025974

[3] Erin Anderson and David C. Schmittlein. 1984. Integration of the Sales Force: An Empirical Examination. *The RAND Journal of Economics* 15, 3 (1984), 385–395. http://www.jstor.org/stable/2555446

[4] Jeffrey P. Bigham, Michael S. Bernstein, and Eytan Adar. 2015. Human-Computer Interaction and Collective Intelligence. In *Handbook of Collective Intelligence*. MIT Press, 57–84. http://repository.cmu.edu/cgi/viewcontent.cgi?article=1264{&}context=hcii

[5] Lydia B. Chilton, John J. Horton, Robert C. Miller, and Shiri Azenkot. 2010. Task Search in a Human Computation Market. In *Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP '10)*. ACM, 1–9. https://doi.org/10.1145/1837885.1837889

[6] Mary L. Gray, Siddharth Suri, Syed Shoaib Ali, and Deepti Kulkarni. 2016. The Crowd is a Collaborative Network. In *Proceedings of the 19th ACM Conference on Computer–Supported Cooperative Work & Social Computing (CSCW '16)*. ACM, 134–147. https://doi.org/10.1145/2818048.2819942

[7] Kotaro Hara, Abigail Adams, Kristy Milland, Saiph Savage, Chris Callison-Burch, and Jeffrey P Bigham. 2018. A Data-Driven Analysis of Workers' Earnings on Amazon Mechanical Turk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 449.

[8] Lilly C. Irani and M. Six Silberman. 2013. Turkopticon: Interrupting Worker Invisibility in Amazon Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, 611–620. https://doi.org/10.1145/2470654.2470742

[9] B.M. Jewell. 1921. *The problem of piece work*. Number nos. 1-16 in The Problem of Piece Work. Bronson Canode Print. Co. https://books.google.com/books?id=NN5NAQAAIAAJ

[10] Sanjay Kairam and Jeffrey Heer. 2016. Parting Crowds: Characterizing Divergent Interpretations in Crowdsourced Annotation Tasks. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '16)*. ACM, 1637–1648. https://doi.org/10.1145/2818048.2820016

[11] Joy Kim, Sarah Sterman, Allegra Argent Beal Cohen, and Michael S Bernstein. 2017. Mechanical Novel: Crowdsourcing Complex Work through Revision. In *Proceedings of the 20th ACM Conference on Computer Supported Cooperative Work \& Social Computing*.

[12] John Le, Andy Edmonds, Vaughn Hester, and Lukas Biewald. 2010. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *SIGIR 2010 workshop on crowdsourcing for search evaluation*. 21–26.

[13] David Martin, Benjamin V. Hanrahan, Jacki O'Neill, and Neha Gupta. 2014. Being a Turker. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*. ACM, 224–235. https://doi.org/10.1145/2531602.2531663

[14] Brian McInnis, Dan Cosley, Chaebong Nam, and Gilly Leshed. 2016. Taking a HIT: Designing Around Rejection, Mistrust, Risk, and Workers' Experiences in Amazon Mechanical Turk. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, 2271–2282. https://doi.org/10.1145/2858036.2858539

[15] Tanushree Mitra, C.J. Hutto, and Eric Gilbert. 2015. Comparing Person- and Process–centric Strategies for Obtaining Quality Data on Amazon Mechanical Turk. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, 1345–1354. https://doi.org/10.1145/2702123.2702553

[16] Dražen Prelec. 2004. A Bayesian truth serum for subjective data. *science* 306, 5695 (2004), 462–466.

[17] Jeffrey M. Rzeszotarski and Aniket Kittur. 2011. Instrumenting the Crowd: Using Implicit Behavioral Measures to Predict Task Performance. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, 13–22. https://doi.org/10.1145/2047196.2047199

[18] Aaron D. Shaw, John J. Horton, and Daniel L. Chen. 2011. Designing Incentives for Inexpert Human Raters. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work (CSCW '11)*. ACM, 275–284. https://doi.org/10.1145/1958824.1958865

[19] Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. 2008. Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*. ACM, 614–622. https://doi.org/10.1145/1401890.1401965

[20] William F. Whyte and Burleigh B. Gardner. 1945. The Position and Problems of the Foreman. *Applied Anthropology* 4, 2 (1945), 17–25. http://www.jstor.org/stable/44135127

[21] Michael Winerip. 2012. Facing a robo-grader? Just keep obfuscating mellifluously. *New York Times* 22 (2012).