On Quality Control and Machine Learning in Crowdsourcing

Matthew Lease
School of Information
University of Texas at Austin
ml@ischool.utexas.edu

Abstract

The advent of crowdsourcing has created a variety of new opportunities for improving upon traditional methods of data collection and annotation. This in turn has created intriguing new opportunities for data-driven machine learning (ML). Convenient access to crowd workers for simple data collection has further generalized to leveraging more arbitrary crowd-based human computation (von Ahn 2005) to supplement automated ML. While new potential applications of crowdsourcing continue to emerge, a variety of practical and sometimes unexpected obstacles have already limited the degree to which its promised potential can be actually realized in practice. This paper considers two particular aspects of crowdsourcing and their interplay, data quality control (QC) and ML, reflecting on where we have been, where we are, and where we might go from here.

Introduction

Crowdsourcing has attracted much attention in the research community by enabling data, particularly labeled data, to be obtained more quickly, cheaply, and easily (Snow et al. 2008). Such new abundance of labeled data has been a boon to data-driven machine learning (ML) and led to a surge in use of crowdsourcing in ML studies. However, crowdsourcing has typically been found to yield noisier data than traditional practices of in-house annotation, which has generated significant interest in developing effective quality control (QC) mechanisms in order to improve data quality. This, in turn, has led to interest in QC in its own right as a challenging problem for ML.

While many QC challenges being encountered are not new, some people may be encountering them for the first time due to the various ways in which crowdsourcing has reduced traditional barriers to data collection which formerly encouraged many researchers to re-use existing data rather than collect and annotate their own. Annotation (a.k.a. "labeling", or in social sciences, "coding") is well-studied with a wealth of prior methodology and experience (e.g. the Linguistic Data Consortium) related to design of annotation guidelines, organizing and managing the annotation workflow, measures of inner-annotator agreement, etc., that can usefully inform crowdsourcing-based annotation activities.

Crowdsourcing is also quickly changing the landscape for the quantity, quality, and type of labeled data available for training data-driven ML systems. With regard to cost, we might assume the cost differential between crowd-based and in-house annotation is around an order of magnitude, meaning we might expect a ten-fold increase in the amount of labeled data we can now afford. Effort traditionally required to collect and annotate data has also significantly limited quantity of labeled data. Crowdsourcing has now greatly reduced this as well. Finally, turn-around time from idea to training data has now dropped from weeks or days to hours or minutes, depending on complexity of the learning task. Assuming current trends continue and crowd-specific QC is increasingly relegated to lower-level automated system infrastructure, cost and effort will further decrease while quality and speed increase. Overall, such trends have potential for such explosive growth in total availability and creation time of labeled data for training ML systems.

Quality Control (QC)

Effective QC plays an important role in determining the success of any data collection venture, be it via crowdsourcing or otherwise. Early crowdsourcing studies have seen QC receive somewhat inconsistent treatment, at times being not appreciated until too late, and at times seeming to have been overlooked entirely (if a human supplied the data, it must be high quality, right?). Certainly crowdsourcing has made it easier than ever before to collect voluminous amounts of lousy data on an unprecedented scale. If we care about data quality, however, we probably want to think about QC.

QC is certainly not a new problem, as with many other issues being addressed in crowdsourcing research today (Adar 2011), though it may appear in altered form or be more prevalant than in traditional settings. We may not have the same workers annotating all examples, affecting how we measure inner-annotator agreement. Managing a workflow of distributed workers on micro-tasks differs from managing a team of in-house, hourly workers. While irresponsible workers (or test subjects) are not new, they may be more prevalent or have greater detrimental impact. While telecommuting is a well-established labor practice, crowd work takes it to a new extreme, often with lower quality communication channels and fewer opportunities to build rapport with remote workers, especially if they are anonymous.

Human Factors. Since it is people who form a crowd, crowdsourcing is inherently a human-centric enterprise. As such, human factors merit particular consideration for effective design and use of crowdsourcing, and one would expect that established principles and methodology from the field of human-computer interaction (HCI) could be particularly informative. The sad irony is that while crowdsourcing has attracted some of the greatest interest in computer science (CS) areas like ML, computer vision, and natural language processing (NLP), HCI has typically received relatively little attention in traditional CS curricula and research. As a consequence, those who might now benefit from HCI methodology may not have sufficient access or opportunity.

For example, we in the ML community often optimize metrics and evaluate on synthetic data as proxies for dealing with human subjects, with the assumption that improvements in these artificial settings will translate to tangible improved use by people in practice. Published ML research typically does not include user studies or limits such studies to a relatively small group of CS graduate students or researchers as the test subjects. While we recognize the importance of human factors and human-centered evaluation, we have been largely happy to leave this to others.

While such specialization and division of labor has been an invaluable organizing structure for facilitating scientific progress, it can be problematic when a disruptive shift like crowdsourcing crosses the traditional artificial boundaries we have constructed between knowledge areas. The myriad of new opportunities crowdsourcing is enabling in areas like ML and NLP has led to many of us suddenly taking up collecting our own data for the first time, and unsurprisingly, making a few mistakes along the way. We who thought we did not have to worry about designing user interfaces or interaction mechanisms, or worry about human relations (HR) issues, suddenly find ourselves doing all of these activities in crowdsourcing. When we ask users to perform a task that is simple and obvious to us, yet they screw it up, we may infer perhaps that the workers are lazy or deceitful, when in fact it may our our own poor design that is truly to blame.

As an analogy, consider the popular waterfall model for software engineering. This model assumes that users know in advance what features they want, allowing an initial requirements analysis to hand off a specification that is then built and delivered. The problem is that when the above assumption does not actually hold in practice, users are dissatisfied with the delivered product, at which point the naive software engineer laments his fickle users rather than appreciate his own design error. Similarly, the real culprit limiting the quality of collected data in crowdsourcing may often be our own inattention to human factors. An omnipresent challenge for all researchers is to become more aware of the disciplinary lens through which they observe the world and how that shapes their interpretations of observed phenomena. An ML researcher observing low quality data may naturally see the problem with an eye toward ML issues, affording relatively less attention to considering non-ML ways such poor quality may have arisen or strategies for addressing it.

Automation. A large amount of work to date on crowdsourcing QC has investigated ML strategies for detecting "spammers" (or assessing worker quality more generally) and aggregating labels from multiple workers (i.e. *consensus* methods) to cancel-out mistakes made by individual workers (Whitehill et al. 2009). Such QC may be performed as a distinct post-hoc cleanup stage after data collection, or it may take the form of real-time monitoring of annotation to determine which examples to annotate next and at what degree of pluralism (redundancy) labeling should be performed, etc. While there is clearly an important need for such work, it is also important to recognize the limitations of this approach to QC and any assumptions of worker behavior that underly it. For example, do we adopt a prior model of our workers as by-and-large responsible or irresponsible? While many such irresponsible behaviors have been observed, counter examples have been reported as well, with hypothesized correlation between certain worker behaviors and the underlying labor model (Kochhar, Mazzocchi, and Paritosh 2010). It is also certainly the case that it is easy to make catastrophic mistakes early in data collection from which no amount of ML post-processing will ever be able to cleanup. ML-based QC may really be best at just filtering out spammers and computing majority vote (Ipeirotis 2011).

Today we are seeing aspects of QC being increasingly automated and pushed down into a lower, system-level layer of data collection engines (Little et al. 2009). Such a separation of concerns recognizes that at least some QC issues can be generalized across the different types of data being collected, and thus handled in one place for the benefit of many. It thereby creates a useful level of abstraction for practitioners, who can focus on articulating their particular annotation guidelines and let the system worry about low-level QC. Such infrastructure seems critical to enabling wider adoption of crowdsourcing practices by those who are less familiar with these issues and do not have the time, interest, or risk tolerance to address low-level QC issues themselves. Vendors like CrowdFlower[1] are seeking to fill this need.

Annotation. We should also remember that QC is not merely a question of workers' intelligence, effort, and personal biases. QC is also a question of how well annotation guidelines are communicated, how well the guidelines cover the infinite variety of data encountered in practice, and internal-consistency of the guidelines themselves. The term "guidelines" itself conveys their lack of completeness and definitive rules for procedural execution. The value of human annotation is not merely sensory but also analytical and interpretive, especially (but not only) when we are interested in annotating more complex phenomena. Annotation guidelines represent a living document, especially at early stages of the annotation process. Traditional practice iteratively revises them as annotators become more familiar with the data and encounter examples for which existing guidelines are ambiguous, unclear, or not covered. While annotators could individually make arbitrary decisions for such difficult examples and blithely march on, such a process would be a recipe for inconsistency. Instead, annotators typically dis-

[1] `http://crowdflower.com`

cuss such questionable cases to arrive at consensus on how such examples should be annotated (consistently across annotators), and codify their decisions by revising the guidelines (Kochhar, Mazzocchi, and Paritosh 2010).

Worker and Task Organization. With crowdsourcing, we must still contend with such challenges, but we must learn how to appropriately address them in the new environment. One labor model close to the traditional one would assign largely traditional roles to distributed workers. They would engaged at a high-level, assigned trust and responsibility, and would essentially function as tele-commuters, just as many of us already do in our own professional roles. Such a model has been successfully demonstrated by Google's quality raters and Metaweb (Kochhar, Mazzocchi, and Paritosh 2010), among others. At the other end of the spectrum, we can simply view the crowd as "HPU" (Davis et al. 2010) automata : given the current guidelines, we "turn the crank" and then inspect the output and look for problems. Based on any issues we find, we analyze and infer the cause, then revise the guidelines and repeat. Between these extremes lies an interesting design space of alternative labor models to explore. We can seek new ways to automate more of the work, better structure tasks and organize human workers, and try to optimize the overall man-machine annotation pipeline.

The minority voice. A final area for significant concern with crowd QC, particularly when automated, is how to distinguish rare insights from spurious noise? How can we recognize when the majority is wrong, when a single voice has recognized another valid interpretation based on their context? Crowdsourcing studies have often equated majority vote with quality, or that agreement with an expert is definitive. Can we learn to better recognize when other or better truths exist? Crowdsourcing is lauded for the diversity of opinions it can give voice to, yet we must find new ways to listen for those voices to be heard. Whereas with only one or two annotators we clearly understood the lack of diversity, a risk with crowdsourcing is assuming our crowdsourced data encompasses broad diversity when our QC process is systematically eliminating it. One should also examine crowd demographics (Ross et al. 2010) to better understand how diverse and representative our crowd and workers really are.

Machine Learning (ML)

Crowdsourcing has attracted much attention in the research community by enabling data, particularly labeled data, to be obtained more quickly, cheaply, and easily (Snow et al. 2008). While crowdsourcing is most often understood as yielding noisier data than traditional annotation practices, it also has potential for the exact opposite, reducing annotation bias through greater employing a larger and more diverse set of annotators for the same cost. Crowdsourcing's initial and primary continuing appeal has been largely as a tool for better data collection, providing researchers with a means of gaining new traction on pre-existing problems. However, many researchers first drawn to crowdsourcing as mere users have found unexpected and interesting challenges, such as QC, and led them to study crowdsourcing in its own right.

More labeled data. Supervised learning methods have historically outperformed unsupervised methods on the same task (since providing a learner with more information can intuitively enable it to more easily learn a desired pattern). In recent years, however, we have seen this trend reverse due to massive growth in Web content providing unsupervised and semi-supervised methods with free and seemingly limitless training data. This trend, along with an observation that magnitude of training data seems more important than model sophistication, has been characterized as the *unreasonable effectiveness of data* (Halevy, Norvig, and Pereira 2009). Crowdsourcing has now introduced another potentially disruptive shift since far more labeled data has suddenly become practically obtainable than was previously.

How might access to this new volume of labeled data alter the balance in which we utilize supervised, semi-supervised, and unsupervised methods? We might revisit the supervised learning curves for various tasks as a function of labeled data quantity, specifically seeking points where traditional costs associated with acquiring labeled data formerly led us to abandon supervised methods and exploit unlabeled data instead. To reach a target accuracy, is it now more cost-effective to proceed further along those supervised learning curves before moving to unlabeled data? Can we reach new accuracy levels by simply labeling more data (a labeled corrollary to the unreasonable effectiveness of data)? To what degree does relative benefit from unlabeled data diminish when we can saturate our supervised learners with labeled data? What impact will an order of magnitude more labeled data have on how we build practical learning systems?

More hybrid systems. Traditionally there has been a wide gap between automated accuracy and human accuracy for various tasks, with published research progressively demonstrating incremental improvement toward closing the gap. However, new hybrid systems that blend human computation with automation provide an opportunity in some cases to immediately close such gaps, calling upon crowd workers in near real-time to supply key judgments or interventions to supplement limitations of automation, e.g. where automated predictions are most uncertain or certain examples are critically important (Yan, Kumar, and Ganesan 2010).

As such, we have suddenly enlarged the design space for application developers. Matching human-level competence in an application is no longer simply a futuristic research goal, but is now a practical reality that is achievable at a certain cost tradeoff (e.g. time and expense) which can be navigated in the design space as a function of task context and user need. We can continue to analyze system behavior in terms of accuracy vs. time vs. cost tradeoff space, but instead of paying only for storage and CPU cycles, we may pay for human computation as well. The choice is ours. Moreover, beyond matching human-level accuracy, hybrid systems further create new opportunities to exceed our former limitations, amplifying human cognition with new forms of human-computer interaction and task decomposition. The capacities of our hybrid systems may exceed the sum of their parts, augmenting existing capabilities of man and machine.

More uncertain data. While we have always had noise in labeled data, in-house annotation and QC typically produced low error rates (as measured by inner-annotator agreement) which could be largely ignored for training and evaluating our learning systems. As mentioned in the previous section on QC, recent ML work on *consensus* has sought to transform relatively noisy crowd labels into high quality in-house labels though automated QC (Whitehill et al. 2009). The advantage of this approach is it creates a separation of concerns between QC and learning; QC fixes the data, and ML can go on training and evaluating learnings systems as we typically have before, assuming labels are reliable. The disadvantage of this approach, however, is that label disagreements between workers may arise from a combination of factors, including genuine uncertainty as to the correct label to be assigned. The cost of simplicity with consensus is that it masks such underlying uncertainty in the data, which is contrary to a fundamental tenet of AI that uncertainty should be modeled, propagated, and exposed. Discarding uncertainty via consensus precludes the system any possibilty of recovering later from mistakes made early by QC preprocessing.

An alternative to consensus is to model uncertainty of labels as a first class citizen in our learning systems. This means training and evaluating our systems on noisy labels via explicit or implicit probability distribution over possible labels for each example (Smyth et al. 1995). Such a distribution could be used to simply weight the example-specific gain/loss accorded to the system for predicting a given example correctly. If the system estimates a probability distribution over possible predictions, evaluation can measure distributional similarity between predicted vs. empirical label distributions for each example (*class-based estimation*).

More diverse data. Costs and effort involved with traditional annotation often restricted work to one or two workers, increasing the likelihood that labels produced would exhibit a particular bias. With crowd labeling, however, we stand to benefit from getting many more sets of eyes on the same data and thereby providing a greater diversity of labels. Such diversity is particularly valuable in subjective labeling tasks such as relevance assessment for Web search, in which case the same search query may be given by different users to express different information needs and intents (Alonso, Rose, and Stewart 2008). Another example is social tagging, in which different people often associate different labels with the same item.

One interesting challenge related to diversity and *wisdom of crowds* (WoC) (Surowiecki 2004) is to what extent we can develop a more formal, computational understanding of WoC (Wallach and Vaughan 2010). WoC's requirements of diversity, independence, decentralization, and aggregation are suggestive of methodology for effective, representative sampling. For example, uniform random sampling provides stochastic guarantees that with larger sample sizes we will capture greater diversity in the population and thereby a more representative sample. We can then infer a more representative statistic or distribution for the population (aggregation) by selecting or combining elements from the sample. Can we articulate a more precise understanding of WoC via computational and statistical principles?

Moreover, to what extent can we connect disparate knowledge of WoC, ensemble learning, and consenus methods? It seems that despite our best efforts to create a single, effective learner, we often achieve the best results by cobbling together a diverse ensemble of independent learners of varying abilities and aggregating their predictions (Bell and Koren 2007). We have theory and empirical experience from co-training, for example, relating the importance of independent learners to the effectiveness of the combined ensemble (Blum and Mitchell 1998). On the other hand, we have also seen that using less diverse strong learners outperformed an ensemble of models dumbed-down for the sake of promoting diversity (Gashler, Giraud-Carrier, and Martinez 2008). ML consensus work has to date been largely divorced from work in ensemble learning, despite similarities of both compensating for weakness of individual models/workers via plurality of redundant computation (machine or human). Both combine labels from multiple independent, weak annotators to arrive at a more accurate single labeling.

More specific data. While on one hand crowdsourcing enables us to collect more diverse labels by virtue of a diverse population of crowd workers, via active learning it also creates an opportunity for better focusing annotation effort on the examples that will be most informative to the learner. While active learning is not new (Settles 2009), historically it was often investigated in the context of iterative interaction between the system and one or few expert annotators (or simulated by selecting pre-labeled examples rather than labeling arbitrary new examples). As a result, it was often slow, expensive, and studied at relatively small scales. Moreover, because feedback came from reliable in-house annotators, noiseless "oracle" feedback was often assumed. Because active learning requires balancing costs of computation time (example selection vs. training) vs. human time (labeling), traditional costs associated with in-house annotation were factored in. Active learning has also often involved researchers labeling data themselves, making it often preferable in practice to keep annotation and learning separate.

With crowdsourcing, active learning can now be practically applied at scale and with frequent interactions between system and crowd labelers. Depending on how QC is handled, active learning must also contend with noisy labels: we must estimate not only how informative a given example will be to the system, but also how likely it is to be annotated correctly (example difficulty), and the cost to the system if the supplied label is incorrect. Sometimes it may be optimal to select an example to label which while being less informative to the system has greater expected probability of being labeled correctly (Brew and Cunningham 2010). As in some question answering systems (Horowitz and Kamvar 2010), the learner might also try to perform routing: predicting which example should be assigned to which annotator. Such routing may depend on a variety of factors: expertise of the worker relative to the given example, the importance of the example, the accuracy and cost of the worker, etc.

The typical benefits championed with active learning is faster learning curves relative to time and cost. One important benefit of crowdsourced active learning will simply be greater ability to realize such benefits in practice. Even if crowd workers were paid the same as traditional annotators, the convenient access to online workers to perform the labeling will by itself make active learning far more attractive and usable in both research and industry. In this sense, crowdsourcing provides a two-part cost savings: greater ability to realize traditionally-claimed savings of active learning, as well as reduced cost of crowd annotation vs. traditional annotators. A second important benefit will again be the implications for use of labeled vs. unlabeled data for training when labeled data is plentiful. Instead of comparing to past supervised learning curves, we might instead consider past learning curves for active learning, which will be steeper in comparison. As many prior studies in active learning considered smaller training sizes due to traditional costs of labeling, there may be greater potential for active learning than supervised learning to benefit from crowdsourcing.

More ongoing data. Lifetime, continuous, never-ending learning may also be made easier by crowdsourcing. People routinely update their knowledge and hypotheses as new information becomes available, and our systems ought to do the same. We should distinguish here between collecting additional labels characterizing a stationary data distribution (larger sample size) vs. collecting fresh labels for a non-stationary distribution (adapting to change in the underlying data). For example, with a temporal distribution like a search engine's query stream, users are always searching for different things with an ever-evolving query language, so it is valuable being able to continually update the set of examples used rather than sticking to a fixed, slice-of-time dataset. The ease of mechanized integration between automation and crowdsourcing can facilitate a never-ending process by which crowd annotators supply or the system requests additional labels to further learning. This workflow is largely agnostic as to whether online or batch learning is employed; there is simply the question of volume of labels to be processed (based on cost and accuracy goals) and whether there is sufficient scalability in the learning algorithm and crowd workforce to meet the target volume.

More rapid data. The ability to rapidly obtain new labeled data on demand has a variety of implications.

- *Relative cost of researcher time*. Cheap and rapid data annotation is akin to cheap and fast computation: it changes the way you think about spending your time and mental energy. Rather than speculate about and debate the potential benefit of two alternative approaches (expensive researcher time), one can simply try both and let the data speak for itself. Researcher time can then be focused instead on harder problems for which cheap computation and data collection cannot so easily solve.
- *Rapidly solving new problems.* When an important new problem or task presents itself, we typically think along the lines of data re-use: is there an existing dataset that is closely aligned or that could be easily adapted to it? If not, we often turn to unsupervised approaches rather than creating new labeled data. This is especially true when there is a time constraint driving us to address the problem quickly. When one can rapidly and easily create new labeled data, we might instead simply do that and just as quickly deploy an existing learner trained on the new data.
- *Rapid evaluation and tuning*. Automatic machine translation (MT) was typically slow to develop and evaluate since it required human judges to evaluate quality of system translations. The introduction of Bleu scoring (Papineni et al. 2002) enabled rapid turn-around in system tuning since evaluation was now fully-automatic. However, no automated proxy metric is a perfect substitute for human judging, so faster and cheaper human judging is still very valuable. Crowdsourcing offers a middle way between traditional judging and proxy metrics, with far faster turn-around than the former with potentially equal quality. Thus we can evaluate both more rapidly than traditionally and more accurately than with proxies. Fast, cheap evaluation lets us be more creative and daring in exploring a wider variety of system configurations.

More on-demand evaluation. Instead of evaluating a system against some fixed dataset, we can evaluate it instead against the crowd. Search engine evaluation often evaluates systems in terms of their accuracy on some subset of the top-ranked webpages they return (rather than the entire Web, which would be impossible to label in entirety). Re-usable test collections have been created in the past by pooling the top-ranked documents from many systems and labeling those, assuming all other documents are not relevant (Voorhees 2002). A significant problem we have encountered with ever-increasing collection sizes like the Web is that we cannot form a large enough pool in this manner to find a sufficient proportion of relevant documents to make the collection reusable for other systems which did not participate in the pooling. Another limitation of this approach is dependence on community to build a single test collection.

Crowdsourcing enables us to rapidly and easily label new examples on-demand for evaluation. When system tweaking changes the set of highly ranked webpages, we can simply label the new examples. This requires solid QC in place so labels are reliable, accounting for unmatched samples in evaluating significance of observed differences. Memoization let us cache each new label so we can score the system again later on the same examples with confidence of label consistency for idempotent evaluation (Little et al. 2009).

Less re-use. Since labeled data has traditionally been difficult to obtain, data re-use has often shaped both our methodology and the set of problems we choose to work on.

- *Benchmarks*. Established datasets serve a valuable role in enabling comparable evaluation across institutions and gauging overall progress by the field. On the other hand, dogged tuning on the same datasets year after year raises significant concerns of over-fitting. Crowdsourcing makes it easier to continually create new, diverse benchmarks for training and evaluation. The risk of such ease, however, is that everyone creates and experiments upon their own datasets, making it more difficult to compare alternative methods for the same task. Such is largely the case with ML work on consensus, in which many groups have evaluated on their own crowdsourced datasets. Lack of common benchmarks has traditionally been one of the main motivators for shared task evaluations.
- *Shared tasks*. Besides enabling comparable evaluation, shared tasks have also served as an important source of creating new, reusable labeled data for the community. Such data benefits both immediate participants (for whom

early data access motivates participation), as well as subsequent dataset re-use in the community. Because of the traditional costs and effort of annotation, shared tasks served a valuable role by amortizing such costs across the community, building one datasets for use by all. In some cases this has required each team to pitch-in and label a portion of the data themselves. With crowdsourcing, the benefit of community-based data production is diminished, as is the need for (and willingness of) participating research groups to perform such labeling themselves.

- *New tasks*. While creation of new tasks is not an end in-and-of-itself, which problems we choose to work on is partially a function of both their difficulty and importance (where difficulty can both spur work as well as deter it). When labeled data is easy to come by, a variety of problems become less difficult. This impacts not only which problems researchers choose to work on, but also the learning methodology they use to approach them.

Conclusion

We live in exciting times. For anyone who hates *information overload*, crowdsourcing will exacerbate it while also providing new tools for handling it. For those who love *big data*, crowdsourcing will give us more of it, change its nature, and challenge us to rethink how we can best utilize it.

A variety of interesting challenges remain to be faced. All the reasons we might normally prefer automation over manual labor will still apply and challenge us to think creatively about when and how human effort and interventions should be employed. Can we scale crowd labor to meet this growing demand, especially for applications requiring greater human expertise, or when greater privacy, security, or intellectual property concerns are present? Will pay-based crowd work be sustainable at volume and how might the market and pricing evolve? How can we increasingly leverage human computation without dehumanizing our workers? How can we effectively decompose work and simplify cognitive load while protecting workers from de-contextualized or misleading work in which they cannot assess their role for informed consent? Can computational principles and understanding help inform human education and innovation? Computational wisdom of crowds (WoC) and ensemble thinking may help us better understand how to mine and aggregate human wisdom, while active learning theory may provide new insights for more rapidly training our HPUs (Davis et al. 2010) to perform focused tasks. An interesting future awaits, with hopefully at least one or two more workshops (Adar 2011).

Acknowledgments. We thank the anonymous reviewers for their valuable feedback. This work was partially supported by a John P. Commons Fellowship for the author.

References

Adar, E. 2011. Why I Hate Mechanical Turk Research (and Workshops). *ACM Conference on Human Factors in Computing Systems (CHI) Workshop on Crowdsourcing and Human Computation*.

Alonso, O.; Rose, D.; and Stewart, B. 2008. Crowdsourcing for relevance evaluation. *ACM SIGIR Forum* 42(2):9–15.

Bell, R., and Koren, Y. 2007. Lessons from the Netflix prize challenge. *ACM SIGKDD Explorations Newsletter* 9(2):75–79.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory (COLT)*, 92–100.

Brew, A., and Cunningham, P. 2010. The interaction between supervised learning and crowdsourcing. In *NIPS Workshop on Computational Social Science and the Wisdom of the Crowds*.

Davis, J.; Arderiu, J.; Lin, H.; Nevins, Z.; Schuon, S.; Gallo, O.; and Yang, M. 2010. The HPU. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 9–16.

Gashler, M.; Giraud-Carrier, C.; and Martinez, T. 2008. Decision tree ensemble: small heterogeneous is better than large homogeneous. In *Machine Learning and Applications, 2008. ICMLA'08. Seventh International Conference on*, 900–905. IEEE.

Halevy, A.; Norvig, P.; and Pereira, F. 2009. The unreasonable effectiveness of data. *Intelligent Systems, IEEE* 24(2):8–12.

Horowitz, D., and Kamvar, S. 2010. The anatomy of a large-scale social search engine. In *Proceedings of WWW*, 431–440. ACM.

Ipeirotis, P. G. 2011. The unreasonable effectiveness of simplicity. February 6. `http://behind-theenemy-lines.blogspot.com/2011/02/unreasonableeffectiveness-of.html`.

Kochhar, S.; Mazzocchi, S.; and Paritosh, P. 2010. The anatomy of a large-scale human computation engine. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, 10–17. ACM.

Linguistic Data Consortium. `http://www.ldc.upenn.edu`.

Little, G.; Chilton, L.; Goldman, M.; and Miller, R. 2009. Turkit: Tools for iterative tasks on mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, 29–30. ACM.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the ACL*, 311–318.

Ross, J.; Irani, L.; Silberman, M.; Zaldivar, A.; and Tomlinson, B. 2010. Who are the crowdworkers?: shifting demographics in mechanical turk. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, 2863–2872. ACM.

Settles, B. 2009. Active Learning Literature Survey. Technical Report 1648, University of Wisconsin-Madison Computer Sciences.

Smyth, P.; Fayyad, U.; Burl, M.; Perona, P.; and Baldi, P. 1995. Inferring ground truth from subjective labelling of venus images. *Advances in neural information processing systems* 1085–1092.

Snow, R.; O'Connor, B.; Jurafsky, D.; and Ng, A. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 254–263.

Surowiecki, J. 2004. The Wisdom of Crowds.

von Ahn, L. 2005. *Human Computation*. Ph.D. Dissertation, Carnegie Mellon University. Tech. Report CMU-CS-05-193.

Voorhees, E. 2002. The philosophy of information retrieval evaluation. In *Evaluation of cross-language information retrieval systems*, 143–170.

Wallach, H., and Vaughan, J. W. 2010. Workshop on Computational Social Science and the Wisdom of Crowds. In *NIPS*.

Whitehill, J.; Ruvolo, P.; Wu, T.; Bergsma, J.; and Movellan, J. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, 2035–2043.

Yan, T.; Kumar, V.; and Ganesan, D. 2010. CrowdSearch: exploiting crowds for accurate real-time image search on mobile phones. In *MobiSys*, 77–90.