Towards Building a High-Quality Workforce with Mechanical Turk

Paul Wais, Shivaram Lingamneni, Duncan Cook, Jason Fennell,
Benjamin Goldenberg, Daniel Lubarov, David Marin, and Hari Simons*
Yelp, Inc.
706 Mission Street
San Francisco, CA 94103
{pwais,shivaram,duncan,daniel,jfennell,benjamin,dave,hari}@yelp.com

Abstract

Online crowdsourcing services provide an inexpensive and scalable platform for large-scale information verification tasks. We present our experiences using Amazon's Mechanical Turk (AMT) sto verify over 100,000 local business listings for an online directory. We compare the performance of AMT workers to that of experts across five different types of tasks and find that most workers do *not* contribute high-quality work. We present the results of preliminary experiments that work towards filtering low-quality workers and increasing overall workforce accuracy. Finally, we directly compare workers' accuracy on business categorization tasks against a Naïve Bayes classifier trained from user-contributed business reviews and find that the classifier outperforms our workforce. Our report aims to inform the community of empirical results and cost constraints that are critical to understanding the problem of quality control in crowdsourcing systems.

1 Introduction

Crowdsourcing services such as Amazon Mechanical Turk[1] (AMT) provide a useful platform for processing large quantities of simple tasks. However, many publicized experiences with AMT indicate that work quality does not scale with quantity. In this report, we discuss an AMT-based system that has processed well over 100,000 business listing changes (see Table 1). During the lifetime of the system, we observed a high frequency of errors. For example, workers would accept incorrect phone numbers and inappropriate categories (e.g. the "Aquariums" category for a seafood restaurant). Our goal is to design a mechanism for filtering low-quality workers in order to build a reliable workforce that has high accuracy. (For our particular application, we desire at least 90% accuracy to justify continued use of AMT). Furthermore, we wish to minimize both the cost and time needed to filter workers.

There are several studies that propose techniques for identifying and filtering the work of biased or erroneous workers. Ipeirotis et al [1] devise an EM-based algorithm for scoring the bias of workers. Their synthetic results show that a ssmall workforce with 90% accuracy may be trained by assigning at least 10 workers per task and recording 60 labels per worker. Snow et all [2] present a similar Bayesian algorithm that corrects for worker bias and recovers 4.0% in accuracy when assigning 10 workers per task. Dekel and Shamir [3] devise a technique for pruning the labels of erroneous workers in order to minimize the error of an SVM trained from worker labels. Their AMT-based experiment shows that their technique improves accuracy up to 12% on a label set created using at least 15 worker labels per example. Note that each of these approaches requires at least 10 workers to process a single task, resulting in a costly training process.

[*] The majority of the work presented here was completed while this author was at Yelp, Inc.

[1] http://www.mturk.com

Table 1: Summary of business listing changes processed

Task Type	Real Tasks	Test Tasks	USD per Task per Worker
Phone Number	28131	1548	0.05
Category	138222	3185	0.05
Hours	93336	5080	0.05
Website URL	88222	9834	0.025
Address	47962	3088	0.05
New Business	82872	—	0.17

Given the large number of changes our application needs to process, financial constraints require us to assign *at most three* workers per real business change. Although AMT provides a very inexpensive work platform, we find that is it financially impractical to assign more than this many workers per task for tens of thousands of tasks. In particular, our experience is that we were able hire a team of on-site experts for not much more than the cost of our AMT-based system. A successful crowdsourcing system must be able to achieve high accuracy with fewer workers than previously studied. In this study, we explore first assigning workers to test tasks and then selecting the most accurate workers to complete real tasks. From a pool of 4,660 applicants, 79 workers achieved high enough accuracy on our test tasks to qualify to work on real tasks. The remainder of this paper reviews our tasks in detail, discusses techniques for filtering the work of erroneous workers, and directly compares workforce performance with a Naïve Bayes classifer for our business categorization task.

2 Task design

Our system supports six different kinds of business listing changes. The Phone Number, Hours, and Address tasks ask workers to choose between two candidate listings or to reject both candidates. The Category and Website URL tasks ask workers to accept or reject proposed categories or links to official business websites. The New Business task asks workers to verify several components of a basic business listing. We expect workers to verify changes by calling the business or finding the correct information on the business's official website. Each task provides workers with as much information as possible to help them decide. For each task, we observed a median worker completion time of about 5 minutes.

In this study, we discuss the performance of workers on *test* tasks derived from expert-verified business listing changes and directly compare workers' agreement with the decisions of experts. Though we had access to plenty of expert-approved changes to serve as the basis for test tasks, expert-*rejected* changes were rare for some tasks, so we synthesized many rejection tasks. For example, for a synthetic Address rejection task we use the address of a different verified business located within a few miles of the test business. We chose not to investigate test tasks for the New Business type due to difficulties in synthesizing realistic rejectable changes. For our experiment, we circulated tens of thousands of test tasks on AMT and allowed up to 15 workers to complete each task. We always accepted work and paid an additional 5 cent bonus only if the worker provided the correct label for the test change. We very briefly experimented with doubling the bonus but did not observe any significant changes in accuracy.

3 Vetting competent workers

Before a worker may complete test or real tasks, she or he must pass our AMT *Qualification Test*. The test provides workers with instructions and background information for completing tasks and asks workers to complete eight multiple choice questions that are very similar in content to the real and test business listing tasks. We designed the test based upon specific examples of poor quality work that we observed prior to the test.

Of the 4,660 workers who took this test, only 1,658 (35.6%) workers earned a passing score, and over 25% of workers answered fewer than half of the questions correctly. To investigate the high failure rate, we conversed with workers directly on TurkerNation[2] and through private email. Based upon worker's names and email addresses, we believe that we conversed with a representative sample of workers both inside and outside the United States. We found that the test was not too difficult and

[2] http://www.turkernation.com

that most workers comprehended the questions. We believe that many applicants simply try to gain access to tasks as quickly as possible and do not actually put care into completing the test.

4 Improving workforce accuracy

In this section we discuss empirical results of workforce accuracy and techniques for filtering poor-quality work. Figure 1 shows that indeed most workers exhibit rather poor accuracy on even our simple listing verification tasks.

In Table 2, we present the results of using majority consensus[3] among workers as well as historical worker accuracy in an effort to filter low-quality work. For Category and Website URL tasks, accepting the label that has a simple majority of votes provides the greatest accuracy. For other tasks, ignoring consensus completely and accepting the label of the worker with the highest task-accuracy yields the best overall accuracy. We tested a combination of these methods where we choose either the label of the highest-accuracy worker in the absence of at least a $\frac{2}{3}$ consensus or otherwise we choose the consensus label. Surprisingly, we find that this mechanism does not improve overall accuracy on any of the tasks. We hypothesize that there are many "hard" tasks that confuse most workers, including the good workers.

We studied the relationship between completion time and accuracy but found no strong correlation. We calculated the Pearson's correlation between mean task completion times and task accuracies. The Hours task exhibited the strongest (though least probable) correlation ($r = 0.45, p = 0.0007$) and the Address task observed the weakest ($r = 0.08, p = 0.636$). The median completion time for each task was less than 5 minutes; however, many workers took 30 minutes or longer to complete some tasks and exhibited very inconsistent completion times. In particular, about 33% of our workers did not complete all tasks within 3 standard deviations of their mean completion time. We believe that some workers will employ a strategy of "accepting" a task several minutes or hours before they start working on the task due to competition for simple tasks on AMT.

We also studied the relationship between worker accuracy and locale. Ross et al conducted a survey of AMT workers and found that at least 40% of workers lived outside the U.S. as of November 2009 [4]. Unfortunately, since AMT does not allow requesters to query for a worker's locale, we were not able to recover exact locale data for our analysis, so we used task submission time-of-day in order to estimate locale. We observed a slight negative Pearson's correlation ($r = -0.004, p = 0.933$) between accuracy and task submission time for workers who had completed at least 10 tasks. We did not further investigate using features of AMT to restrict our test tasks to specific locales because we did not want to limit our overall throughput.

We hypothesize that many workers do poorly because they find the tasks boring and "cruise" through them as quickly as possible. The high frequency of 50% accuracy workers on the Website URL task suggests that most workers simply check the first radio box they see. In the future we plan to implement a more interactive interface that may increase worker engagement (e.g. using a framework such as TurkIt [5]).

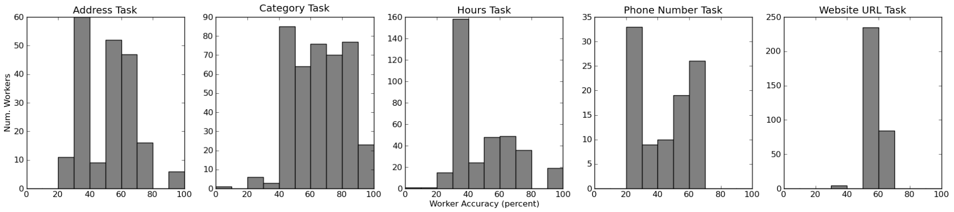


Figure 1: Worker accuracies by task. We omit accuracies for workers who completed fewer than three tasks of a single type. Workers exhibit greatest accuracy on the Category task, where the 90$^{\mathrm{th}}$ percentile accuracy rate is 86%.

5 Workers versus Naïve Bayes category classifier

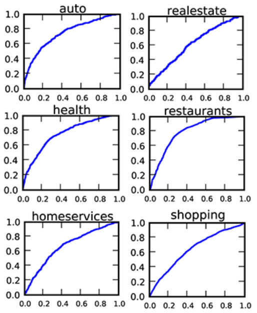
In some cases, crowdsourcable tasks may also be fulfilled using standard machine learning techniques. In this section, we directly compare the accuracy of a preliminary Naïve Bayes classifier

[3] In order to prevent "accidental" consensus, the user interface randomizes the order of radio button choices.

Table 2: Using consensus and worker accuracy to improve overall accuracy

Task Type	Consensus	Best Worker	2/3 Consensus + Best Worker
Phone Number	73.3%	**89.0%**	80.9%
Category	**79.0%**	66.9%	77.9%
Hours	59.1%	**76.0%**	67.3%
Website URL	**92.7%**	75.0%	89.8%
Address	68.5%	**77.7%**	73.4%

on the Category task with the accuracy of our workforce. Our Naïve Bayes model is an adaption of the spam e-mail filtering model first proposed by Sahami et al [6] and first applied to our domain by Fennell et all [7]. The model chooses category $C^\star$ that maximizes the posterior probability $C^\star = \operatorname{argmax}_i P(C_i) \prod_{j=1}^{n} P(w_j|C_i)$, where $P(C_i)$ is the frequency of category C_i in the training set, and we estimate the likelihood $P(w_j|C_i)$ using the frequency of word w_j in reviews of businesses with category C_i. For the purposes of this comparison, we considered the top three high-probability categories calculated from the model. We trained the model using over 12 million user-contributed reviews from Yelp[4]. To improve accuracy, our implementation stems review text using the Porter stemmer [8], ignores stop words irrelevant to our domain, and applies heuristics for pruning high-likelihood categories that we found the Naïve Bayes model often conflated (e.g. the "Car Rental" and "Automotive" categories).



(a) Receiver operating characteristic curves for Naïve Bayes classifier.

Category	Workers			Naïve Bayes	
	Accept	Reject	Overall	Accept	Reject
Automotive	59.9%	77.8%	*65.8%*	**80.5%**	**82.1%**
Health	56.6%	75.4%	*58.1%*	**80.0%**	**78.5%**
Home Services	50.1%	65.0%	*57.7%*	**74.2%**	**75.0%**
Real Estate	35.8%	50.0%	*68.6%*	**74.5%**	**77.3%**
Restaurants	60.3%	78.4%	*60.9%*	**88.1%**	**87.9%**
Shopping	51.5%	67.1%	*58.7%*	**76.8%**	**78.4%**

(b) Comparison of worker and Naïve Bayes accuracies on 1,615 categorization tasks. Recall that category tasks ask workers to accept or reject a set of specific categories for a business.

Figure 2: Comparison of worker and Naïve Bayes performance on categorization tasks.

Figure 2(a) shows the Naïve Bayes classifier's ability to discriminate categories on a test set of 3,183 businesses. We select 1,615 categorization tasks associated with businesses from this test set having the six categories listed in Figure 2(b). The selected categories are the most important to the Yelp online directory. We directly compare the categorization accuracies of the workers and the classifier on the same tasks and find that the classifier outperforms the workforce. We note that for some categories workers tend to have slightly worse accuracy on this test set than on the overall set of category tasks. Moreover, workers are substantially more accurate at rejecting categories than accepting them. We believe that most of the rejectable category changes are blatently incorrect, so it may be easier for workers to identify changes that appear immediately inappropriate. Though the Naïve Bayes classifier exhibits the best performance, in future work we may use AMT to help train or improve upon noisy expert category labels (e.g. using techniques described in Sheng et al [9]). Finally, we note that some tasks are more amenable to crowdsourcing (e.g. the Phone Number and Hours tasks) than machine learning techniques.

6 Conclusion

In this paper, we present empirical results that illustrate the scope of the challenges facing crowdsourcing quality control. Using a combination of pre-screening and the test tasks described above, only 79 workers of 4,660 applicants qualified to process real business changes. Though this filtering improved our workforce accuracy, we were unable to achieve our throughput or financial goals. Future work must aim to more quickly identify and filter erroneous workers in order to reduce costs.

[4] http://www.yelp.com

Acknowledgments

We are very grateful to Jennifer Wortman Vaughan for her helpful comments on a draft of this report. We would also like to thank the anonymous reviewers for their valuable comments and suggestions for improving the quality of this paper.

References

[1] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on amazon mechanical turk," in *HCOMP '10: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2010, pp. 64–67.

[2] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks," in *EMNLP*, 2008, pp. 254–263.

[3] O. Dekel and O. Shamir, "Vox populi: Collecting high-quality labels from a crowd," in *In Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

[4] J. Ross, L. Irani, M. S. Silberman, A. Zaldivar, and B. Tomlinson, "Who are the crowdworkers?: shifting demographics in mechanical turk," in *CHI Extended Abstracts*, 2010, pp. 2863–2872.

[5] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller, "Turkit: tools for iterative tasks on mechanical turk," in *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*. New York, NY, USA: ACM, 2009, pp. 29–30.

[6] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A bayesian approach to filtering junk e-mail," in *AAAI Workshop on Learning for Text Categorization*, 1998.

[7] J. Fennell, K. Shiells, and B. Sitaraman, "Auto-categorization of businesses on yelp.com," 2009. [Online]. Available: http://nlp.stanford.edu/courses/cs224n/2009/fp/15.pdf

[8] M. F. Porter, *An algorithm for suffix stripping*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 313–316.

[9] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2008, pp. 614–622.