A Study on the Influence of the Number of MTurkers on the Quality of the Aggregate Output

Arthur Carvalho[1], Stanko Dimitrov[2], and Kate Larson[3]

[1] Rotterdam School of Management, Erasmus University, Rotterdam, Netherlands,
carvalho@rsm.nl

[2] Department of Management Sciences, University of Waterloo, Waterloo, Canada,
sdimitro@uwaterloo.ca

[3] Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada,
kate.larson@uwaterloo.ca

Abstract. Recent years have seen an increased interest in crowdsourcing as a way of obtaining information from a large group of workers at a reduced cost. In general, there are arguments for and against using multiple workers to perform a task. On the positive side, multiple workers bring different perspectives to the process, which may result in a more accurate aggregate output since biases of individual judgments might offset each other. On the other hand, a larger population of workers is more likely to have a higher concentration of poor workers, which might bring down the quality of the aggregate output.

In this paper, we empirically investigate how the number of workers on the crowdsourcing platform Amazon Mechanical Turk influences the quality of the aggregate output in a content-analysis task. We find that both the expected error in the aggregate output as well as the risk of a poor combination of workers decrease as the number of workers increases. Moreover, our results show that restricting the population of workers to up to the overall top 40% workers is likely to produce more accurate aggregate outputs, whereas removing up to the overall worst 40% workers can actually make the aggregate output less accurate. We find that this result holds due to top-performing workers being consistent across multiple tasks, whereas worst-performing workers tend to be inconsistent. Our results thus contribute to a better understanding of, and provide valuable insights into, how to design more effective crowdsourcing processes.

1 Introduction

Recent technological advances have facilitated the outsourcing of a variety of tasks to "the crowd", *e.g.*, the development and testing of large software applications, the design of websites, professional translation of documents, transcription of audio, *etc.* Such a practice of obtaining relevant information or services from a large group of people is traditionally referred to as *crowdsourcing*.

The crowdsourcing process, as considered in this paper, is as follows: a number of workers are asked to individually complete a common *task*. After completing the task, each worker must report back an *output*. The reported outputs are then aggregated to obtain an *aggregate output*. A crucial question that arises during this process is how many workers to include in the task. In particular, how does the number of workers influence the quality of the aggregate output?

Arguments can be made in favor and against the use of multiple workers. On the one hand, multiple workers bring diversity to the process so that biases of individual judgments can offset each other, which may result in a more accurate aggregate output. On the other hand, a larger population of workers might bring down the quality of aggregate outputs due to the likely inclusion of poor workers.

In this paper, we empirically investigate the above question through an experiment using one of the most popular crowdsourcing platforms: *Amazon Mechanical Turk* (AMT). In our experiment, we asked workers to solve three content-analysis tasks. Due to the nature of the tasks, we are able to derive *gold-standard outputs*, *i.e.*, outputs of high quality provided by experts with relevant expertise.

The existence of gold-standard outputs allows us to investigate how different combinations of workers affect the accuracy of aggregate outputs. We first analyze the accuracy of aggregate outputs as the number of workers increases. Focusing on simple averages to aggregate outputs, we find a substantial degree of improvement in expected accuracy as we increase the number of workers, with diminishing returns for extra workers. Moreover, the standard deviation of errors in the aggregate outputs decreases with more workers, which implies less risk when aggregating outputs.

Our experimental results also show that combining only the overall top-performing workers results in more accurate aggregate outputs, and these workers are consistent across multiple tasks. On the other hand, removing the overall worst-performing workers from the population of workers might result in less accurate aggregate outputs. The reason for this surprising result is that the overall worst-performing workers can produce good outputs on some tasks, which implies that they tend to be inconsistent across multiple tasks. Our results thus contribute to a better understanding on how to design more effective crowdsourcing processes.

2 Related Work

Many different research questions involving crowdsourcing have been recently addressed by the multi-agent systems community, *e.g.*, how to assign tasks to workers [5, 13], how to design optimal workflows to coordinate the work of the crowd [7, 15], how to induce honest behavior in crowdsourcing settings [1, 4], *etc.* We refer the interested readers to the papers by Yuen *et al.* [14] and Quinn and Bederson [10] for comprehensive surveys on crowdsourcing-related works.

To the best of our knowledge, this paper is the first one to address the question of how the number of workers affects the quality of aggregate outputs in crowdsourcing settings. Similar studies have been performed in different domains. For example, it is well-known in decision analysis and operations research that combining multiple forecasts often leads to improved forecasting performance [3]. Sheng *et al.* [11] showed in a data mining/machine learning domain that labeling the same data set with different "labelers" might sometimes improve data quality.

However, some unexpected results are apparently specific to crowdsourcing. For example, our experimental results show that removing the overall worst-performing workers from the population of workers might result in less accurate aggregate outputs. Thus, we expect our work to shed light on how to design more effective crowdsourcing processes.

3 The Content-Analysis Experiment

In this section, we describe a content-analysis experiment designed to investigate the question of how the number of workers affects the quality of aggregate outputs. In the following subsections, we describe Amazon Mechanical Turk, the crowdsourcing platform used in our experiments, followed by the experimental design.

3.1 Amazon Mechanical Turk

Amazon Mechanical Turk[4] (AMT) is currently one of the most popular crowdsourcing platforms. AMT has consistently attracted thousands of workers, the so called *MTurkers*, willing to complete hundreds of thousands of outsourced tasks for relatively low pay. Most tasks posted on AMT are tasks that are relatively easy for human beings, but nonetheless challenging or even currently impossible for computers, *e.g.*, audio transcription, filtering adult content, extracting data from images, proofreading texts, *etc.*

AMT has also been widely used as a platform for conducting behavioral experiments [8]. The main advantage that AMT offers to behavioral researchers is the access to a large, diverse, and stable pool of workers willing to participate in the experiments for relatively low pay, thus simplifying the recruitment process and allowing for faster iterations between developing theory and executing experiments. Furthermore, AMT provides a built-in reputation system that helps requesters distinguish between good and bad workers and, consequently, to ensure data quality. AMT also provides an easy-to-use built-in mechanism to pay workers that greatly reduces the difficulties of compensating individuals for their participation in the experiments.

3.2 Experimental Design

We asked workers on AMT to review three short texts under three different criteria: *Grammar*, *Clarity*, and *Relevance*. The first two texts were extracts

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

from published poems, but with some original words intentionally replaced by misspelled words. The third text contained random words presented in a semi-structured way. Appendix A contains detailed information about the texts. For each text, three questions were presented to the workers, each one having three possible responses ordered in decreasing negativity order:

- Grammar: does the text contain misspellings, syntax errors, *etc.*?
  - A lot of grammar mistakes
  - A few grammar mistakes
  - No grammar mistakes
- Clarity: does the text, as a whole, make any sense?
  - The text does not make sense
  - The text makes some sense
  - The text makes perfect sense
- Relevance: could the text be part of a poem related to love?
  - The text cannot be part of a love poem
  - The text might be part of a love poem
  - The text is definitely part of a love poem

Words with subjective meaning were intentionally used so as to emphasize the subjective nature of content analysis, *e.g.*, "a lot", "a few", *etc.* In order to conduct numerical analysis, each individual response was translated into a *score* inside the set $\{0, 1, 2\}$. The most negative response received the score 0, the middle response received the score 1, and the most positive response received the score 2. Thus, each worker reported a vector of 9 scores (3 criteria for each of the 3 texts). Henceforth, we denote by *output* a vector of 3 scores for a given text. Thus, each worker reported 3 outputs.

A total of 50 workers were recruited on AMT, all of them residing in the United States of America and older than 18 years old. They were required to accomplish the task in at most 20 minutes. After accomplishing the task, every worker received a payment of \$0.20. A study done by Ipeirotis [6] showed that more than 90% of the tasks on AMT have a baseline payment less than \$0.10, and 70% of the tasks have a baseline payment less than \$0.05. Thus, our baseline payment was much higher than the payment from the vast majority of other tasks posted on AMT.

Since the source and original content of each text were known *a priori*, *i.e.*, before the content-analysis experiment was conducted, we were able to derive gold-standard outputs for each text. In order to avoid confirmation bias, we asked five professors and tutors from the English and Literature Department at the University of Waterloo to provide their outputs for each text. We set the gold-standard score for each criterion in a text as the median of the scores reported by the professors and tutors. Coincidentally, each median value was also the mode of the underlying scores. We show the gold-standard outputs in Appendix A.

4 Accuracy of Aggregate Outputs by the Number of Workers

In this section, we study the influence of the number of workers on the quality of the aggregate output. In order to do so, we generated combinations of the 50 workers in our population. For $r \in \{1, \ldots, 4\}$ and $r \in \{46, \ldots, 50\}$, we calculated all possible combinations of workers, *i.e.*, $\binom{50}{r}$. For example, for $r = 2$, we generated all $\binom{50}{2} = 1225$ pairs of workers. Due to the intractable number of combinations for $r \in \{5, \ldots, 45\}$, we randomly generated 10^5 different combinations of workers for any $r \in \{5, \ldots, 45\}$.

For each combination of workers, we aggregated the outputs from the underlying workers by taking the average of them. For instance, for two workers, we calculated the average output for all $\binom{50}{2} = 1225$ possible pairs of workers.

We then measured the accuracy of each aggregate output. For each aggregate output, we calculated the *root-mean-square deviation* (RMSD) between the aggregate output and the gold-standard output. For example, suppose that a pair of workers report the outputs $(1, 2, 0)$ and $(2, 2, 1)$ for Text 1. Thus, the aggregate output is $(1.5, 2, 0.5)$. Given that the gold-standard output for Text 1 is $(1, 2, 2)$ (see Appendix A), the root-mean-square deviation between the aggregate output and the gold-standard output is:

$$\sqrt{\frac{(1.5-1)^2 + (2-2)^2 + (0.5-2)^2}{3}} \approx 0.9129$$

We denote by *error* the RMSD between the aggregate output and the gold-standard output. Clearly, the lower the error, the more accurate the aggregate output. In our experiments, the range of the error is $[0, 2]$. The resulting *average error* for a given r can be seen as the *expected error* when aggregating outputs using r workers. For instance, the average of the $\binom{50}{2} = 1225$ errors from all possible pairs of workers is the *expected error* when aggregating outputs using 2 workers chosen at random. The average error, the standard deviation of the errors, and the maximum error per text for each $r \in \{1, \ldots, 50\}$ are illustrated in Figure 1. The complete numerical data is shown in Appendix B.

An interesting feature of Figure 1 is that the influence of the number of workers on the quality of the aggregate output is qualitatively the same for all texts. That is, the average error decreases as the number of workers r increases, which means that the expected accuracy of the aggregate output increases with more workers.

Figure 2 shows the percentage of the reduction of the average error when one extra worker is added. From the starting point of one worker, adding a second worker reduces the average error by $3.6\% - 16.5\%$. Given two workers, adding a third worker decreases the average error by $2\% - 8.3\%$, and so on. Clearly, there are diminishing returns for extra workers. For example, while adding a fourth worker reduces the average error by $1.19\% - 4.79\%$, adding a tenth worker reduces the average error by only $0.07\% - 0.79\%$. After the sixth worker, adding another worker always decreases the average error by less than 2% for all texts.

Fig. 1. The average error, the standard deviation of the errors, and the maximum error per text for each $r \in \{1, \ldots, 50\}$.

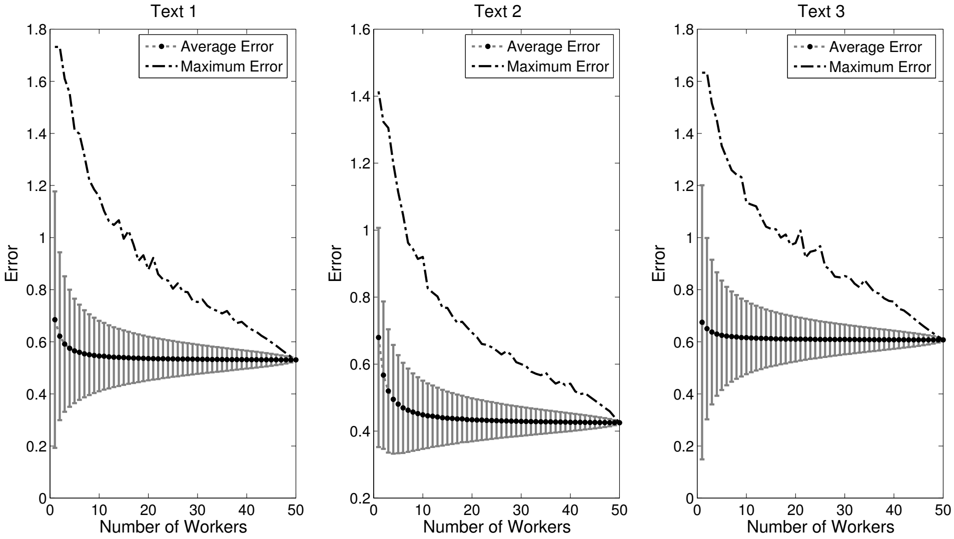


Fig. 2. The percentage of the reduction of the average error when one extra worker is added.

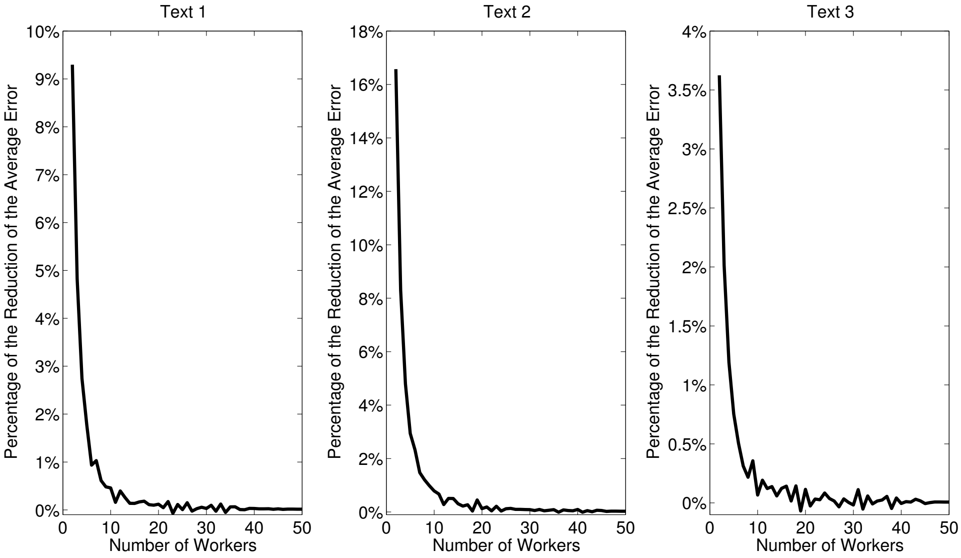


Figure 1 also shows that the standard deviation of the errors decreases with the number of workers r. The initially high standard deviation indicates an opportunity to get considerably low error with a single worker. Of course, the other side of the coin is a greater risk of high error with a single poor worker. As the number of workers increases, this risk decreases because combinations of exclusively poor workers become less likely. This fact is also shown in the reduction of the maximum error when r increases, which implies less risk when aggregating outputs.

5 Accuracy of Outputs from the Top Workers

The analysis performed in the previous section is based on combinations of workers from the full population of workers. Two interesting follow-up questions are: 1) how much can accuracy be improved by restricting attention to combinations of the overall top-performing workers? and 2) how much can accuracy be improved by removing the overall worst-performing workers from the population of workers?

In order to answer these questions, we first sorted workers based on the *overall error*. Recall that each worker reported three outputs, each one consisting of three scores. We denote by *overall output* a vector of all nine reported scores. Likewise, we denote by *overall gold-standard output* the vector of all nine scores from the gold-standard outputs. Then, the *overall error* of a worker is the RMSD between his overall output and the overall gold-standard output. For example, suppose that a worker reports the following outputs for Text 1, 2, and 3: $(1, 2, 2)$, $(1, 2, 0)$, and $(1, 0, 0)$. Hence, his overall output is $(1, 2, 2, 1, 2, 0, 1, 0, 0)$. Recall that the gold-standard outputs for Text 1, 2, and 3 are, respectively, $(1, 2, 2)$, $(1, 2, 1)$, and $(0, 0, 0)$. Thus, the overall gold-standard output is $(1, 2, 2, 1, 2, 1, 0, 0, 0)$. Consequently, the worker's overall error is:

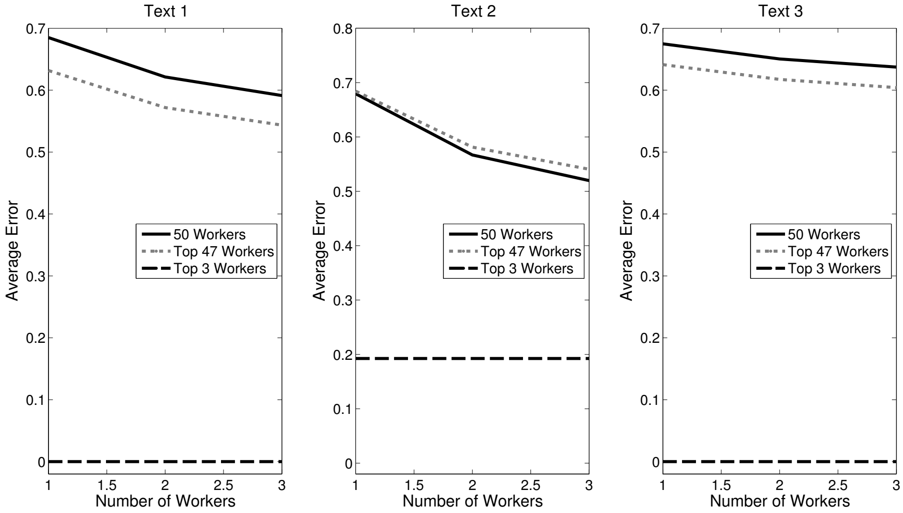
$$\sqrt{\frac{x}{9}} \approx 0.4714$$

where:

$$\begin{aligned} x =& (1-1)^2 + (2-2)^2 + (2-2)^2 + (1-1)^2 + (2-2)^2 + \\ & (0-1)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 \\ & = 2 \end{aligned}$$

For ease of exposition, in the following discussion we focus on the overall accuracy of the top 3 workers and on the accuracy of the population of workers without the 3 overall worst-performing workers, *i.e.*, the top 47 workers. We note, however, that the following results are qualitatively the same for the top k and the top $50 - k$ workers, for any $k \in \{2, \ldots, 20\}$. We return to this point later in this section, when we also suggest a different way of ordering workers.

Fig. 3. The average error per text for different populations of workers and $r \in \{1, 2, 3\}$.



After ordering workers in terms of overall errors, we considered all possible combinations of the top 3 workers, *i.e.*, we calculated the aggregate outputs and errors for all $\binom{3}{r}$ possible combinations of workers, for $r \in \{1, 2, 3\}$. Moreover, we removed the three overall worst-performing workers from the full population of workers and calculated the aggregate outputs and errors for all $\binom{47}{r}$ combinations of workers, for $r \in \{1, 2, 3\}$ in order to allow quantitative comparisons across different populations of workers. The resulting average error per text for different populations of workers is illustrated in Figure 3. The complete numerical data is shown in Table 1 in Appendix B.

Focusing first on Text 1 and 3, any combination of the top 3 workers results in a perfect aggregate output with zero error, whereas removing the three overall worst-performing workers reduces the average error by $4.96\% - 8.10\%$ in comparison with the complete population of workers, for the same group size $r \in \{1, 2, 3\}$.

Looking at the numerical values for Text 1 in Table 1 (see Appendix B), the average error for combinations of 1, 2, and 3 workers from the top 47 workers (*i.e.*, 0.632, 0.572, and 0.543) is less than the average error for combinations of 1, 4, and 11 workers from the complete population of workers (*i.e.*, 0.685, 0.575, and 0.544). In other words, the aggregate outputs of 1, 2, and 3 randomly selected workers from the top 47 workers are expected to be more accurate in Text 1 than the aggregate outputs of 1, 4, and 11 randomly selected workers from the complete population of workers. These numbers for Text 3 are, respectively, 2, 8, and 50. Thus, for Text 1 and 3, it is beneficial to remove some worst-performing workers from the full population of workers.

The striking result comes from Text 2, where the average error for the full population of workers is $0.69\% - 3.85\%$ *lower* than the average error for the top 47 workers. The reason for this counter-intuitive result is that there were workers amongst the three overall worst-performing workers who excelled in Text 2, while performing poorly in Text 1 and 3. This shows that some workers are not consistent across multiple tasks. We return to this point in the next section.

For all populations of workers, the average error, the standard deviation of the errors, and the maximum error decrease as the number of workers increases, showing that combining multiple workers is always beneficial since it improves accuracy and reduces risks.

As stated before, for ease of exposition, our discussion in this section has been focused on the implications of restricting the population of workers to the overall top 3 workers and of removing the three overall worst-performing workers from the full population of workers. The obtained results are, however, more general. Any combination of up to k workers, for $k \in \{1, \ldots, 20\}$, from the top k workers results in a lower average error than a combination of the same number of workers from both the complete population of workers and the top $50 - k$ workers. Moreover, removing any number $k \in \{2, \ldots, 20\}$ of worst-performing workers from the complete population of workers results in an increase of the average error for Text 2.

The above results are statistically significant for any $k \in \{3, \ldots, 20\}$ (rank-sum test, p-value ≤ 0.05). For combination of size $k \in \{1, 2\}$, the three populations of workers have many combinations of workers in common. In general, as k increases, the fraction of combinations of workers shared between the top k workers, the top $50-k$ workers, and the full population of workers decreases, thus allowing us to make stronger statistical comparisons. For example, for $k \geq 4$, the p-values from the rank-sum tests are approximately 0.

It could be argued that the results in this section hold true because our experimental setting is biased, *e.g.*, the overall top-performing workers are expected to be more accurate in all texts because the overall error contains information about errors from all individual texts. However, if such a bias existed, combinations of top-performing workers would always result in lower average errors than combinations of the same number of workers from the full population of workers, a fact which is not true for $k \in \{21, \ldots, 25\}$. For example, for $k \in \{23, 24, 25\}$ and Text 1 and 2, a random combination of workers from the complete population of workers results in a lower average error than a random combination of the same number of workers from both the top k workers and the top $50 - k$ workers. In general, we find no clear pattern for values of $k \in \{21, \ldots, 25\}$.

Another way to compute the overall error and, thus, of ranking workers is by using a leave-one-out cross-validation approach. That is, given n texts, each worker receives a *historical rank* based on his errors on $n - 1$ texts. Then, the performance of different populations of workers is measured on the left-out text. However, the leave-one-out cross-validation approach may not work well with small data sets, such as the one in this study. We tried this approach on our data set and had mixed results. For example, when defining workers' historical ranks based on their performance in Text 1 and 2, and measuring the performance of different populations of workers in Text 3, a random combination of workers from the top k workers resulted in *higher* average error than a random combination of the same number of workers from both the full population of workers and the top $50 - k$ workers, for some values of k. We conjecture that the above result is an artifact of having a small number of texts since the effect of a single text on the historical rank would likely be diluted if there was a larger number of texts.

To summarize, our results in this section imply that combining outputs from any number of the overall top 40% workers yields substantial improvements in expected accuracy in comparison to a combination of the same number of workers from the full population of workers, whereas removing workers amongst the overall worst 40% workers might result in less accurate aggregate outputs.

6 Consistency of Workers Across Multiple Tasks

Our previous analysis shows that the relative performance of some workers is not necessarily consistent across multiple tasks. In order to further investigate this issue, we first calculated the *overall ranking* of workers in terms of overall errors, *i.e.*, we sorted workers in ascending order according to their overall errors.

Next, we calculated the *individual rankings* of each worker in terms of individual errors, *i.e.*, for each reported output, we sorted workers in ascending order according to their errors. Thus, each worker was ranked three times according to his errors. Ties in rankings were allowed, *i.e.*, workers with similar (overall) errors received the same ranking.

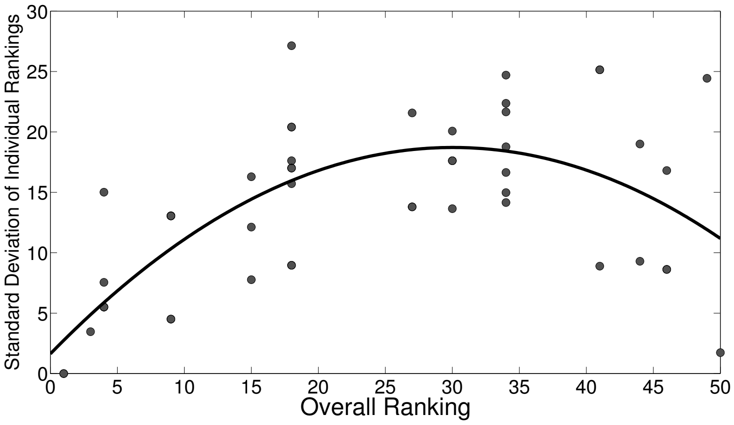
In the following analysis, we use the standard deviation of a worker's individual rankings as a measure of how stable the overall ranking of that worker is, where a high standard deviation indicates more ranking inconsistency across multiple tasks. For example, suppose that the outputs of a worker result in the lowest error in Text 1, the third lowest error in Text 2, and the second lowest error in Text 3. Thus, the standard deviation of that worker's individual rankings is equal to 1, showing high consistency across multiple tasks. On the other hand, a worker with individual rankings equal to 5, 48, and 22 is much more inconsistent across multiple tasks since the standard deviation of his individual rankings is 21.66.

Figure 4 shows the standard deviation of individual rankings as a function of the overall ranking of each worker. For the sake of a better visualization, we fit a quadratic function to the data in a least-squares sense (norm of residuals equal to 35.664). We note that 2 is the optimal degree for polynomial fitting according to the Akaike information criterion (AIC). The resulting quadratic function is:

$$f(x) = -0.018922 * x^2 + 1.1371 * x + 1.6287$$

where x is a worker's overall ranking, and $f(x)$ is the standard deviation of that worker's individual rankings. Figure 4 shows that the overall top-performing workers are more consistent across multiple tasks than the other workers. For example, the standard deviations of the individual rankings of the top 7 workers are always less than 15, whereas 4 out of the 7 worst-performing workers have

Fig. 4. The standard deviation of individual rankings as a function of the overall ranking of each worker.



standard deviations greater than 15. In general, the most inconsistent workers are the workers with overall ranking between 15 and 35.

The results presented in this section, together with the results from the previous section, suggest that removing workers with high overall error from the population of workers might be a mistake since those workers can sometimes produce high quality outputs, as can be inferred from Figure 4. Furthermore, restricting the population of workers to a few overall top-performing workers is likely to produce more accurate aggregate outputs because these workers consistently report outputs with low errors.

7 Conclusion

In this paper, we empirically studied the influence of the number of workers on the accuracy of aggregate outputs in a crowdsourcing setting. We first showed that adding more workers reduces the average error of the aggregate output, which was measured in terms of the root-mean-square deviation between the aggregate output and a gold-standard output. In other words, the expected accuracy of the aggregate output increases as the number of workers increases.

We also showed that there are diminishing returns for extra workers, where the reduction in the average error is always less than 2% after the sixth worker. Adding extra workers also implies that the risk of obtaining a combination of exclusively poor workers decreases because both the standard deviation of errors in aggregate outputs and the maximum error decrease as the number of workers increases.

We then moved to analyze the benefits of removing the overall worst-performing workers from the population of workers as well as the benefits of restricting the population of workers to only the overall top-performing workers. We found that an aggregate output from any combination of up to k top-performing workers, for $k \in \{1, \ldots, 20\}$, is, in expectation, more accurate than an aggregate output from a random combination with the same number of workers from the complete population of workers.

Unexpectedly, removing any number $k \in \{2, \ldots, 20\}$ of worst-performing workers does not necessarily result in more accurate aggregate outputs. The reason for this unexpected result is that the worst-performing workers are not always consistent across multiple tasks, which implies that a poor worker can eventually produce an accurate output.

Based on our results, our first recommendation for an organization or a decision maker who wants to design a crowdsourcing process is: in the absence of prior knowledge about the accuracy of the workers, having more workers is always beneficial because both the expected error in the aggregate output and the risk of obtaining a poor combination of workers decrease as the number of workers increases. Clearly, the marginal costs as well as the marginal benefits of adding extra workers must be considered in practice. Our results showed that most of the benefit occurs with the first five to six workers. Thereafter, the marginal benefit of adding another worker is very low, and it might not outweigh the cost of adding the extra worker.

Our second recommendation for a more efficient design of crowdsourcing processes concerns the case when there exists prior knowledge about the accuracy of the workers. In this case, one should focus only on combinations of the overall top-performing workers since this greatly reduces the expected error in the aggregate output. We found that almost perfect accuracy can be achieved by using only combinations of the very top workers. In practice, however, workers have constraints on the number of outputs they are willing to provide. This issue can be addressed by increasing the pool of top-performing workers. Our results show that when the size of the pool is up to 40% of the size of the full population, the aggregate outputs from the top-performing workers are, in expectation, still more accurate than the aggregate outputs from the full population of workers, for the same number of workers.

It is noteworthy that our study focused on simple averages to combine workers' outputs. More sophisticated combination procedures are available (*e.g.*, see the work by Carvalho and Larson [2]), but simple averages have been shown to perform well empirically and to be robust when eliciting expert opinions in different domains [3]. In addition, an averaging approach is easy to use, requiring neither assessments regarding the worker's judgment process nor self-assessed confidence in the accuracy of the reported outputs.

An exciting open question is whether or not the results obtained in our study hold true in different settings, *e.g.*, for different tasks, number of answers, *etc.* We argue that an answer to this question is of great importance to the crowdsourcing community given its potential to create less costly and more effective crowdsourcing processes.

Moreover, it would be of theoretical value to make stronger connections between our results in this paper and results from statistical theory. For example, an interpretation of our results in Section 4 is that we are estimating the population average error through empirical distributions of average errors, one for each group size $r \in \{1, \dots, n\}$. Under this interpretation, the Central Limit Theorem then implies the reduction of the variance (risk) observed in our results. Exploring this connection might be useful to determine the optimal number of workers to hire, but now taking the risk of poor combinations of workers into account.

Acknowledgments. The authors acknowledge Craig Boutilier, Pascal Poupart, Daniel Lizotte, and Xi Alice Gao for useful discussions. The authors thank Carol Acton, Katherine Acheson, Stefan Rehm, Susan Gow, and Veronica Austen for providing gold-standard outputs for our experiment. The authors also thank the Natural Sciences and Engineering Research Council of Canada for funding this research.

References

1. Carvalho, A., Dimitrov, S., Larson, K.: The Output-Agreement Method Induces Honest Behavior in the Presence of Social Projection. ACM SIGecom Exchanges 13(1), 77–81 (2014)
2. Carvalho, A., Larson, K.: A Consensual Linear Opinion Pool. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence. pp. 2518–2524. AAAI Press (2013)
3. Clemen, R.T.: Combining Forecasts: A Review and Annotated Bibliography. International Journal of Forecasting 5(4), 559–583 (1989)
4. Gao, X.A., Mao, A., Chen, Y.: Trick or Treat: Putting Peer Prediction to the Test. In: Proceedings of the 1st Workshop on Crowdsourcing and Online Behavioral Experiments (2013)
5. Ho, C.J., Vaughan, J.W.: Online Task Assignment in Crowdsourcing Markets. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence. pp. 45–51 (2012)
6. Ipeirotis, P.G.: Analyzing the Amazon Mechanical Turk Marketplace. XRDS Crossroads: The ACM Magazine for Students 17(2), 16–21 (2010)
7. Lin, C.H., Weld, D.S.: Dynamically Switching Between Synergistic Workflows for Crowdsourcing. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence. pp. 132–133 (2012)
8. Mason, W., Suri, S.: Conducting Behavioral Research on Amazon's Mechanical Turk. Behavior Research Methods 44(1), 1–23 (2012)
9. Neruda, P.: 100 Love Sonnets. Exile (2007)
10. Quinn, A.J., Bederson, B.B.: Human Computation: a Survey and Taxonomy of a Growing Field. In: Proceedings of the 2011 SIGCHI Conference on Human Factors in Computing Systems. pp. 1403–1412 (2011)
11. Sheng, V.S., Provost, F., Ipeirotis, P.G.: Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers. In: Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining. pp. 614–622 (2008)
12. Taylor, J., Taylor, A., Greenaway, K.: Little Ann and Other Poems. Nabu Press (2010)

13. Tran-Thanh, L., Stein, S., Rogers, A., Jennings, N.R.: Efficient Crowdsourcing of Unknown Experts Using Multi-Armed Bandits. In: Proceedings of the 20th European Conference on Artificial Intelligence. pp. 768–773 (2012)
14. Yuen, M.C., King, I., Leung, K.S.: A Survey of Crowdsourcing Systems. In: Proceedings of IEEE 3rd International Conference on Social Computing. pp. 766–773 (2011)
15. Zhang, H., Horvitz, E., Parkes, D.: Automated Workflow Synthesis. In: Proceedings of the 27th AAAI Conference on Artificial Intelligence. pp. 1020–1026 (2013)

Appendix A - Description of the Texts

We describe in this appendix the texts used in our experiments as well as the gold-standard scores.

Text 1

An excerpt from the "Sonnet XVII" by Neruda [9]. Intentionally misspelled words are highlighted in bold.

> "I do not love you as if you **was** salt-rose, or topaz
> or the **arrown** of carnations that spread fire:
> I love you as certain dark things are loved,
> secretly, between the **shadown** and the soul"

The gold-standard scores for the criteria *grammar*, *clarity*, and *relevance* are, respectively, 1, 2, and 2.

Text 2

An excerpt from "The Cow" by Taylor *et al.* [12]. Intentionally misspelled words are highlighted in bold.

> "THANK you, **prety** cow, that made
> **Plesant** milk to soak my bread,
> Every day and every night,
> Warm, and fresh, and sweet, and white."

The gold-standard scores for the criteria *grammar*, *clarity*, and *relevance* are, respectively, 1, 2, and 1.

Text 3

Words randomly generated in a semi-structured way. Each line starts with a noun followed by a verb in a wrong verb form. In order to mimic a poetic writing style, all the words in the same line start with a similar letter.

“Baby bet binary boundaries bubbles
Carlos cease CIA conditionally curve
Daniel deny disease domino dumb
Faust fest fierce forced furbished”

The gold-standard scores for the criteria *grammar*, *clarity*, and *relevance* are, respectively, 0, 0, and 0.

Appendix B - Experimental Results

Table 1 shows the numerical results from all the analysis performed in this paper.

Table 1: The average error, the standard deviation of the errors, and the maximum error per text for different populations of agents. All the values are rounded to 3 decimal places.

		Text 1			Text 2			Text 3		
Population	r	Avg	Std	Max	Avg	Std	Max	Avg	Std	Max
All	1	0.685	0.492	1.732	0.679	0.327	1.414	0.675	0.526	1.633
	2	0.621	0.322	1.732	0.567	0.220	1.323	0.650	0.348	1.633
	3	0.591	0.260	1.610	0.520	0.184	1.305	0.637	0.278	1.515
	4	0.575	0.225	1.555	0.495	0.162	1.199	0.630	0.237	1.451
	5	0.565	0.201	1.414	0.480	0.146	1.114	0.625	0.210	1.352
	6	0.560	0.183	1.398	0.469	0.135	1.045	0.622	0.189	1.305
	7	0.554	0.167	1.314	0.462	0.124	0.962	0.620	0.173	1.259
	8	0.550	0.155	1.220	0.457	0.116	0.944	0.618	0.161	1.242
	9	0.548	0.145	1.185	0.452	0.109	0.914	0.616	0.150	1.232
	10	0.545	0.135	1.158	0.449	0.102	0.920	0.616	0.140	1.134
	11	0.544	0.128	1.102	0.446	0.096	0.827	0.615	0.132	1.127
	12	0.542	0.121	1.063	0.444	0.092	0.814	0.614	0.124	1.120
	13	0.541	0.115	1.048	0.442	0.087	0.802	0.613	0.118	1.078
	14	0.540	0.110	1.067	0.440	0.083	0.768	0.613	0.111	1.042
	15	0.539	0.105	0.996	0.439	0.079	0.768	0.612	0.106	1.034
	16	0.538	0.100	1.026	0.438	0.076	0.744	0.611	0.102	1.032
	17	0.537	0.096	0.973	0.436	0.073	0.724	0.611	0.097	1.000
	18	0.537	0.092	0.911	0.436	0.069	0.726	0.610	0.093	1.012
	19	0.536	0.088	0.932	0.434	0.067	0.710	0.611	0.089	0.971
	20	0.536	0.084	0.877	0.434	0.064	0.694	0.610	0.086	0.979
	21	0.535	0.081	0.922	0.433	0.062	0.681	0.610	0.082	1.027
	22	0.535	0.078	0.860	0.433	0.059	0.660	0.610	0.079	0.923
	23	0.535	0.074	0.839	0.432	0.057	0.657	0.610	0.075	0.946
	24	0.534	0.072	0.835	0.432	0.055	0.650	0.609	0.073	0.950
	25	0.534	0.069	0.804	0.431	0.053	0.640	0.609	0.070	0.967

Continued on the next page

		Text 1			Text 2			Text 3		
Population	r	Avg	Std	Max	Avg	Std	Max	Avg	Std	Max
All	26	0.534	0.066	0.824	0.431	0.051	0.629	0.609	0.067	0.890
	27	0.534	0.064	0.793	0.430	0.048	0.638	0.609	0.065	0.878
	28	0.534	0.061	0.791	0.430	0.047	0.627	0.609	0.062	0.850
	29	0.533	0.059	0.758	0.429	0.045	0.606	0.609	0.059	0.847
	30	0.533	0.056	0.752	0.429	0.043	0.600	0.609	0.057	0.852
	31	0.533	0.054	0.762	0.429	0.041	0.594	0.608	0.055	0.847
	32	0.533	0.052	0.738	0.428	0.040	0.578	0.609	0.052	0.823
	33	0.532	0.049	0.726	0.428	0.038	0.571	0.608	0.050	0.809
	34	0.532	0.047	0.718	0.428	0.036	0.567	0.608	0.048	0.837
	35	0.532	0.045	0.709	0.428	0.035	0.575	0.608	0.046	0.813
	36	0.532	0.043	0.718	0.428	0.033	0.556	0.608	0.044	0.791
	37	0.532	0.041	0.692	0.427	0.031	0.542	0.608	0.041	0.785
	38	0.532	0.039	0.672	0.427	0.030	0.548	0.608	0.039	0.767
	39	0.531	0.036	0.677	0.427	0.028	0.537	0.608	0.037	0.757
	40	0.531	0.034	0.660	0.426	0.027	0.542	0.608	0.035	0.754
	41	0.531	0.032	0.647	0.426	0.025	0.518	0.608	0.033	0.727
	42	0.531	0.030	0.638	0.426	0.023	0.510	0.608	0.030	0.720
	43	0.531	0.028	0.623	0.426	0.021	0.513	0.607	0.028	0.706
	44	0.531	0.026	0.612	0.426	0.020	0.502	0.607	0.026	0.690
	45	0.531	0.023	0.598	0.426	0.018	0.491	0.607	0.023	0.675
	46	0.531	0.020	0.585	0.426	0.016	0.481	0.607	0.021	0.660
	47	0.531	0.017	0.570	0.425	0.013	0.470	0.607	0.018	0.646
	48	0.531	0.014	0.556	0.425	0.011	0.459	0.607	0.014	0.632
	49	0.530	0.010	0.543	0.425	0.008	0.442	0.607	0.010	0.620
	50	0.530	0.000	0.530	0.425	0.000	0.425	0.607	0.000	0.607
Top 47	1	0.632	0.456	1.732	0.684	0.318	1.414	0.641	0.525	1.633
	2	0.572	0.298	1.555	0.581	0.219	1.323	0.617	0.346	1.633
	3	0.543	0.241	1.503	0.541	0.182	1.305	0.604	0.275	1.515
Top 3	1	0.000	0.000	0.000	0.192	0.333	0.577	0.000	0.000	0.000
	2	0.000	0.000	0.000	0.192	0.167	0.289	0.000	0.000	0.000
	3	0.000	0.000	0.000	0.192	0.000	0.192	0.000	0.000	0.000