Ann Math Artif Intell (2016) 78:45–72
DOI 10.1007/s10472-015-9492-4

How many crowdsourced workers should a requester hire?

Arthur Carvalho[1] · Stanko Dimitrov[2] · Kate Larson[3]

Published online: 6 January 2016

Abstract Recent years have seen an increased interest in crowdsourcing as a way of obtaining information from a potentially large group of workers at a reduced cost. The crowdsourcing process, as we consider in this paper, is as follows: a requester hires a number of workers to work on a set of similar tasks. After completing the tasks, each worker reports back outputs. The requester then aggregates the reported outputs to obtain aggregate outputs. A crucial question that arises during this process is: how many crowd workers should a requester hire? In this paper, we investigate from an empirical perspective the optimal number of workers a requester should hire when crowdsourcing tasks, with a particular focus on the crowdsourcing platform Amazon Mechanical Turk. Specifically, we report the results of three studies involving different tasks and payment schemes. We find that both the expected error in the aggregate outputs as well as the risk of a poor combination of workers decrease as the number of workers increases. Surprisingly, we find that the optimal number of workers a requester should hire for each task is around 10 to 11, no matter the underlying task and payment scheme. To derive such a result, we employ a principled analysis based on bootstrapping and segmented linear regression. Besides the above result, we also find that

✉ Arthur Carvalho
carvalho@rsm.nl

Stanko Dimitrov
sdimitro@uwaterloo.ca

Kate Larson
kate.larson@uwaterloo.ca

[1] Rotterdam School of Management, Erasmus University, Burgemester Oudlaan 50, 3062 PA Rotterdam, The Netherlands

[2] Department of Management Sciences, University of Waterloo, 200 Universtiy Ave W., Waterloo, ON N2L 3G1, Canada

[3] David R. Cheriton School of Computer Science, University of Waterloo, 200 Universtiy Ave W., Waterloo, ON N2L 3G1, Canada

overall top-performing workers are more consistent across multiple tasks than other workers. Our results thus contribute to a better understanding of, and provide new insights into, how to design more effective crowdsourcing processes.

Keywords Crowdsourcing · Human computation · Amazon mechanical turk

Mathematics Subject Classification (2010) 68T99 · 90B99

1 Introduction

Recent technological advances have facilitated the outsourcing of a variety of tasks to "the crowd", e.g., the decision support regarding various phases of managerial decision-making and problem solving [15], the design of advertisements [33], 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, or *outsourcing* tasks to the *crowd*, is traditionally referred to as *crowdsourcing*.

There are many different ways of outsourcing a task to the crowd. The crowdsourcing process we consider in this paper is as follows: a *requester* hires a number of *crowd workers* to work on a set of similar *tasks*. The term *requester* denotes an agent who wants to get the task solved, e.g., an institution, a researcher, etc. The underlying tasks are homogeneous in a sense that they are instances of the same class of tasks, e.g., content-analysis tasks, prediction tasks, and so on. Workers then work on the same set of tasks, but without formally communicating to each other. This is done to preserve the diversity of opinions throughout the process.

After completing the tasks, each worker reports an *output* per task back to the requester. Outputs are context-dependent. For example, for prediction tasks, each output can be either a point estimate or a probability distribution over the plausible outcomes, whereas in sentiment-analysis tasks, the output is usually a score inside a discrete set representing how positive/negative the sentiment behind the underlying text is. After obtaining workers' outputs, the requester then aggregates the reported outputs to obtain an *aggregate output* per task. We focus on averages when aggregating workers' outputs, a simple, yet robust technique [14, 16]. Ideally, aggregate outputs are, in expectation, more accurate than any individual output. This is the basic premise behind the so called *collective intelligence*.

A crucial question that arises during the above crowdsourcing process is: how many crowd workers should a requester hire? Or, less specifically, how does the number of workers influence the quality of the aggregate output? We first note that arguments can be made in favor and against the use of multiple workers. On the one hand, hiring multiple workers might bring diversity to the crowdsourcing process so that biases of individual judgments can offset each other, which might result in a more accurate aggregate output. On the other hand, a larger population of crowd workers might bring down the quality of aggregate outputs due to the likely inclusion of poor-quality workers.

In this paper, we empirically investigate the above questions through a series of studies using a popular crowdsourcing platform: *Amazon Mechanical Turk*. Our studies differ from each other in terms of the underlying tasks and/or payment schemes. In our first study, we ask workers to solve three content-analysis tasks, and we pay workers per completed task.

In our second study, we also ask workers to solve three content-analysis tasks, but their payments are based on the similarity of their reported outputs. In our third study, we ask workers to solve two prediction tasks, and we pay the workers using a proper scoring rule [42].

Due to the nature of the tasks in our studies, we are able to derive *gold-standard outputs* for each task, i.e., either ground-truth outputs or 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. In our first analysis, we find a substantial degree of improvement in expected accuracy as we increase the number of hired 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 workers' outputs.

Our next contribution is a principled method for determining the optimal number of workers a requester should hire. Specifically, the proposed method combines bootstrapping with segmented linear regression analysis to determine the point at which hiring an extra worker has a negligible impact on the expected accuracy of the aggregate output. Surprisingly, we find in our studies that the optimal number of workers a requester should hire for each task is around 10 to 11.

Our experimental results also show that, given a set of similar tasks, combining outputs only from the overall top-performing workers results in more accurate aggregate outputs than combining outputs from the full population of workers. Furthermore, the performance of top-performing workers across multiple tasks is more consistent than the performance of other workers. Finally, more elaborate payment schemes, such as the output-agreement method and proper scoring rules, increase the consistency of the workers' performance across multiple tasks. We conjecture that this result happens due to the fact that the aforementioned payment schemes might induce honest reporting of private information.

Besides this introductory section, the rest of this paper is organized as follows. We review the literature related to our work in Section 2. In Section 3, we describe Amazon Mechanical Turk, the crowdsourcing platform we use in our studies. We describe our first study in Section 4, together with our segmented linear regression analysis, which is also subsequently applied to the data sets from our second and third studies in Sections 5 and 6. Finally, we conclude in Section 7, where we suggest how a requester can take advantage of our findings to design an effective crowdsourcing process. We also discuss some limitations of our work, and suggest directions for future research.

2 Related work

In recent years, the crowdsourcing research community has tackled many problems of different nature, e.g., how to assign tasks to workers [21, 40], how to design optimal workflows to coordinate the work of the crowd [25, 45], how to induce honest behavior in crowdsourcing settings [8, 18], etc. We refer the interested readers to the papers by Yuen et al. [43] and Quinn and Bederson [32] for comprehensive surveys on crowdsourcing-related works.

Our paper tackles the crowdsourcing problem of how many crowd workers a requester should hire and, as a consequence, the problem of how the number of workers influences the quality of the aggregate output. Regarding the latter question, it is well-known in decision analysis and operations research that combining information, such as forecasts, from multiple experts often leads to improved (forecasting) performance [14, 16, 41]. A canonical condition for this result to hold true is that experts' errors are uncorrelated. In practice, however, it is often the case that experts' errors are highly correlated [2]. We further argue that results and techniques that rely on calculations of correlations between experts' errors are not suitable for crowdsourcing settings since, due to the sheer number of workers, it is unlikely that many workers repeatedly complete the same tasks. Our work suggests an alternative way of analyzing the influence of the number of workers on the aggregate output which is practical and suitable for crowdsourcing settings.

Sheng et al. [37] and Ipeirotis et al. [24] tackled a problem closely related to the problems we tackle in this paper. In particular, those authors investigated how many labels a requester should obtain from crowd workers when (re)labeling a data set, which would later be used for supervised learning. Generally speaking, those authors showed that labeling the same data set using labels from different workers might sometimes improve the predictive performance of a model. Besides the empirical nature of our studies, our work is different from the aforementioned works in that we do not focus only on labeling tasks. Furthermore, instead of specific labeling strategies, we propose a general technique to determine the number of workers a requester should hire.

To the best of our knowledge, the work by Carvalho et al. [9] was the first to propose a method to determine the optimal number of workers a requester should hire. Specifically, those authors suggested that such an optimal number occurs when hiring an extra worker decreases the marginal expected error in the aggregate output by less than 2 %. In our paper, we suggest a more principled approach based on bootstrapping and segmented linear regression analysis that does not rely on similar thresholds. We also demonstrate the robustness of the proposed method in different empirical studies.

It is worth mentioning the connection between our work and machine learning, in particular the ensemble learning literature [17]. Ensemble methods are machine learning algorithms that combine classifications/predictions made by individual classifiers/predictors when classifying/predicting new data points. An ensemble method is more accurate than any of its individual members when the individual classifiers/predictors are accurate and diverse [19], where accuracy means that the error rate is lower than random guessing, and diversity means that the errors made by the classifiers/predictors are uncorrelated. To a certain degree, one can think of a crowd worker in our setting as a classifier/predictor. Then, our research question becomes: how many learning algorithms should an ensemble method use? This was the question investigated by Oshiro et al. [29]. Specifically, the authors studied whether there is a point where increasing the number of trees inside a random forest brings no significant performance gain. The authors found that adding extra trees beyond a certain threshold might actually decrease the performance of a random forest. Our results differ in that the expected accuracy of an aggregate output increases with the number of workers.

3 Amazon mechanical turk

Over the years, Amazon Mechanical Turk (AMT)[1] has emerged as the *de facto* crowdsourcing platform. One of the reasons behind the popularity of AMT is that it has consistently attracted thousands of workers, the so called *MTurkers*, willing to complete hundreds of

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

thousands of outsourced tasks for relatively low pay. Most tasks posted on AMT, traditionally called *human intelligence tasks* (HIT), are tasks that are relatively easy for human beings, but nonetheless challenging or even currently impossible for computers, e.g., content analysis, audio transcription, filtering adult content, extracting data from images, etc. Some studies have shown that AMT can effectively collect valid data in those settings [26, 38].

Besides business-related and data collection/cleaning applications, AMT has also been widely used as a platform for conducting behavioral experiments. According to Mason and Suri [27], 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-quality and poor-quality 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.

Paolacci et al. [30] also suggested some advantages of using Amazon Mechanical Turk for conducting experiments. For example, tasks are completed at a very fast rate since several crowd workers might work simultaneously on a task. Moreover, a requester can handpick workers by asking pre-screening questions as well as by defining certain criteria that workers have to fulfill, such as location. Finally, due to workers' unique IDs, a requester is able to contact previously employed MTurkers and, thus, conduct longitudinal experiments.

There are some discussions on whether the outputs from MTurkers are of acceptable quality. Paolacci et al. [30] critically reviewed AMT by comparing this crowdsourcing platform to other types of recruiting and data collection sources, such as studies performed in laboratories, traditional web studies, and web studies with purpose-built websites. Paolacci et al. [30] concluded, among other things, that AMT offers lower risk in terms of susceptibility to coverage error and contaminated subject pool than the alternative approaches.

Buhrmester et al. [5] also conducted some research on the quality of MTurkers. First, the authors concluded that workers on Amazon Mechanical Turk are more diverse and representative of the general population than subjects from some other internet samples and typical American college samples. Furthermore, Buhrmester et al. [5] found that the quality of the data generated by MTurkers is at least as high as the psychometric standard which is associated with published research. Interestingly, those authors held a survey on AMT which took approximately 30 minutes to complete, and the compensation per completion was only \$0.02. Even with this rather low payment of two American cents per 30 minutes, Buhrmester et al. [5] collected data from 25 crowd workers in about five hours. After some experiments, the authors concluded that the level of compensation does not seem to influence the quality of the collected data. However, the length of the tasks and higher compensation rates are, respectively, inversely and directly proportional to how fast the data is collected.

4 Study 1: Content-analysis tasks with payments per task

Our first experiment designed to study the influence of the number of crowd workers on the quality of the aggregate outcome consists of a traditional setting on AMT, namely content-analysis tasks with payment per completed task. In what follows, we describe the experimental design, our analysis, and the obtained results.

4.1 Experimental design

We asked workers on AMT to review three short texts under three different criteria: *grammar*, *clarity*, and *relevance*. The first two texts are extracts from published poems, but with some original words intentionally replaced by misspelled words. The third text contains random words presented in a semi-structured way. Appendix A contains detailed information about the texts. For each text, we presented three questions to the workers, each one having three possible responses ordered in increasing positivity:

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

We intentionally used words with subjective meaning so as to emphasize the subjective nature of content analysis, e.g., “a lot”, “a few”, etc. In order to conduct a numerical analysis, we translate each individual response into a score inside the set {0, 1, 2}. In particular, we assign 0 to the most negative response, 1 to the middle response, and 2 to the most positive response. Thus, each worker reported a vector of 9 scores (3 criteria for each of the 3 texts). In this section, we denote by *output* a vector of 3 scores for a given text. Thus, each worker reported 3 outputs.

We recruited a total of 50 workers on AMT, all of them residing in the United States of America and older than 18 years old. We asked the workers to complete the three tasks in at most 20 minutes. After completing the tasks, every worker received a payment of $0.20. Ipeirotis [23] estimated 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 is much higher than the payment from the vast majority of other tasks posted on AMT.

Since we knew the source and original content of each text *a priori*, i.e., before conducting the content-analysis experiment, we were then able to derive *gold-standard outputs* for each task. In order to avoid confirmation bias,[2] we asked five professors and tutors from the

[2] The tendency to interpret information in a way that confirms one’s preconceptions [31].

English and Literature Department at the University of Waterloo to provide their outputs for each task. 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 reported scores. We show the gold-standard outputs in Appendix A.

4.2 Accuracy of aggregate outputs by the number of workers

In order to determine the optimal number of workers a requester should hire, we start by applying a bootstrapping technique to our collected data. In particular, for each number of workers $n \in \{1, \ldots, 50\}$ and each one of the three content-analysis tasks, we create 100,000 *bootstrap resamples* by randomly sampling with replacement n workers' outputs. For example, for $n = 5$, we randomly sample with replacement 5 outputs for a total of 100,000 times for each task.

For each bootstrap resample, we aggregate the n outputs by taking their average. We next measure the accuracy of each aggregate output. In particular, we calculate the *mean square error* (MSE) between each aggregate output and the respective gold-standard output. For example, consider the case where a bootstrap resample contains two workers' outputs for Text 1, namely (1, 2, 0) and (2, 2, 1). Thus, the aggregate output for that bootstrap resample is (1.5, 2, 0.5). Given that the gold-standard output for Text 1 is (1, 2, 2) (see Appendix A), the MSE between the aggregate output and the gold-standard output is:

$$\frac{(1.5-1)^2+(2-2)^2+(0.5-2)^2}{3} \approx 0.8334$$

Clearly, the lower the MSE, the more accurate the aggregate output. For a given number of workers n, the average MSE, henceforth called *average error*, can be seen as the *expected error* when aggregating outputs from n workers. For example, the average of the 100,000 MSEs for $n = 2$ is an estimate of the expected error when aggregating outputs from 2 workers chosen at random. Figure 1 shows the average error and the standard deviation of the errors for each content-analysis task and number of workers $n \in \{1, \ldots, 50\}$. Appendix B shows statistics regarding workers' errors.

An interesting feature of Fig. 1 is that the influence of the number of workers on the accuracy of the aggregate output is qualitatively the same for all tasks. That is, the average error decreases as the number of workers n increases, which means that the expected accuracy of the aggregate output increases with more workers. Figure 1 also shows that the standard deviation of the errors decreases with the number of workers n, which is just a consequence of the central limit theorem. The initially high standard deviation indicates an opportunity to get considerably low error with a single worker. Obviously, the other side of the coin is a greater risk of high error due to a single poor-quality worker. As the number of workers increases, that risk decreases because combinations of exclusively poor-quality workers become less likely.

Another interesting aspect of Fig. 1 is that the average error quickly converges to values other than zero. In particular, the 99 % confidence intervals for $n = 50$ and Task 1, 2, and 3, are, respectively, [0.2892, 0.2904], [0.1877, 0.1885], and [0.3754, 0.3768]. In practical terms, this result highlights a potential limitation with crowdsourcing, in that averaging outputs from many workers does not necessarily translate into a perfect, gold-standard output.

Looking at Fig. 1, a natural question that arises is: at which point does the average error become stable? By "stable", we mean that the average error no longer significantly changes

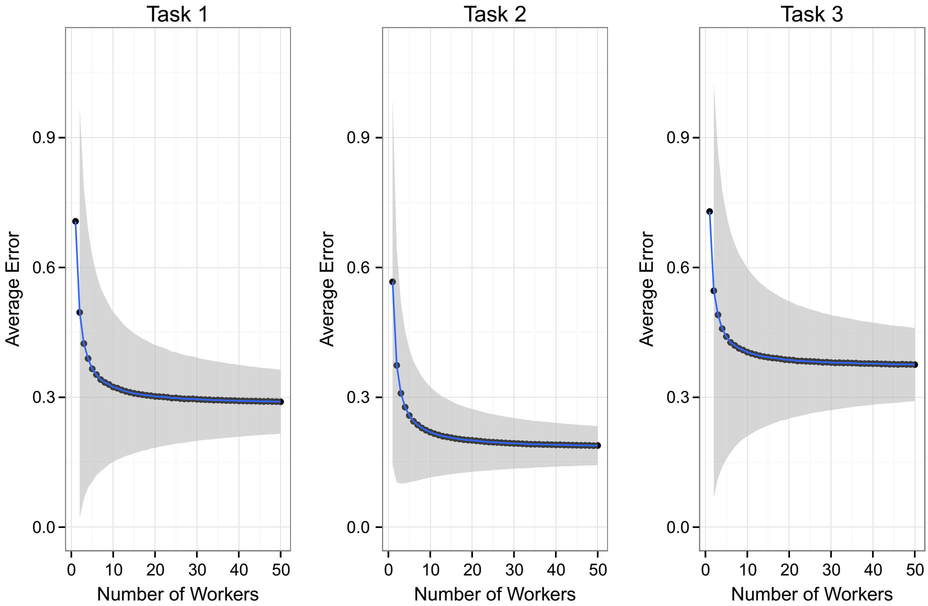


Fig. 1 The average error and the standard deviation of the errors per content-analysis task for each number of workers $n \in \{1, \ldots, 50\}$

with one extra worker. We note that such a point denotes the optimal number of workers a requester should hire given that hiring extra workers thereafter does not significantly affect the average error. A potential solution to the above question is to keep track of the marginal decrease in average error for different number of workers. Then, the optimal number of workers would be the point where the marginal decrease is below some predefined threshold. The problem with such a solution lies in how to define such a threshold.

To answer the above question in a more principled way, we perform a piecewise (segmented) linear regression analysis. The rationale behind this approach is to approximate the error curves in Fig. 1 with a series of line segments, where the last line segment shall capture the (almost) constant part of the error curve. Consequently, the breakpoint that separates the last two line segments is the starting point where the average error can be considered stable, i.e., it represents the optimal number of workers a requester should hire.

We now have to deal with the question on the optimal number of line segments (or breakpoints) in our model. We estimate the optimal number of breakpoints by using the dynamic programming algorithm for minimizing the segmented residual sum of squares suggested by Bai and Perron [4], and implemented by Zeileis et al. [44]. For all content-analysis tasks, we obtain that the optimal number of breakpoints is equal to 2. Moreover, the resulting breakpoints for all the three tasks occur at the x-values 3 and 11. Figure 2 shows the error curves in Fig. 1 approximated by the obtained 3 line segments.

The slopes of the line segments in Fig. 2, here referred to as β, allow us to derive intuitive interpretations of our results. First, we note that there is a considerable average error when aggregating outputs from 3 or less workers ($\beta < -0.11$). Second, the average error is moderate when aggregating outputs from 4 to 10 workers ($-0.009 < \beta < -0.007$). Finally,

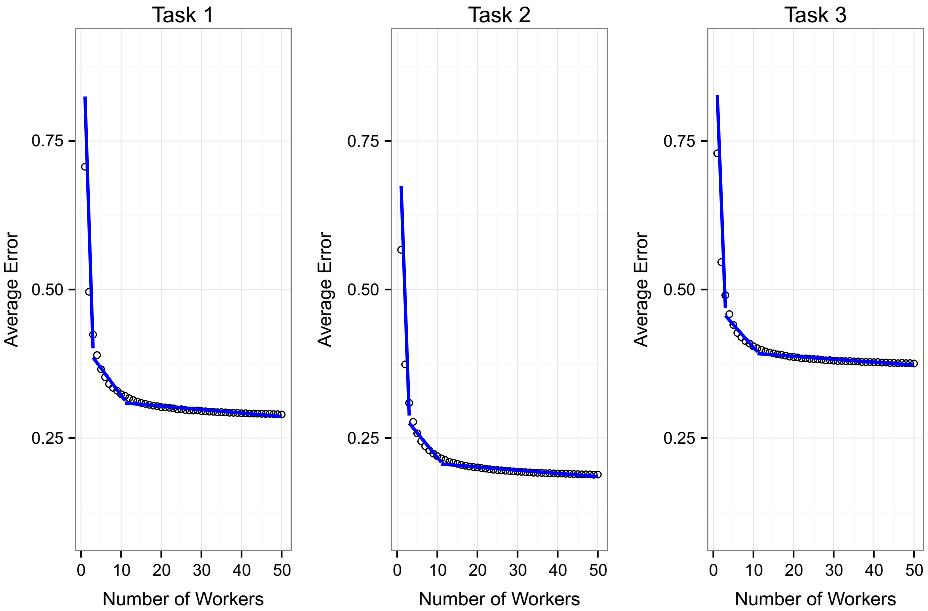


Fig. 2 Approximation of the error curves in Fig. 1 with 3 line segments

the average error becomes stable when aggregating outputs from 11 or more workers ($\beta \approx -0.0005$), and changes in the average error after the 11th worker are negligible. Thus, we conclude that 11 workers is the optimal number of workers to hire in our first study.

4.3 Accuracy of aggregate outputs from top-performing workers

Our previous analysis is based on combinations of workers from the full population of workers. An interesting follow-up question is: can the requester improve accuracy by restricting attention to combinations of outputs from the overall top-performing workers? In order to answer this question, workers must be somehow ranked based on their previous performance on content-analysis tasks. However, such information is not readily available on AMT.

We circumvent the above issue by sorting workers based on their *overall error* on the three content-analysis tasks. 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 MSE between his overall output and the overall gold-standard output. For example, suppose that a worker reports the following outputs for Task 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 Task 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 $x/9 \approx 0.2222$, where $x = (1-1)^2+(2-2)^2+(2-2)^2+(1-1)^2+(2-2)^2+(0-1)^2+(1-0)^2+(0-0)^2+(0-0)^2 = 2$.

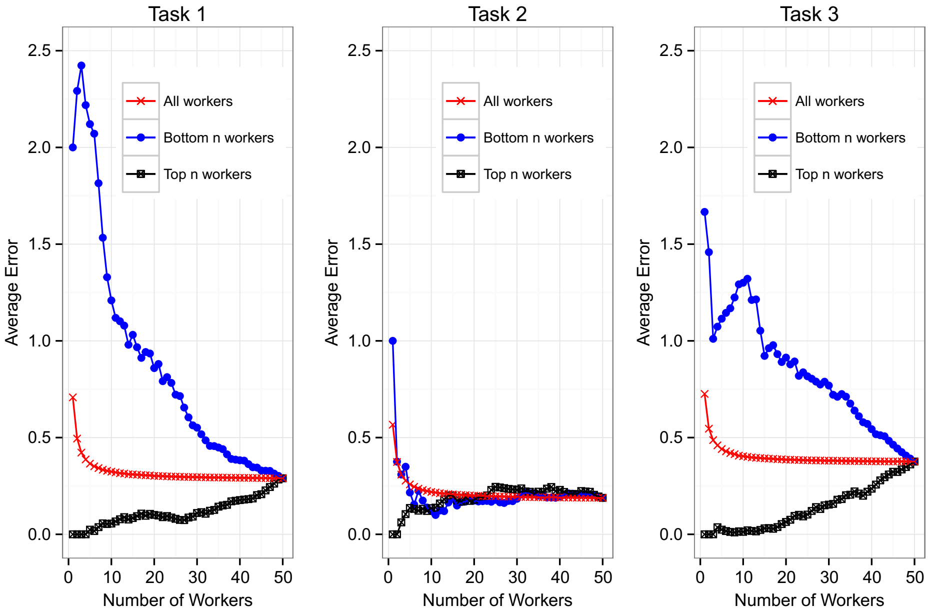


Fig. 3 The average error per task, number of workers, and different populations

After ordering workers in terms of overall errors, we create bootstrap resamples. In particular, for each one of the three content-analysis tasks and each number of workers $n \in \{1, \ldots, 50\}$, we randomly sample with replacement n workers' outputs from the full population of workers as well as the subpopulation defined by the top n workers. For example, for $n = 5$, we randomly sample with replacement 5 outputs for each task from the full population of workers as well as from the top 5 workers. We repeat this sampling procedure for a total of 100,000 times.

Similarly to the procedure described in the previous subsection, we aggregate the n outputs in each bootstrap resample by taking their average. We next measure the accuracy of each aggregate output by calculating the MSE between the aggregate output and the respective gold-standard output. Figure 3 shows the resulting average error per task, number of workers, and different populations.

Focusing first on Task 1 and 3, we note that any combination of the top 3 workers results in a perfect aggregate output with zero error. Generally speaking, the average error tends to increase with more overall poor-quality workers. The striking result comes from Task 2, where the average error for the full population of workers becomes *lower* than the average error for the top n workers after the 23^{rd} worker. As we elaborate in the following subsection, the reason for this counter-intuitive result is that there are workers among the overall worst-performing workers who excel in Task 2, while performing poorly in Task 1 and 3. The above results are statistically significant for any $n \in \{1, \ldots, 49\}$ (rank-sum test, p-value $< 10^{-15}$). As expected, we find no significant difference in the average errors for the top 50 and full population of workers ($n = 50$), since these two populations contain exactly the same number and the very same workers.

For completeness' sake, Fig. 3 also shows the other side of the coin, i.e., the average performance of the bottom n workers. Generally speaking, the average error decreases with the number of workers and, except for Task 2, the average error from the bottom n workers is never less than the average error from the top n workers as well as from n random workers from the full population of workers.

4.4 On the 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 calculate the *overall ranking* of workers in terms of overall errors, i.e., we sort workers in ascending order according to their overall errors. Next, we calculate the *individual rankings* of each worker in terms of individual errors, i.e., for each reported output, we sort workers in ascending order according to their errors. Thus, we rank each worker three times according to his individual errors.

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 for Task 1, the third lowest error for Task 2, and the second lowest error for Task 3. Then, the standard deviation of that worker's individual rankings is equal to 1, thus 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 workers' overall rankings. We also fit a regression line through the origin to the data ($R^2 = 0.73$, sum of the squared residuals = 2551), and display its 95 % confidence interval. The resulting linear function is:

$$f(x) = 0.403 * x \tag{1}$$

where x is a worker's overall ranking, and $f(x)$ is the estimate of the standard deviation of that worker's individual rankings. Figure 4 shows that the overall top-performing workers tend to be more consistent across multiple tasks than the other workers. For example, the standard deviations of the individual rankings of the top 5 workers are always less than 7, whereas 2 out of the 5 worst-performing workers have standard deviations greater than 7. In general, the most inconsistent workers are the ones with overall ranking between around 20 and 40. The coefficient of the regression line in (1) means that an increase by 1 in a worker's overall ranking implies an expected increase of 0.403 in the standard deviation of that worker's individual rankings (the 99 % confidence interval being $[0.3091, 0.4958]$), i.e., higher inconsistency across multiple tasks.

The results presented in this subsection together with the results from the previous subsection suggest that 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.

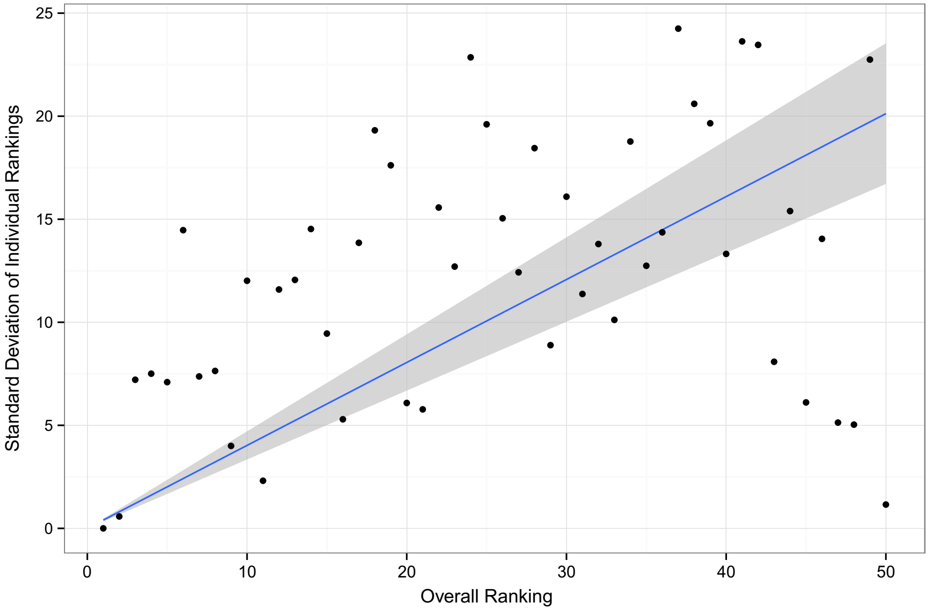


Fig. 4 The standard deviation of individual rankings as a function of workers' overall rankings

5 Study 2: Content-analysis tasks with payments based on number of agreements

Our second study to investigate the optimal number of crowd workers a requester should hire is a replication of our first study, but with one major change: the payment structure. In particular, we pay the workers based on how popular their reported outputs are. Our primary objective with this second study is to understand whether the underlying payment structure qualitatively changes the results from our first study. In what follows, we describe the experimental design, our analysis, and the obtained results regarding our second study.

5.1 Experimental design

The experimental design of our second study follows very close the experimental design of our first study. Specifically, we recruited a total of 50 workers on AMT, all of them living in the United States of America, to review the three texts described in Appendix A. As before, each worker reviewed each text under three different criteria: grammar, clarity, and relevance.

Besides recruiting different workers, we also used a different payment scheme to reward the workers. In particular, after completing the content-analysis tasks, each worker received a baseline payment of \$0.20. Moreover, each worker could earn an additional *bonus* of up to \$0.10. We informed the workers that their bonuses would be proportional to the number of answers similar to their reported answers. Recall that each worker reported 9 answers (3 texts times 3 criteria). For each answer reported by a worker i, we calculated the total number of agreements (#agreements) between worker i's reported answer and the answers reported by worker i's peers. Consequently, for each reported answer, there could be at most 49 similar reported answers because we recruited 50 workers. We then used the formula $\frac{10}{9} \times \frac{\#\text{agreements}}{49}$ to calculate the bonus for an individual reported answer.

Rewarding crowd workers based on pairwise comparisons between reported answers has been empirically proven to be an effective payment structure in different domains. For example, Huang and Fu [22] showed that informing the workers that their rewards will be based on how similar their answers are to other workers' answers results in more accurate answers than informing the workers that their rewards will be based on how similar their answers are to gold-standard answers. Moreover, Shaw et al. [36] found that payment schemes based on the similarity of reported answers result in more accurate answers.

The payment scheme we used to calculate workers' bonuses is often referred to as the *output-agreement method* [1]. Carvalho et al. [8] suggested a potential explanation for the effectiveness of the output-agreement method, namely that it induces honest reporting of private information under the assumption that the psychological phenomenon called social projection holds true. Specifically, when communication between workers is not allowed and a risk-neutral worker believes that his true answer to a multiple-choice question is the most popular answer amongst other workers, then the action that maximizes that worker's expected reward is to honestly report his true answer, as opposed to something else.

5.2 Accuracy of aggregate outputs by the number of workers

We perform the same bootstrapping technique and analysis described in Section 4.2 to investigate the optimal number of workers a requester should hire in our second study. Figure 5 shows the average error and the standard deviation of the errors for each content-analysis task and number of workers $n \in \{1, \ldots, 50\}$. Appendix B shows statistics regarding workers' errors. Similar to the results of our first study, the influence of the number of workers on the quality of the aggregate output is qualitatively the same for all texts, in a sense that the average error and the standard deviation of the errors decrease as the number of workers increases. Furthermore, the average error quickly converges to values other than zero, thus showing again that aggregating outputs from many workers does not necessarily translate into a perfect, gold-standard output. In particular, the 99 % confidence intervals for $n = 50$ and Task 1, 2, and 3, are, respectively, $[0.2696, 0.2706]$, $[0.1423, 0.1429]$, and $[0.2102, 0.2112]$.

We also find the optimal number of workers a requester should hire using the segmented linear regression analysis suggested in Section 4.2. Specifically, for all content-analysis tasks, we obtain that the optimal number of breakpoints is equal to 2. Moreover, the resulting breakpoints for all the three content-analysis tasks occur at the x-values 3 and 11. Thus, we conclude once again that 11 workers is the optimal number of workers a requester should hire. Figure 6 shows the error curves in Fig. 5 approximated by the obtained three line segments. The slopes of the first, second, and third line segments are inside the following ranges: $[-0.126, -0.118]$, $[-0.009, -0.007]$, and $[-0.0005, -0.0004]$.

An interesting point to note is that even though we obtain the same results regarding the optimal number of workers to hire in our first and second studies, the error curves for each task are quite different from each other. For example, for Tasks 1, 2, and 3, and $n = 50$,

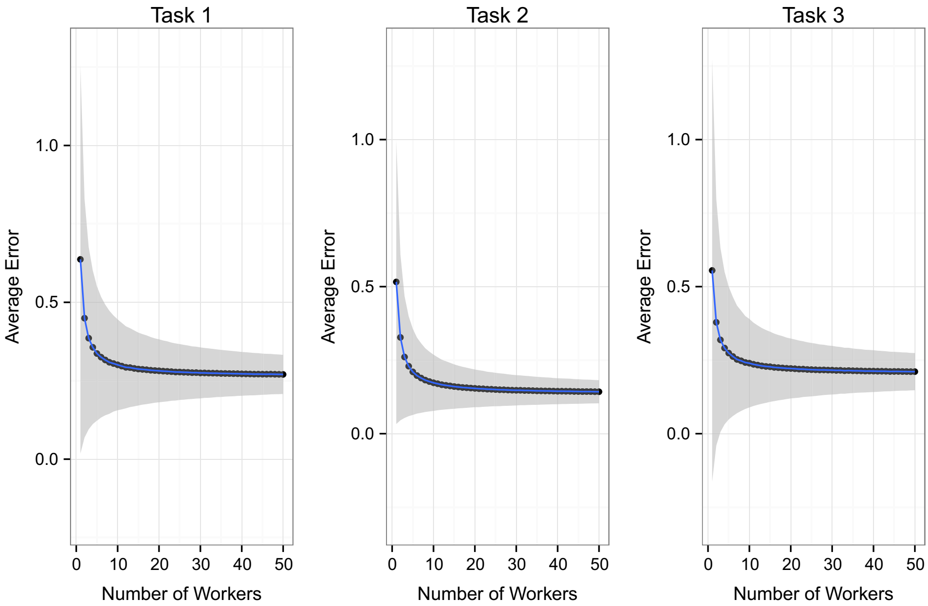


Fig. 5 The average error and the standard deviation of the errors per content-analysis task for each number of workers $n \in \{1, \ldots, 50\}$

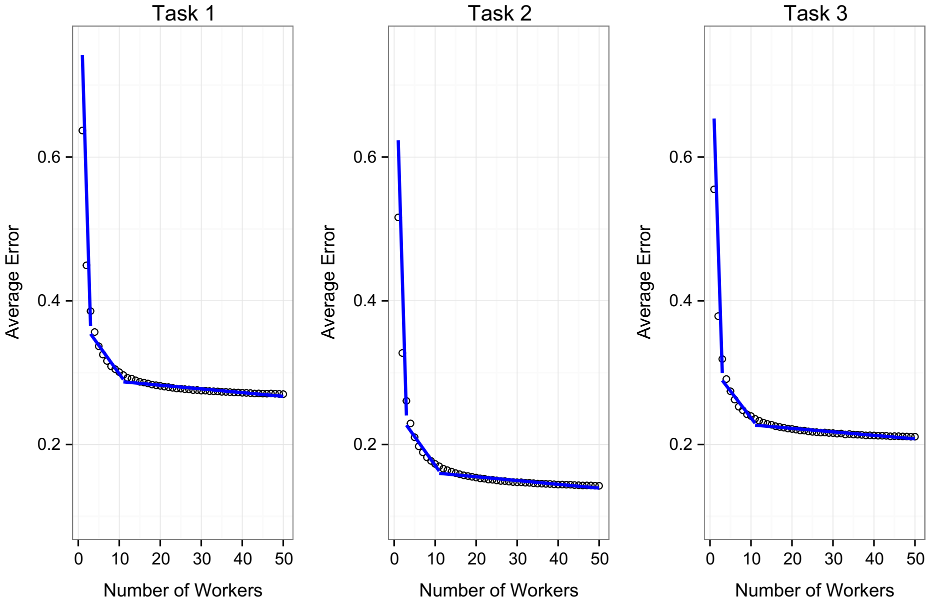


Fig. 6 Approximation of the error curves in Fig. 5 with 3 line segments.

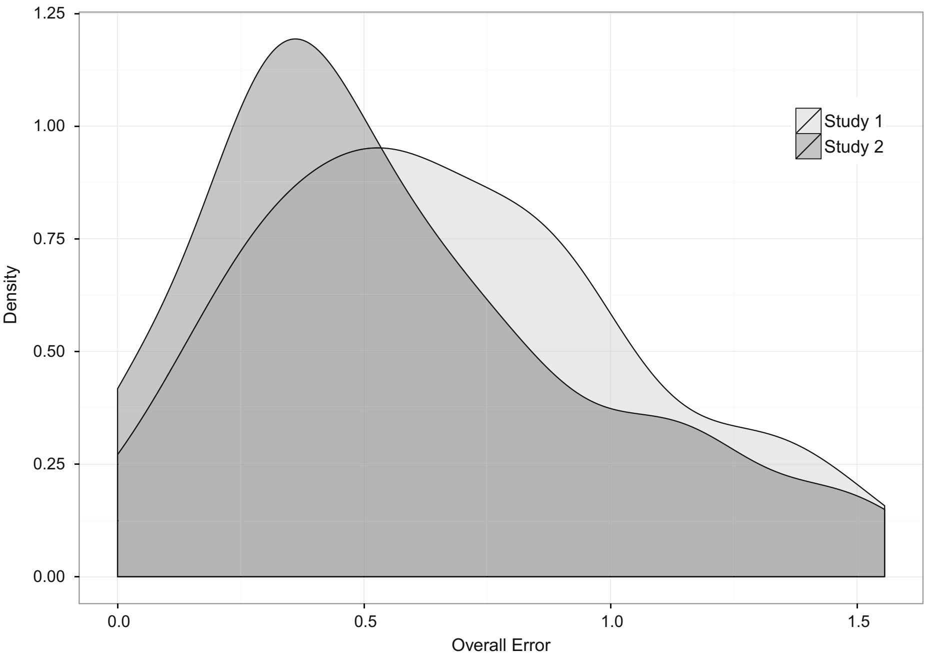


Fig. 7 Kernel density estimates of workers' overall errors for Study 1 and 2

the average errors in our first study (0.2898, 0.1881, and 0.3761) are always greater than the average errors in our second study (0.2701, 0.1426, and 0.2109). These results are all statistically significant (t-test, p-value $< 10^{-15}$), and allow us to conclude two things. First, using more elaborate payment structures, such as the output-agreement method, seems to result in more accurate aggregate outputs (at least in expectation). This result can also be seen in Fig. 7. Specifically, we calculate the overall error for each individual worker in Studies 1 and 2, and plot the kernel density estimates for both cases using the Gaussian kernel. One can see that the kernel density estimate for Study 2 can almost be obtained from the kernel density estimate for Study 1 by shifting probability mass towards 0, which means that workers in Study 2 are often more accurate than workers in Study 1. Second, our approach to find the optimal number of workers is robust in that different payment schemes do not seem to affect its results. That is, even though the workers in our second study are, on average, more accurate than the workers in our first study, our method suggests that, in both cases, the average error becomes nearly constant at around 11 workers.

5.3 Accuracy of aggregate outputs from top-performing workers

For our second study, we also investigate whether a requester can increase accuracy by combining only the outputs from the overall top-performing workers. Similar to Section 4.3, we start by ranking workers based on their overall error on the three content-analysis tasks. Thereafter, for each one of the three content-analysis tasks and each number of workers $n \in \{1, \ldots, 50\}$, we randomly sample with replacement n workers' outputs from the full

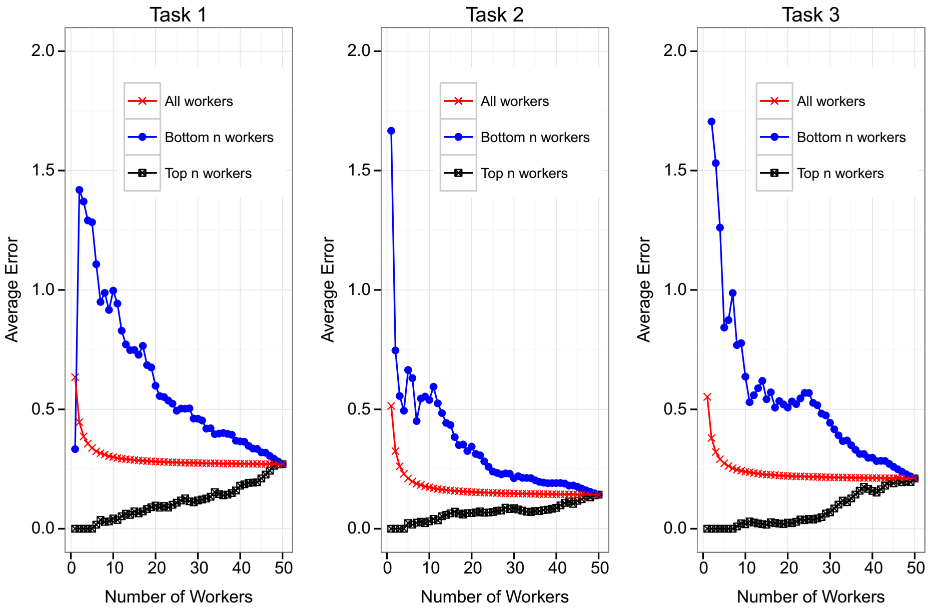


Fig. 8 The average error per task, number of workers, and different populations

population of workers as well as the subpopulation defined by the top n workers. In total, we create 100,000 bootstrap resamples per combination of task, number of workers, and population. Finally, we aggregate the outputs in each bootstrap resample by taking their average, and we measure the accuracy of each aggregate output by calculating the MSE between the aggregate output and the respective gold-standard output. Figure 8 shows the resulting average error per task, number of workers, and different populations.

We note that, for all tasks, any combination of the top 4 workers results in a perfect aggregate output. Moreover, the average error non-monotonically increases with more overall poor-quality workers. However, different than the results from our first study, the average error for the full population of workers is always greater than or equal to the average error for the top n workers. The above results are statistically significant for any $n \in \{1, \ldots, 49\}$ (rank-sum test, p-value $< 10^{-15}$).

A potential explanation for the above result is that more elaborate payment schemes, such as the output-agreement method, might induce less inconsistency across multiple tasks. In other words, top-ranked (low-ranked) workers are more often accurate (inaccurate) across multiple homogeneous tasks. We further investigate this point in the following subsection.

Looking at the symmetric case in Fig. 8, i.e., the bottom n workers, one can see that, except for $n = 1$ in Task 1, the average error from the bottom n workers is never less than the average error from the top n workers as well as from n random workers from the full population of workers. Recall that workers are sorted based on their overall errors. So, it might happen that one worker performs quite well in one task, but then performs very poorly in the remaining tasks, thus significantly lowering his rank. That is precisely what is happening in Fig. 8. Specifically, the overall worst-performing worker performed quite well in Task 1, but very poorly in Tasks 2 and 3. This explains why the average error of the

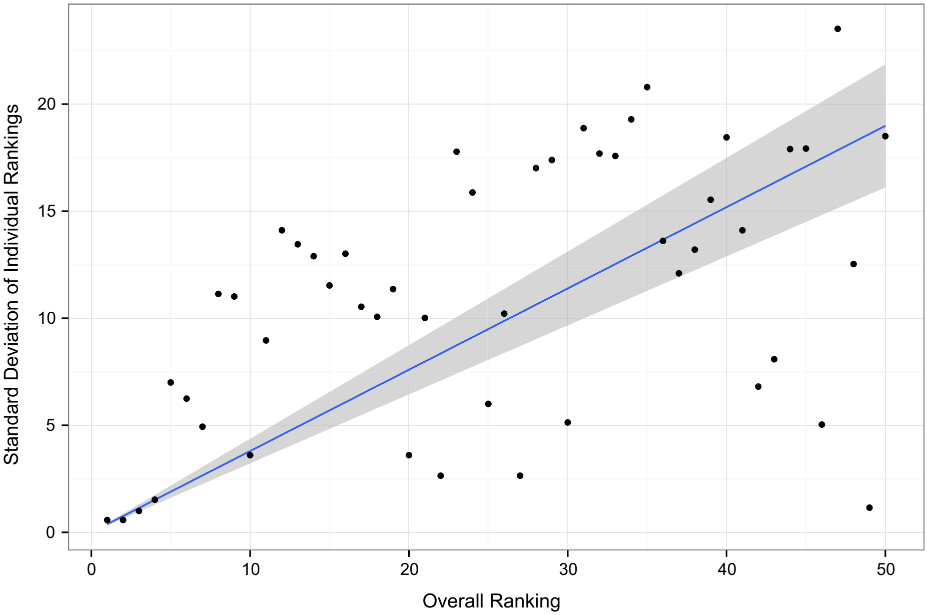


Fig. 9 The standard deviation of individual rankings as a function of workers' overall rankings

bottom 1 worker is considerably lower than that of the bottom 2 workers for Task 1, but the same is not true for Tasks 2 and 3.

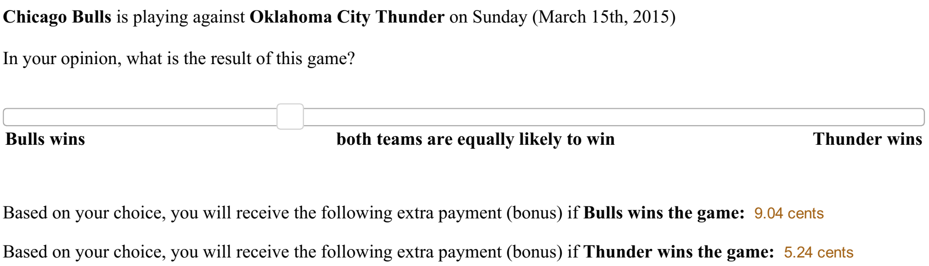
5.4 On the consistency of workers across multiple tasks

Figure 8 shows that the average error increases non-monotonically with the number of overall poor-quality workers. This lack of monotonicity might be taken as an indication that workers are, relatively speaking, still inconsistent across multiple tasks. Similar to Section 4.4, we further investigate this issue by analyzing the relationship between workers' overall rankings and the standard deviations of their individual rankings. Recall that the standard deviation of a worker's individual rankings is a measure of stability of that worker's overall ranking. Figure 9 shows the standard deviation of individual rankings as a function of workers' overall rankings. We also fit a regression line through the origin to the data ($R^2 = 0.77$, sum of the squared residuals = 1811), and display its 95 % confidence interval. The resulting linear function is:

$$f(x) = 0.38 * x \quad (2)$$

Figure 9 shows that the overall top-performing workers tend to be more consistent across multiple tasks than the other workers. The coefficient of the regression line in (2) means that an increase by 1 in a worker's overall ranking implies an expected increase of 0.38 in the standard deviation of that worker's individual rankings, the 99 % confidence interval being $[0.3011, 0.4584]$.

We note that the slope in (2) is lower than the respective coefficient in (1), which is equal to 0.403. Consequently, given that the expected increase in inconsistency is lower in



Chicago Bulls is playing against **Oklahoma City Thunder** on Sunday (March 15th, 2015)

In your opinion, what is the result of this game?

Based on your choice, you will receive the following extra payment (bonus) if **Bulls wins the game:** 9.04 cents

Based on your choice, you will receive the following extra payment (bonus) if **Thunder wins the game:** 5.24 cents

Fig. 10 Example of the graphical interface used in our third study

our second study than in our first study, though not necessarily statistically significantly so, we conjecture with a larger participant pool that the output-agreement method causes workers to be more consistent across multiple tasks. The reduction in the sum of the squared residuals (from 2551 to 1811) as well as the increase in the R^2 value (from 73 to 77) also show that a regression line with a positive slope better fits the data in our second study than in our first study, thus supporting the above conclusion.

6 Study 3: Prediction tasks with payments based on proper scoring rules

In our third study to determine the optimal number of crowd workers a requester should hire, we investigate the robustness of our previous findings. In particular, our experiments now involve prediction tasks, as opposed to content-analysis tasks, a reduced number of workers (40 instead of 50) and tasks (2 instead of 3), and payments based on proper scoring rules [42].

6.1 Experimental design

We asked 40 workers on AMT, all older than 18 years old and residing in the United States of America, to complete two *prediction tasks*. Specifically, workers had to predict the outcomes of two NBA games per task. The first task was about one home and one away game played by the Chicago Bulls team, whereas the second task was about one home and one away game played by the Los Angeles Lakers team. For each task, each worker's output was a vector containing two probability values, each one describing the likelihood that the home team would win a certain game. Consequently, each worker reported 4 probability values in our third study (2 games times 2 tasks).

Similar to our previous studies, we were able to derive gold-standard outputs for the prediction tasks. To do so, we observed the result of each predicted game: if the home team was the winner of the game, then the gold-standard probability was set to 1. Otherwise, the gold-standard probability was set to 0. Consequently, each gold-standard output was a vector containing either zeros or ones.

For each game, we informed the workers about the names of the teams playing the game and the date of the game. We also informed the workers that besides the baseline payment of \$0.10 for completing the task, each worker could earn an additional bonus of up to \$0.10 per game based solely on the accuracy of the reported prediction. To compute such a bonus, we used a *proper scoring rule* [42].

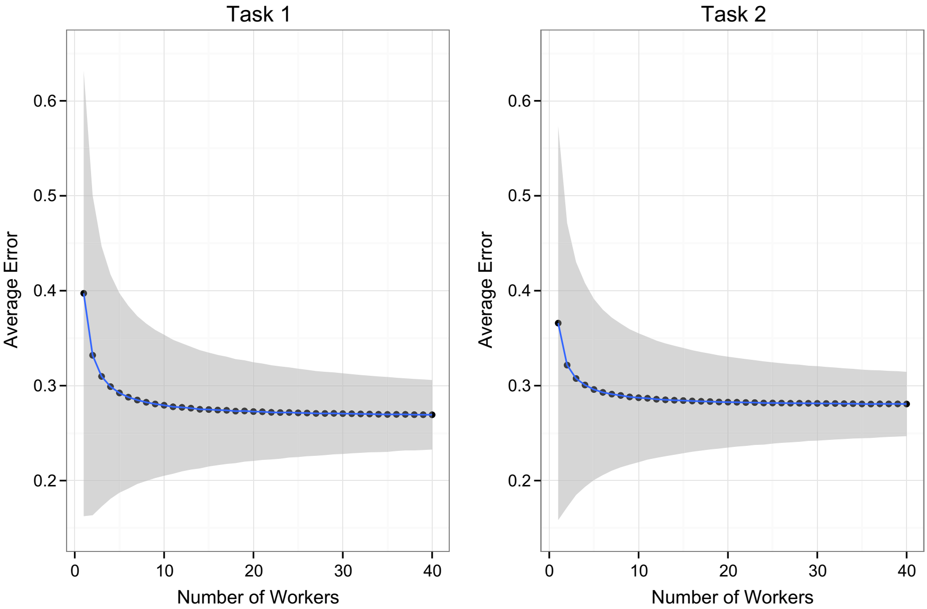


Fig. 11 The average error and the standard deviation of the errors per prediction task for each number of workers $n \in \{1, \ldots, 40\}$

Scoring rules are traditional techniques for measuring the accuracy of forecasts as well as to promote honesty in forecasting settings [34, 42]. Consider a set of exhaustive and mutually exclusive outcomes $\theta_1, \theta_2, \ldots, \theta_z$, for $z \geq 2$, and a prediction $\mathbf{q} = (q_1, q_2, \ldots, q_z)$, where q_x is the probability associated with the occurrence of outcome θ_x, for $x \in \{1, \ldots, z\}$. Formally, a scoring rule is a real-valued function, $R(\mathbf{q}, \theta_x)$, that provides a score for the prediction $\mathbf{q}$ upon observing the outcome θ_x, which in our setting is the winner of a game. Scoring rules measure the accuracy of predictions in a sense that the more probability mass is assigned to the observed outcome, the higher the resulting score. The condition that R is *proper* implies that the prediction reported by a worker maximizes his expected score when the worker is honest, i.e., when the worker reports his true prediction as opposed to something else [6, 34, 42]. Proper scoring rules have been used as a tool to promote honest reporting in a variety of domains, e.g., when sharing rewards based on peer evaluations [10–12], to incentivize agents to accurately estimate their own efforts to accomplish a task [3], in prediction markets [20], to incentivize honest reporting in the peer-review process [7], etc. In our experiments, we used the following positive affine transformation of the well-known proper scoring rule called *quadratic scoring rule*[3]:

$$R(\mathbf{q}, \theta_x) = 5 \times \left(2q_x - \sum_{k=1}^{n} q_k^2 \right) + 5$$

[3] The proof that the quadratic scoring rule is indeed proper as well as some of its interesting properties can be seen in the work by Selten [35].

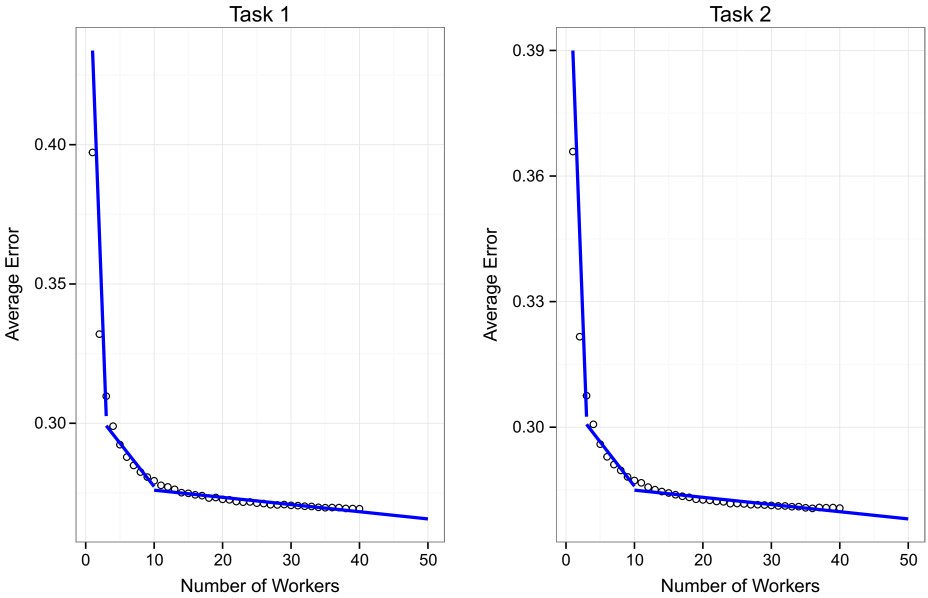


Fig. 12 Approximation of the error curves in Fig. 11 with 3 line segments

The range of the resulting proper scoring rule is [0, 10], which means that a worker could receive a bonus of up to 10 cents per reported prediction. It is worth mentioning that instead of reporting probability values directly, the workers reported their predictions by sliding a horizontal bar. The more a worker moved the bar towards the left side, the higher the probability value associated with the home team would be. Figure 10 shows an example of the graphical interface used in our experiments. For each possible outcome, the workers could also visualize the resulting bonus they would receive if the outcome turned out to be true.

6.2 Accuracy of aggregate outputs by the number of workers

In order to investigate the optimal number of workers a requester should hire in our third study, we run the same bootstrapping technique and segmented linear regression analysis described in Section 4.2 and 5.2. The main changes are that the number of workers n is inside the set $\{1, \ldots, 40\}$, and the MSE calculations involve probability values. Figure 11 shows the average error and the standard deviation of the errors for each prediction task and number of workers $n \in \{1, \ldots, 40\}$. Appendix B shows statistics regarding workers' errors.

Similar to the results from our first and second studies, the average error and the standard deviation of the errors decrease as the number of workers increases. Furthermore, the average error quickly converges to values greater than zero. In particular, the 99 % confidence intervals for $n = 40$ and Task 1 and 2 are, respectively, $[0.2690, 0.2696]$ and $[0.2804, 0.2809]$.

We also find the optimal number of workers a requester should hire using the segmented linear regression analysis suggested in Section 4.2. In particular, for both prediction tasks, we obtain that the optimal number of breakpoints is equal to 2 according to the dynamic algorithm by Bai and Perron [4]. Moreover, the resulting breakpoints for both prediction

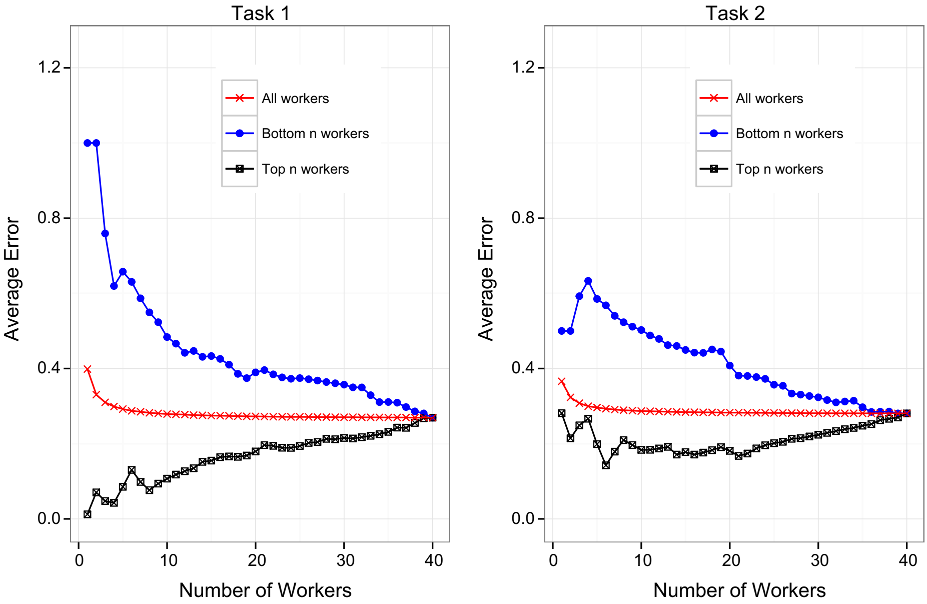


Fig. 13 The average error per task, number of workers, and different populations

tasks occur at the x-values 2 and 10. As a consequence, the optimal number of workers a requester should hire is 10. The closeness of this value to the previously found value of 11 in our first and second studies indicates the robustness of our approach. Figure 12 shows the error curves in Fig. 11 approximated by the resulting 3 line segments. The slopes of the first, second, and third line segments are inside the following ranges: [−0.044, −0.029], [−0.003, −0.002], and [−0.0003, −0.0001].

6.3 Accuracy of aggregate outputs from top-performing workers

For our third study, we also investigate whether a requester can increase predictive accuracy by combining only the predictions from the overall top-performing workers. Similar to Sections 4.3 and 5.3, we start by ranking workers based on their overall errors. An important difference, however, is that each overall error is now equal to the average of the squared differences between the 4 probability values reported by a worker and the 4 gold-standard (0/1) probability values.

Thereafter, for each one of the two prediction tasks and each number of workers $n \in \{1, \ldots, 40\}$, we randomly sample with replacement n workers' outputs from the full population of workers as well as the subpopulation defined by the top n workers. In total, we create 100,000 bootstrap resamples per combination of task, number of workers, and population. After aggregating the outputs in each bootstrap resample, we calculate the MSE between the aggregate output and the respective gold-standard output. Figure 13 shows the resulting average error per task, number of workers, and different populations.

We note that, for both tasks, no combination of top-performing workers results in a perfect aggregate output. Surprisingly, for Task 2, the average error considerably decreases first

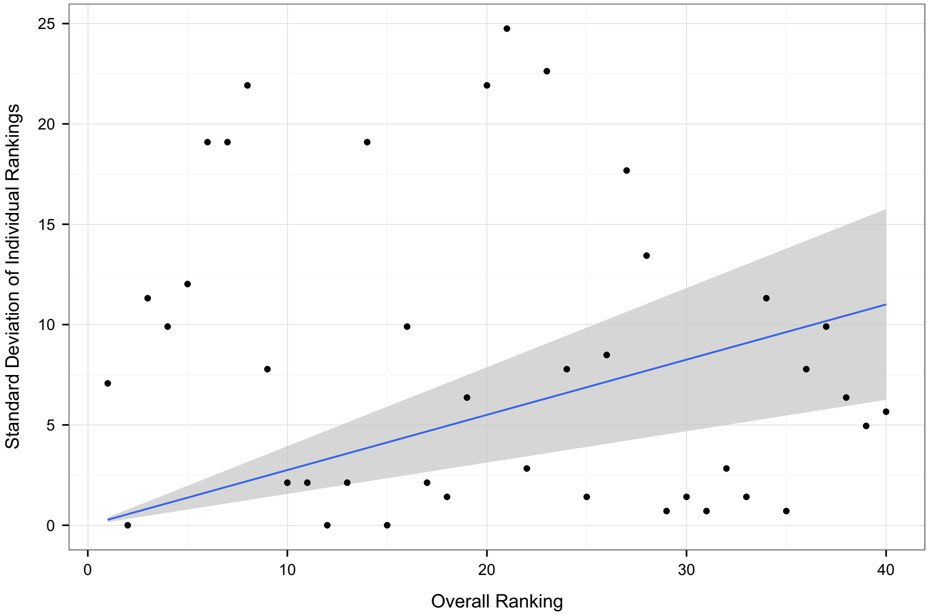


Fig. 14 The standard deviation of individual rankings as a function of workers' overall rankings

before non-monotonically increasing with more overall poor-quality workers. We believe this result happens due to the small number of games per task in our third study. We conjecture that, with more games per task, the average error would (almost monotonically) increase with more overall poor-quality workers. We also observe that, similar to our second study, the average error for the full population of workers is always greater than or equal to the average error for the top n workers (rank-sum test, p-value $< 10^{-15}$). As we mentioned in Section 5.3, this result might be due fact that payment schemes that induce honest reporting, such as proper scoring rules, result in less inconsistency in terms of performance across multiple tasks.

Finally, Fig. 13 also shows that an alternative analysis in terms of the bottom n workers supports our first analysis in that the average error from the bottom n workers is never less than the average error from the top n workers as well as from n random workers from the full population of workers.

6.4 On the consistency of workers across multiple tasks

For completeness' sake, we also investigate the consistency of workers' performance across the prediction tasks. Similar to Sections 4.4 and 5.4, we analyze the relationship between workers' overall rankings and the standard deviations of their individual rankings. Recall that the standard deviation of a worker's individual rankings is taken as a measure of how stable the overall ranking of that worker is. Figure 14 shows the standard deviation of individual rankings as a function of workers' overall rankings. We also fit a regression line through the origin to the data ($R^2 = 0.34$, sum of the squared residuals = 3176), and display its 95 % confidence interval. The resulting linear function is:

$$f(x) = 0.275 * x \tag{3}$$

Figure 14 shows that, in general, workers are more consistent in our third study than in our first and second studies. This is reflected in the coefficient of the regression line in (3), which means that an increase by 1 in a worker's overall ranking implies an expected increase of 0.275 in the standard deviation of that worker's individual rankings, the 99 % confidence interval being equal to [0.1109, 0.4393]. We note, however, that this result might be due to a smaller number of tasks (2 as opposed to 3) in our third study.

7 Conclusion

In this paper, we empirically studied the question of how many crowd workers a requester should hire. To this end, we started by investigating the influence of the number of workers on the accuracy of aggregate outputs. Specifically, we performed three studies involving different tasks and payment schemes. In our first study, we asked workers to complete three content-analysis tasks, and we paid workers using the canonical payment-per-completed task. Our second study involved the same three content-analysis tasks as in our first study, but we made extra payments based on the output-agreement method, which, under certain conditions, might induce honest reporting [8]. Finally, in our third study, we asked workers to complete two prediction tasks, and we made extra payments based on proper scoring rules [42], another technique which also induces honest reporting.

In all studies, we obtained that the average error of the aggregate output decreases with the number of workers. In other words, the expected accuracy of the aggregate output increases as the number of workers providing outputs increases. Our results also showed that there are diminishing returns for extra workers. Moreover, hiring extra workers also implies that the risk of obtaining a combination of exclusively poor-quality workers decreases because the standard deviation of errors in aggregate outputs decreases as the number of workers increases.

Thereafter, by employing a principled approach based on bootstrapping and segmented linear regression analysis, we found that the optimal number of workers a requester should hire for each task in each study was either 10 or 11. This is a rather surprising result in a sense that different tasks and payment schemes do not seem to influence the optimal number of workers much. The conclusions we derive on the number of workers to hire is a necessary first step in studying crowdsourcing. However, with any empirical study, the conditions we consider are by no means exhaustive and the optimal number of workers to hire may change with additional task conditions. As such, an interesting research direction would be to perform further empirical studies in order to validate and to better understand our initial results.

One can argue that a potential limitation of our analysis is that the optimal number of workers is defined solely in terms of expected errors in the aggregate output. We note, however, that one can easily incorporate costs into our analysis. In particular, after finding the optimal number of workers for a certain type of task, the requester can apply a very straightforward rule: given the current budget, the requester should hire as many workers as possible up to the optimal number of workers. Nonetheless, an interesting and relevant open question is on how to determine the optimal number of workers by also considering other criteria, such as risk reduction.

We then moved to analyze the question of whether it is beneficial to focus only on combinations of outputs from the overall top-performing workers. Generally speaking, we found that there is a considerable gain in expected accuracy when a requester combines outputs only from the overall top-performing workers. A potential explanation of this result is that the performance of overall top-performing workers tend to be more consistent across multiple tasks than the performance of other workers. Another interesting finding regarding consistency was that payment schemes that induce honest reporting, such as the output-agreement method and proper scoring rules, caused workers to be more consistent across multiple tasks.

Based on our results, our first recommendation for a requester designing a crowdsourcing process is: in the absence of prior knowledge about the accuracy of the crowd workers, hiring 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 benefits of hiring extra workers must be considered in practice. Our results showed that most of the (accuracy-wise) benefit occurs with the first 10 to 11 workers. Thereafter, the marginal benefit of hiring another worker is very low.

Our second recommendation for a more efficient design of crowdsourcing processes concerns the case when there exists prior knowledge about the accuracy of the crowd workers for specific tasks. In this case, the requester should focus only on combinations of the overall top-performing workers since this greatly reduces the expected error in the aggregate output. Our last recommendation is for a requester to use payment structures that induce honest reporting, such as the output-agreement method and proper scoring rules, since these techniques apparently also increase the consistency of workers' performance across multiple tasks.

As we have suggested throughout this section, our research opens up many avenues for future research. One particular direction concerns the use of monetary payoffs. In this regard, a valid and relevant research question is to what extent increasing monetary incentives would improve individual and collective accuracy. One can argue, for example, that individual accuracy increases with the amount of money. If that is the case, then less high-paid workers would be needed to achieve a certain accuracy in comparison to low-paid workers. This point raises the question: what is the relationship between the optimal group size and monetary incentives? In particular, from both cost and accuracy perspectives, would it be better to hire more low-paid workers or fewer high-paid workers?

On a final note, it is worth mentioning two limitations of our work that lead to an interesting research question. First, our analyses focused on simple averages to combine workers' outputs. Although simple averages have been shown to perform well empirically and to be robust when eliciting information from the crowd in different domains [14, 16], there are several other aggregation procedures, from voting protocols to sophisticated consensus-based algorithms [13]. Second, although our studies included different combinations of tasks and payment schemes, we by no means argue that those combinations are exhaustive. Thus, an exciting open question is whether the results obtained in our studies hold true in different settings, e.g., for different aggregation procedures, tasks, payment schemes, etc. An answer to this question is of great importance to the crowdsourcing community given its potential to create more effective crowdsourcing processes.

Acknowledgments The authors acknowledge the four anonymous reviewers, Nils Bulling, Craig Boutilier, Pascal Poupart, Daniel Lizotte, Hadi Hosseini, Xi Alice Gao, and the participants of the 12th European Conference on Multi-Agent Systems for useful discussions and comments. The authors thank Carol Acton, Katherine Acheson, Stefan Rehm, Susan Gow, and Veronica Austen for providing us with gold-standard outputs for the experiments in our first and second studies. The authors also thank the Natural Sciences and Engineering Research Council of Canada for funding this research.

Appendix A: Description of the texts in the first and second studies

We describe in this appendix the texts we used in our experiments in Sections 4 and 5. We also describe the gold-standard scores reported by the five professors and tutors, here called *experts*, from the English and Literature Department at the University of Waterloo in Canada.

Text 1. An excerpt from the "Sonnet XVII" by Neruda [28]. 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

Table 1 shows the experts' reported answers. The gold-standard answer for each question is the median/mode of the reported answers.

Table 1 Answers reported by the experts for Text 1

Criterion	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Median/Mode
Grammar	1	0	1	0	1	1
Clarity	2	2	2	1	2	2
Relevance	2	2	2	2	2	2

Text 2. An excerpt from "The Cow" by Taylor et al. [39]. 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.

Table 2 shows the experts' reported answers. The gold-standard answer for each question is the median/mode of the reported answers.

Table 2 Answers reported by the experts for Text 2

Criterion	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Median/Mode
Grammar	1	1	1	1	1	1
Clarity	2	2	2	1	2	2
Relevance	1	0	0	1	1	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

Table 3 shows the experts' reported answers. The gold-standard answer for each question is the median/mode of the reported answers.

Table 3 Answers reported by the experts for Text 3

Criterion	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Median/Mode
Grammar	0	1	0	0	0	0
Clarity	0	0	0	0	0	0
Relevance	0	1	0	0	0	0

Appendix B: Basic statistics regarding workers' errors

In this appendix, we present basic statistics regarding workers' errors. Specifically, for each task in each study, we calculate the MSE between each worker's output and the corresponding gold-standard output. Table 4 shows the resulting average MSE and standard deviation.

Table 4 Basic statistics regarding workers' errors

Study	Task	Average MSE	Standard deviation
Study 1	Task 1	0.71	0.79
	Task 2	0.57	0.43
	Task 3	0.73	0.74
Study 2	Task 1	0.63	0.63
	Task 2	0.51	0.49
	Task 3	0.55	0.72
Study 3	Task 1	0.4	0.24
	Task 2	0.37	0.21

References

1. von Ahn, L., Dabbish, L.: Designing games with a purpose. Commun. ACM **51**(8), 58–67 (2008)
2. Armstrong, J.S.: Combining Forecasts. In: Armstrong, J.S. (ed.) Principles of Forecasting: A Handbook for Researchers and Practitioners, pp. 1–19. Kluwer Academic Publishers (2001)
3. Bacon, D.F., Chen, Y., Kash, I., Parkes, D.C., Rao, M., Sridharan, M.: Predicting your own effort. In: Proceedings of the 11th International conference on autonomous agents and multiagent systems, pp. 695–702 (2012)
4. Bai, J., Perron, P.: Computation and analysis of multiple structural change models. J. Appl. Econ. **18**(1), 1–22 (2003)
5. Buhrmester, M.D., Kwang, T., Gosling, S.D.: Amazon's mechanical turk: a new source of inexpensive, yet high-quality, data. Perspect. Psychol. Sci. **6**(1), 3–5 (2011)
6. Carvalho, A.: Tailored proper scoring rules elicit decision weights. Judgment and Decision Making **10**(1), 86–96 (2015)
7. Carvalho, A., Dimitrov, S., Larson, K.: Inducing honest reporting without observing outcomes: an application to the peer-review process (2013). arXiv preprint arXiv: 1309.3197
8. 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)
9. Carvalho, A., Dimitrov, S., Larson, K.: A study on the influence of the number of mturkers on the quality of the aggregate output. In: Bulling, N. (ed.) Multi-agent systems, lecture notes in computer science, vol. 8953, pp. 285–300. Springer (2015)
10. Carvalho, A., Larson, K.: Sharing a reward based on peer evaluations. In: Proceedings of the 9th International conference on autonomous agents and multiagent systems, pp. 1455–1456 (2010)
11. Carvalho, A., Larson, K.: A truth serum for sharing rewards. In: Proceedings of the 10th International conference on autonomous agents and multiagent systems, pp. 635–642 (2011)
12. Carvalho, A., Larson, K.: Sharing rewards among strangers based on peer evaluations. Decis. Anal. **9**(3), 253–273 (2012)
13. Carvalho, A., Larson, K.: A consensual linear opinion pool. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp. 2518–2524 (2013)
14. Chen, Y., Chu, C.H., Mullen, T., Pennock, D.M.: Information markets vs. opinion pools: an empirical comparison. In: Proceedings of the 6th ACM Conference on Electronic Commerce, pp. 58–67 (2005)
15. Chiu, C.M., Liang, T.P., Turban, E.: What can crowdsourcing do for decision support. Decis. Support. Syst. **65**, 40–49 (2014)
16. Clemen, R.T.: Combining forecasts: a review and annotated bibliography. Int. J. Forecast. **5**(4), 559–583 (1989)
17. Dietterich, T.G.: Ensemble methods in machine learning. In: Multiple classifier systems, lecture notes in computer science, vol. 1857, pp. 1–15. Springer (2000)
18. 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)
19. Hansen, L.K., Salamon, P.: Neural Network Ensembles. IEEE Trans. Pattern Anal. Mach. Intell. **12**(10), 993–1001 (1990)
20. Hanson, R.: Combinatorial information market design. Inf. Syst. Front. **5**(1), 107–119 (2003)
21. 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)
22. Huang, S.W., Fu, W.T.: Enhancing reliability using peer consistency evaluation in human computation. In: Proceedings of the 2013 conference on computer supported cooperative work, pp. 639–648 (2013)
23. Ipeirotis, P.G.: Analyzing the amazon mechanical turk marketplace. XRDS Crossroads: The ACM Magazine for Students **17**(2), 16–21 (2010)
24. Ipeirotis, P.G., Provost, F., Sheng, V.S., Wang, J.: Repeated labeling using multiple noisy labelers. Data Min. Knowl. Disc **28**(2), 402–441 (2014)
25. 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)
26. Marge, M., Banerjee, S., Rudnicky, A.I.: Using the amazon mechanical turk for transcription of spoken language. In: Proceedings of the 2010 IEEE International conference on acoustics speech and signal processing, pp. 5270–5273 (2010)
27. Mason, W., Suri, S.: Conducting behavioral research on amazon's mechanical turk. Behav. Res. Methods **44**(1), 1–23 (2012)

28. Neruda: P.: 100 Love Sonnets. Exile (2007)
29. Oshiro, T.M., Perez, P.S., Baranauskas, J.A.: How many trees in a random forest In: Perner, P. (ed.) Machine learning and data mining in pattern recognition, Lecture notes in computer science, vol. 7376, pp. 154–168. Springer, Berlin (2012)
30. Paolacci, G., Chandler, J., Ipeirotis, P.G.: Running experiments on amazon mechanical turk. Judgment and Decision making **5**(5), 411–419 (2010)
31. Plous, S.: The Psychology of Judgment and Decision Making. Mcgraw-Hill Book Company (1993)
32. 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)
33. Ren, J., Nickerson, J.V., Mason, W., Sakamoto, Y., Graber, B.: Increasing the crowd's capacity to create: how alternative generation affects the diversity, relevance and effectiveness of generated ads. Decis. Support. Syst. **65**, 28–39 (2014)
34. Savage, L.J.: Elicitation of personal probabilities and expectations. J. Am. Stat. Assoc **66**(336), 783–801 (1971)
35. Selten, R.: Axiomatic characterization of the quadratic scoring rule. Exp. Econ **1**(1), 43–62 (1998)
36. Shaw, A.D., Horton, J.J., Chen, D.L.: Designing incentives for inexpert human raters. In: Proceedings of the ACM 2011 conference on computer supported cooperative work, pp. 275–284 (2011)
37. 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)
38. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: 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, pp. 254–263 (2008)
39. Taylor, J., Taylor, A., Greenaway, K.: Little Ann and Other Poems. Nabu Press (2010)
40. 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)
41. Winkler, R.L., Clemen, R.T.: Multiple experts vs. multiple methods: combining correlation assessments. Decis. Anal **1**(3), 167–176 (2004)
42. Winkler, R.L., Murphy, A.H.: "Good" Probability Assessors. J. Appl. Meteorol **7**(5), 751–758 (1968)
43. 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)
44. Zeileis, A., Leisch, F., Hornik, K., Kleiber, C.: strucchange: an R package for testing for structural change in linear regression models. J. Stat. Softw **7**(2), 1–38 (2002)
45. Zhang, H., Horvitz, E., Parkes, D.: Automated workflow synthesis. In: Proceedings of the 27th AAAI conference on artificial intelligence, pp. 1020–1026 (2013)