Picking Peaches or Squeezing Lemons: Selecting Crowdsourcing Workers for Reducing Cost of Redundancy

⋆

Paulina Adamska[1 [0000−0002−8391−9877]], Marta Juźwin[1 [0000−0002−4243−2229]], and Adam Wierzbicki[1 [0000−0003−0075−7030]]

Polish-Japanese Academy of Information Technology, Warsaw, Poland
{tiia,marta.juzwin,adamw}@pja.edu.pl

Abstract. Crowdsourcing (CS) platforms are constantly gaining attention from both researchers and companies, due to the offered possibility of utilizing the "wisdom of crowds" in order to solve a great variety of problems. Despite the obvious advantages of such mechanisms, there are also numerous concerns regarding the quality assurance of work results produced by a large group of anonymous workers. In this work, we use data gathered from a real CS platform in order to study the performance of various approaches to worker selection, including a novel approach that utilizes automatic real-time monitoring of the produced results. We compare the performance of these mechanisms with respect to both overall cost and the accuracy of the final results to benchmark algorithms that aggregate results from a group of workers without pre-selection, relying solely on the "wisdom of crowd". We find that our novel approach is capable of reducing the cost of obtaining high-quality aggregated results by a factor of four, without sacrificing quality.

Keywords: crowdsourcing · quality · effort · skill · aggregation · algorithm

1 Introduction

Crowdsourcing (CS) platforms offer companies a possibility to delegate some tasks to a potentially large group of anonymous workers, using the "wisdom of crowd" to solve a great variety of problems. Despite numerous advantages of this model, one major concern of each potential requester is quality control. New requesters are often forced to pay third parties for the information regarding former performance of CS workers, or apply custom techniques such as: (1) utilizing custom skill tests in order to verify worker qualifications, (2) providing "benchmark" tasks, with already known solution, which are later on utilized

⋆ This work is supported by the Polish National Science Centre grant 2015/19/B/ST6/03179.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_38

to monitor worker reliability and attitude to solving "real" tasks, (3) requesting a certain amount of redundant solutions hoping to reduce the impact of low quality submissions on the aggregated final result. In this work we analyze the impact of various approaches to worker selection on the accuracy of the final outcome, and the overall requester costs. We also propose and evaluate an experimental real-time worker performance monitoring technique that involves machine-learning-based solution quality estimation, and prevents further interaction with an unreliable worker. In order to obtain results that are as realistic as possible, the performance of all the previously mentioned techniques was verified using a dataset gathered from a popular CS platform.

2 Related work

There have been many suggestions regarding quality control of the content produced by CS workers. Most of them could be assigned to one of two groups of approaches. The first group consists of research concentrated on improving initial worker selection process [5, 12, 14, 20]. The second group of approaches is more similar to our own, as these mechanisms attempt to reduce the amount of tasks solved by unreliable CS workers. For example Hirth et al. [7] attempted to use validation tasks in order to detect cheaters among CS workers. On the other hand, our research aims to detect spammers using only one validation task, which in turn helps to reduce costs. Moreover our model does not assume that only cheaters can provide low-quality solutions. Hirth et al. compared his method to a simple majority voting approach, while our algorithm is validated using a state-of-the-art EM-based aggregation algorithm. Hirth et al. [6] also proposed to detect spammers using application level information gathering mechanism that is similar to the one used in our experiment as it gathers data regarding worker interaction with the application UI. Our proposal, however, uses a different set of features in order to estimate solution accuracy. Moreover, our classifier provides the actual accuracy estimation, not a binary classification ("qualifie" or "non-qualified"), which allows the requester to control the required accuracy threshold. Additionally, the approach proposed by Hirth et al. was verified on a significantly smaller sample of CS workers (behavioral traces from 215 workers were distributed between training and test dataset).

Liu et al. [11] attempted to answer the question of how many control tasks are required in order to identify reliable workers. The proposed Gaussian model assumes utilization of such variables as bias or label variance, which can be learned from the results of the control tasks and remain constant. In our experiment, we concentrate more on user expertise (which is assumed to be constant) and effort level (which may change over time). Feldman et al. [2] proposed a quality control mechanism that attempted to monitor user interaction with UI components (such as mouse speed acceleration, scrolling speed, etc.) to automatically detect a decrease in work quality in real-time. This approach differs from our proposal in terms of features used for performance prediction. Moreover, authors did not provide any cost analysis. Rangi and Franceschetti [16] modeled

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_38

task assignment in a budget-limited setting as a Bounded Knapsack Problem. This solution however would require some major modifications in the way that requesters and workers interact with each other. Shanshan et al. [18] proposed an algorithm, that computed a proper ordering of workers' submissions based on the estimated quality of the solutions. Unlike our approach, this mechanism assumed availability of requester's feedback for all the solutions provided by the workers in the past. This assumption is realistic for the "creative" tasks, for which the algorithm was designed (like brand logo design), however may not be feasible for a typical CS task. Moreover, for such tasks it may be difficult to objectively evaluate the real quality of the solutions.

3 Experiment Design

In order to gather all data required for our analysis, we decided to study the behavior of CS workers using a popular general-purpose crowdsourcing platform. An experiment has been carried out on Amazon Mechanical Turk.[1] Our analysis is based on 8100 work results provided by 810 different CS workers - each experiment participant was asked to provide solutions to 10 Human Intelligence Tasks (HITs).

Our experimental CS task had a very high level of redundancy, as each job could have up to 810 solutions. This unique experimental approach allowed us to repeatedly sample our dataset to simulate a situation when requesters interact with different, smaller groups of workers. In a realistic case, a requester would require a much smaller redundancy (perhaps 5 solutions for each job) and would interact with a much smaller population of workers. Every time a requester uses Amazon Mechanical Turk, she may encounter a different group of workers. This means that every time, the requester has to fit workers models from scratch, encountering a cold-start problem. Our approach allows to test robustness of models to the composition of a training set of workers.

We wanted our worker sample to reflect the population of workers on Amazon Mechanical Turk as faithfully as possible. The only worker filtering method that we have applied before performing the experiment has been based on worker location - workers from outside of U.S. were not allowed to participate. This decision was made due to the nature of the chosen task, which required language-related skills. Experiment participants were asked to identify and mark misspelled words in English texts in order to help the requester evaluate the performance of different algorithms designed for automatic text processing. All the texts were excerpts of books selected from the Gutenberg project website.[2] The spelling errors were added manually by replacing one word with a similar one that was grammatically or semantically incorrect in a particular context. For example, the word "cut" was replaced by "cat", etc. The general concept behind this approach was to mimic spelling errors that could be caused by a poorly designed OCR tool, and more importantly - to eliminate the possibility of using spell

[1] https://www.mturk.com/mturk/welcome
[2] https://www.gutenberg.org/

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_38

check tools without reading and understanding the entire text. Our goal was to create a task of medium difficulty that allowed us to precisely measure both CS workers expertise, and the accuracy of the provided solutions. Each experiment participant was asked to solve 10 HITs. Additionally, workers were asked to solve a time-limited skill test (before solving the first HIT). All the stages of the experiment are described in detail in the following sections.

3.1 Skill test

The main goal of this part of the experiment was to determine the real expertise level of each worker regarding this particular task type. In order to achieve this goal, the experiment participants were asked to mark all the words that contained spelling errors in the presented text. The total number of words was 127, and 9 of them were misspelled. The time limit for completing the test was set to one minute, and after this time workers were automatically redirected to the main task without the possibility of reviewing their solutions. For each worker we have recorded the time required to submit the solution along with the computed skill (interpreted as the accuracy of the results with respect to our 'gold solution').

3.2 Human Intelligence Task

In the main task, CS workers were asked to compare a pair of texts in terms of spelling errors and to provide two types of feedback. The first one was to indicate which of the texts contained more misspelled words. The second one was to mark the misspelled words in each text. Such task design not only allowed us to gather the required data, but also seemed to be a typical type of work for a CS setting. For example, Good et al. [4] used word tagging to capture mentions of disease in PubMed abstracts, while Finin et al. [19] applied the same technique to collect named entities annotations for Twitter status updates. Klebanov et al. [10] asked CS workers to mark words that most contributed to the overall sentiment of a sentence in order to construct subjectivity lexicon for recognizing sentiment polarity in essays. Similarly Hsueh et al. [9] utilized word tagging in order to consider the problem of classifying sentiment in political blog snippets. On the other hand, Filatova [3] performed a CS experiment that involved text tagging, aiming to create a corpus that could be used for identifying irony and sarcasm, and Prabhakaran et al. [15] attempted to collect textual data, that could be used as a training set for an automatic modality tagger.

Figure 1 shows how a pair of texts for comparison was presented to the workers. In order to gather at least some additional information about the worker effort, we divided each text into 5 parts that could be displayed by clicking the "Next" button. We have recorded the timestamp for almost every user interaction with our application, and thus we could find out whether the worker at least had a chance to see the entire text before submitting the answer, and for how long each part was displayed on the screen. The average number of words in the evaluated texts (considering all of the 10 available HITs) was approximately 120 (the shortest text consisted of 95, and the longest one of 130 words). Each

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_38

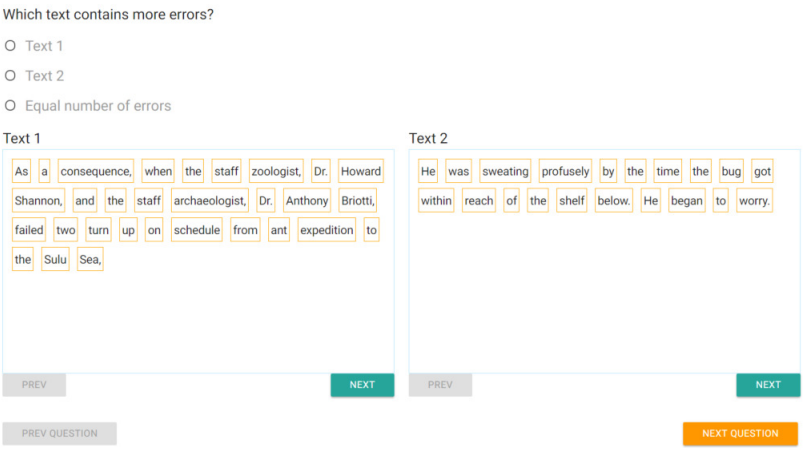

Fig. 1. A sample text comparison task UI

pair of texts contained 10 misspelled words, and the distribution of errors was designed in such a manner that the number of the read text parts affected the quality of the final solution. Experiment participants were allowed to review the solutions submitted earlier and correct them as many times, as they wanted. The average time of completing all 10 HITs was about 30 minutes, which was approximately equal to the expected completion time.

4 Worker characteristics and solution quality

Existing theoretical models often assume that the skill level is strongly correlated with the probability of providing a high-quality answer [8, 13]. The results gathered in our experiment indicated that the correlation between skill and accuracy in the real life setting is not particularly high. We believe that this is caused by the fact that worker skill is not the only factor contributing to the accuracy of the submitted solutions. Table 1 presents inconsistencies between the skill-based and observed-accuracy-based classifications.

Note that in each group identified during the skill test (skill column), there are CS workers, whose observed behavior while working on the real tasks was incompatible with the initial skill-based prediction. Particularly, among the workers who turned out to provide the best solutions for almost all tasks, most have initially been identified as regular workers. Moreover, about 8% of the workers who have been assigned to the group of experts actually turned out to be spammers in real tasks. This phenomenon can be associated with the presence of smart adversaries in the population. Such workers put more effort into a task that is officially referred to as a "skill test", but their performance degrades significantly for ordinary tasks. Both observations lead to the conclusion that solely using a

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_38

Table 1. Skill-based and observer-accuracy-based classification

Label		No. of workers	% of workers	Time
Skill	Observed			*(average)*
expert	expert*	39	5	20
($skill > 0.8$)	regular$^{\diamond}$	27	3	18
	spammer†	6	1	4
regular	expert*	157	19	27
($0.3 \leqslant skill \leqslant 0.8$)	regular$^{\diamond}$	328	40	24
	spammer†	135	17	9
spammer	expert*	7	1	25
($skill < 0.3$)	regular$^{\diamond}$	49	6	27
	spammer†	62	8	11

* within 25% of workers with the highest accuracy in more than 7 tasks
$^{\diamond}$ within 25% of workers with the highest accuracy in 1 to 7 tasks
† never in the group of 25% of workers with the highest accuracy
(all the differencies are statistically significant according to the Kruskal-Wallis test)

skill test may not be the best strategy to recognize the most reliable workers and thus achieve the best results. One additional observation can be made based on the data presented in Table 1. The differences in average task completion time for the group of spammers identified by the skill test suggest that about 44% of workers who did not perform well during the time-limited skill test were actually willing to compensate for the lack of skill with a certain degree of effort, achieving at least reasonable accuracy while solving the real tasks. To sum up, the presented results indicate that there is no single, simple feature that would allow to initially select a group of reliable CS workers, simply by setting up a fixed threshold. Therefore, the requesters should probably consider approaches that not only assess worker skill, but also monitor the amount of time spent on solving the task.

5 Analyzed state-of-the-art approaches

Apart from proposing an experimental approach to improving the quality of the information produced by CS workers, in our study we also analyzed three alternative state-of-the-art approaches. These mechanisms are described in the following sections.

5.1 Squeezing lemons

The first and simplest state-of-the-art approach introduces some redundancy level with no initial worker selection. Removing the initial selection process allows to gather the results as fast as possible, due to the lack of additional requirements regarding worker skills. On the other hand, having several solutions

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_38

to the same task allows the requester to use aggregation algorithms to generate a single output from multiple solutions, hopefully reducing the impact of potential spammers or workers who do not have a desired skill level. In this article, we are using a state-of-the-art EM-based algorithm to aggregate all solutions into a single final one. The algorithm was proposed by Raykar et al. [17] and is a generalization of the approach proposed by Dawid and Skene [1]. Note that the algorithm is not only able to estimate the ground truth label for each word, but also simultaneously estimates features of the CS workers defined by formula 1 and 2 respectively:

$$\alpha^j \coloneqq P[y_i^j = 1 \mid y_i = 1] \sim Beta(2.078, 1.843) \tag{1}$$

$$\beta^j \coloneqq P[y_i^j = 0 \mid y_i = 0] \sim Beta(2.078, 1.843) \tag{2}$$

where y_i is the true label of word i, y_i^j is the label assigned to word i by worker j. The parameters of both distributions were computed using the information about the accuracy of skill test results for all the workers.

The aggregation algorithm described in this section is used in all cases in our research; however, in the "squeezing lemons" scenario, the algorithm aggregates results from workers who have not undergone any pre-selection, while in the "peach picking" scenario we apply various approaches of worker selection before we aggregate results from selected workers.

5.2 Peach picking

Skill-test-based selection only This technique is a typical, well-known approach to improve the quality of the redundant solutions provided by CS workers. The main idea behind this concept is that the reason for poor quality of some solutions in a CS setting is the lack of sufficient skills. Therefore, the workers are asked to provide solution to a skill test before they are offered to solve an ordinary task. If the skill of worker w denoted by $skill_w$ is lower than the threshold denoted by $threshold_{skill}$, then the worker does not have the required expertise level to provide valuable contributions. The number of workers selected depends on the specified redundancy level.

In our setting, skill is interpreted as the solution accuracy achieved in the skill test. The accuracy evaluation in our experiment is in many ways similar to binary classifier evaluation, as we have a set of words, which are supposed to be labelled as correct or misspelled by different workers. Therefore we define the following variables: the number of false positives (FP) - the number of correct words which were marked as misspelled; the number of true positives (TP) - the number of misspelled words, which were marked as misspelled; the number of false negatives (FN) - the number of misspelled words that were not marked as misspelled; the number of true negatives (TN) - the number of correct words that were not marked as misspelled. Based on these variables, we calculate the F1 score as a measure of correctness of a worker's solutions.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_38

First-task-based selection only The second of the the studied "peach picking" techniques is based on the idea, that skill-test-based initial filtering may not be the best selection method, as workers may apply a different approach to such task. In our experiment, CS workers were not implicitly informed, whether the skill test results will or will not have any impact on their future work opportunities, however the task was time-limited, and it was called a "skill test", thus potential spammers were aware of the fact, that the main goal of this task was to identify them as reliable, or unreliable workers. In the first-task-based filtering setting, CS workers are not able to distinguish between the "skill test" and the ordinary tasks, as they both look exactly the same. The selection algorithm itself is the same as in the approach described in section 5.2. The requester defines a threshold $threshold_{first}$. If the accuracy of the first task solution is at least at the $threshold_{first}$ level, the author of this solution is offered to solve all the ordinary tasks. The number of workers that are selected depends on the specified redundancy level.

6 Experimental real-time monitoring approach

There is one main disadvantage of the peach picking techniques mentioned in section 5.2. Such approaches do not attempt to verify whether the quality of the solutions provided by CS workers is not changing. Changes in work results quality may occur due to various reasons. One of them may be CS workers deliberately putting more effort into a skill test, and then switching to a spammer-like behavior while solving ordinary tasks. Second reason may be CS workers not performing particularly well when solving a time-limited skill test, but being honest enough, to put as much time and effort as is required for them to accurately solve ordinary tasks. In such situations inflated skill thresholds used in conjunction with initial worker selection techniques automatically reject such workers, thus unnecessarily increasing selection costs. Another possible reason for accuracy changes may be tiredness.

As mentioned in section 4, we seem to have observed all of the previously mentioned phenomena in our experiment. Approximately 8% of workers, identified as experts solely based on their skill test result, behaved like spammers when solving ordinary tasks. On the other hand, there were much more workers that behaved like experts among the skill-test-based identified regular workers in comparison to the group of skill-test-based identified experts. When the requester applies extremely inflated initial filtering threshold, all of these workers do not even get the chance to showcase their real potential and gain reputation they deserve. We have also noticed, that there were more low quality solutions for the last few tasks, which does not have to, but may indicate tiredness.

In order to address these issues we have decided to propose a novel automatic real-time solution accuracy monitoring in our experimental approach. This technique allows the requester to decide after each task, whether to further interact with this particular CS worker. Moreover it does not require to solve the exploration vs exploitation problem, unlike the mechanisms that use multiple "gold

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_38

Algorithm 1 Real-time monitoring

```
T: a set of ordinary tasks
W: a set of selected workers
for all w ∈ W do
  for all t ∈ T do
    solution_t^w ← w.solve(t)
    accuracy_t^w ← solution_w^t.estimateAccuracy(solution_w^t)
    if accuracy_w^t ≥ threshold_estimated then
      solution_t^w.setAccepted(true)
    else
      solution_t^w.setAccepted(false)
      w.setSuspended(true)
      break
    end if
  end for
end for
```

standard" tasks, which are presented to the workers among plain tasks in order to verify their current performance. The proposed approach consists of two main components, namely (1) initial worker selection mechanism, and (2) real-time solution accuracy estimation mechanism. Both of them are described in the following sections.

Initial worker selection Initial worker selection mechanism is based on the accuracy of the first task solution, as described in section 5.2. We have chosen this approach as it turned out to outperform skill-test-based filtering both in terms of cost and accuracy in most cases. The number of workers, that are going to be selected is determined by the redundancy level specified by the requester. Even though skill is not considered in the initial filtering phase, each worker is asked to solve a short skill test as well. This information will be used by the real-time solution accuracy estimator.

Real-time solution accuracy estimation Each worker $w \in W$, that has made it through the initial selection process, is offered to solve the first ordinary task. When the task is solved, a dedicated mechanism estimates the accuracy of the solution. If the solution accuracy exceeds the threshold $threshold_{estimated}$ specified by the requester, it is accepted, and CS worker is offered to solve the next task. Otherwise the solution is not accepted, and further interaction with this worker is suspended. Note that the initial filtering threshold $threshold_{first}$ described in section 5.2 and the accuracy threshold $threshold_{estimated}$ used in the real-time evaluation process may be assigned different values.

We have utilized a random forest regression model to provide the accuracy estimation functionality. The model was trained based on the results of the first 405 HITs, while the simulation of our approach was done on the remaining 405 HITs. The optimal model has been selected using the RMSE metric during the 5 fold cross-validation process (grid search method). The RMSE values for the training and test datasets are 0.1447529, and 0.171867 respectively. The estimated accuracy is computed using the following features of each CS worker:

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_38

Algorithm 2 Acquiring solutions to the remaining tasks

```
T: a set of all ordinary tasks
T_unsolved: a set of ordinary tasks without any accepted solution
W: a set of selected workers
S < T, W >: a map of solutions provided by workers to all tasks
for all w ∈ W do
  for all t ∈ T_unsolved do
    solution_t^w ← S < t, w >
    if ¬(solution_t^w ∈ S < T, W >) then
      solution_t^w ← w.solve(t)
    end if
    solution_t^w.setAccepted(true)
  end for
end for
```

skill, first task solution accuracy, first task completion time, current task no. in the set of all tasks provided by the requester, current task completion time, difference between the completion time of the first task and completion time of the current task, percentage difference between the completion time of the first task and completion time of the current task. The overview of the entire procedure is presented in Algorithm 1. Due to the probabilistic nature of the automatic evaluation process, CS worker receives monetary reward for submitting a solution regardless the accuracy estimation result.

If all of the selected workers have already been suspended based on the real-time accuracy estimation and there are still some unsolved tasks, then for each unsolved task the following procedure is executed: (1) if there are some solutions to this task that have already been provided and have not been accepted they are automatically marked as accepted, (2) all the suspended workers, that have not already solved the task are offered to submit their solutions, which are later on automatically accepted. The overview of the entire procedure is presented in Algorithm 2. The main idea behind this approach is the assumption, that at this point none of the selected workers can still be considered credible, as each of them has been suspended at some point of interaction. Therefore we need to maximize the redundancy level for this particular task.

7 Picking peaches or squeezing lemons?

In order to compare the accuracy and cost of the techniques described in sections 5.1 and 5.2, the entire dataset containing solutions to 10 HITs provided by 810 workers has been divided in two parts. The first 405 workers have been assigned to a training dataset which has only been used to train the worker accuracy classifier for our experimental approach. The remaining 405 workers were assigned to the test dataset which has been used in order to verify the performance of all the studied approaches. For every approach a simulator has been used to generate the data representing a single scenario of interaction between

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_38

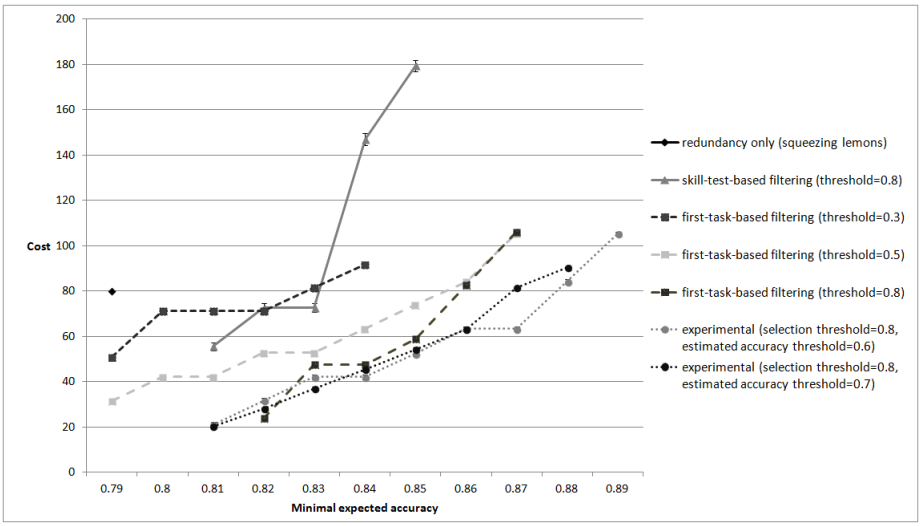

Fig. 2. Comparison of the analyzed approaches in terms of average minimum cost required to achieve a required accuracy of aggregated results. Confidence intervals' width for cost on the figure does not exceed: 5.5 for skill-test-based filtering, 1.4 for first-task-based filtering, and 1.65 for the experimental approach.

the requester and the CS workers (by scenario we mean requester submitting the tasks and recruiting workers to provide solutions to all of them at a specified redundancy level). At each simulation run workers were chosen one at a time, at random from the entire test dataset, ensuring that none of the workers could be chosen multiple times within a single simulation run. If the chosen worker had made it through the initial filtering phase, he was offered to solve ordinary tasks according to the rules of a particular tested approach. The workers were paid $1 for every submitted solution, regardless of whether it has been accepted or not. At the end of each simulation run the following actions were taken:

1. All results available at the specified redundancy level were aggregated using the state-of-the art aggregation algorithm chosen as a benchmark in this article [17] (see Section 5.1).
2. The accuracy of the solution was computed using the 100% correct result as a reference.
3. The total costs were summed up and recorded.

For all the approaches apart from the experimental one, involving real-time monitoring feature, 300 simulation runs were used. For the more complex experimental approach described in section 6, 1000 simulation runs were used. For each of the analyzed approaches we compute the average accuracy of all aggregated HIT solutions from all simulation runs, and average costs from all the simulation runs. We also computed confidence intervals.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_38

The general comparison of all the analyzed approaches is depicted in figure 2. The figure shows the performance of all approaches for various levels of redundancy chosen so that the approach reaches a threshold of accuracy. The average minimum cost of each approach for reaching a certain accuracy is plotted on the figure.

It turned out, that for our task, the best accuracy that was achieved by the "squeezing lemons" (aka redundancy-only-based) approach was 79%. The minimum redundancy level required for this was 8 solutions per task, which translated to the minimum cost of 80.

Our study of the state-of-the-art "picking peaches" approaches shows, that the initial worker filtering based on skill test does improve the average quality of the aggregated final result assuming, that the $threshold_{skill}$ value specified by the requester is set to 0.8. For this technique higher accuracy (81%) can be achieved at a redundancy level set to only 3 solutions per task which translates to the cost of approximately 56. The drawback of this approach is that the cost associated with the worker selection process is noticeably higher, which causes the costs to dramatically increase for higher redundancy levels.

On the other hand, filtering based on the accuracy of the first task seems to significantly improve the quality of the aggregated task solutions even when $threshold_{first}$ is set to as low value as 0.3. This threshold setting can be thought of as an attempt to remove spammers from the worker pool. When the requester sets $threshold_{first}$ to 0.8, a major quality improvement of the average aggregated outcome can be achieved at a noticeably lower cost than for the skill-based filtering with the same $threshold_{skill}$ value. This "picking peaches" technique allows to achieve the accuracy of 82%, when the redundancy level is set to only 2 solutions per task, which translates to a cost of approximately 24.

For the experimental approach, we have found the following parameter settings to be optimal: (1) $threshold_{first}$ for the first task accuracy (initial worker selection phase): 0.8, $threshold_{estimated}$ for estimated ordinary task solution accuracy (real-time quality control): 0.6 and (2) $threshold_{first}$ (initial worker selection phase): 0.8, $threshold_{estimated}$ (real-time quality control): 0.7. For both settings of parameters, the experimental approach outperforms first-task-accuracy-based filtering with the same $threshold_{first}$ value in terms of average costs. An accuracy of 81% is achieved by the experimental approach with a cost of 20. The experimental approach consistently provides similar or better results in terms of accuracy as compared to other approaches. The benefits of applying real-time monitoring become particularly apparent for higher redundancy levels that achieve higher accuracy. Pre-selecting workers based on the accuracy of the first task allows to achieve a maximum accuracy of 87% at a cost of 105, compared to the benchmark accuracy of 79% at a cost of 80 achieved by the "squeezing lemons" approach without worker selection. On the other hand, the experimental approach achieves an accuracy of 87% at a cost of 61, and exceeds the benchmark accuracy of 80% at the cost of 20 (as compared to the benchmark cost of 80).

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_38

8 Conclusions

The majority of research on crowdsourcing has focused on aggregation algorithms, following the seminal work of Dawid and Skene [1]. A possible reason for this trend is the widespread belief in the "Wisdom of Crowds". Our work confirms the fact that increasing redundancy can increase quality of results, but we propose to filter workers who will contribute results that will be the basis of final aggregation. While this idea has been studied before, our research is one of the first systematic and large-scale studies of the subject.

We have conducted an experimental CS task on Amazon Mechanical Turk, mimicing typical CS task design. Our experimental task had a very high level of redundancy: 810 workers solved all our tasks. This approach allowed us to conduct data-driven simulation studies that compared various approaches of workers selection with respect to quality of aggregated results and cost of obtaining the required work results. As a benchmark, we have used a state-of-the-art aggregation algorithm [17] without worker filtering.

Our results indicate that worker pre-selection based on the results of the first real task can outperform aggregation of results without worker filtering: picking peaches is indeed better than squeezing lemons. However, not all peach picking methods work equally well. In particular, we have shown that that CS workers can manipulate simple skill tests, and pre-selecting workers based on results of first real task is more effective than using skill-test results. Furthermore, we have proposed a novel, real-time dynamic filtering algorithm that significantly increases the cost-effectiveness of crowdsourcing. Our algorithm improves on the cost of the benchmark algorithm by a factor of four without decreasing quality. The algorithm also has good results for higher redundancy levels that achieve higher quality, outperforming simple worker pre-selection by a factor of two.

One of the limitations of our study is that we have considered only 10 HITs for each worker. This choice was motivated by the belief that more HITs would not change our results. However, a longtitudinal study of CS workers (perhaps across multiple tasks) could possibly allow to create more in-depth models of CS workers that would further improve the ability of filtering workers for crowdsourcing tasks.

References

1. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the em algorithm. Applied Statistics **28**(1), 20–28 (1979)
2. Feldman, M., Bernstein, A.: Behavior-based quality assurance in crowdsourcing markets. In: Conference on Human Computation & Crowdsourcing 2014. s.n., Pittsburgh, USA (November 2014)
3. Filatova, E.: Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In: In Language Resources and Evaluation Conf. , LREC2012 (2012)
4. Good, B.M., Nanis, M., Su, A.I.: Microtask crowdsourcing for disease mention annotation in pubmed abstracts. CoRR **abs/1408.1928** (2014)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_38

5. Hassan, U., Curry, E.: A capability requirements approach for predicting worker performance in crowdsourcing. In: 9th IEEE Int. Conf. on Collaborative Computing: Networking, Applications and Worksharing. pp. 429–437 (2013)
6. Hirth, M., et al.: Predicting result quality in crowdsourcing using application layer monitoring. In: 2014 IEEE Fifth International Conference on Communications and Electronics (ICCE). pp. 510–515 (July 2014)
7. Hirth, M., Hoßfeld, T., Tran-Gia, P.: Analyzing costs and accuracy of validation mechanisms for crowdsourcing platforms. Mathematical and Computer Modelling **57**(11-12), 2918–2932 (2013)
8. Ho, C., Jabbari, S., Vaughan, J.: Adaptive task assignment for crowdsourced classification. In: Proc. of the 30th Int. Conf. on Machine Learning - Volume 28. pp. I–534–I–542. ICML'13, JMLR.org (2013)
9. Hsueh, P.Y., Melville, P., Sindhwani, V.: Data quality from crowdsourcing: A study of annotation selection criteria. In: Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing. pp. 27–35. HLT '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009)
10. Klebanov, B.B., Burstein, J., Madnani, N., Faulkner, A., Tetreault, J.R.: Building subjectivity lexicon(s) from scratch for essay data. In: Computational Linguistics and Intelligent Text Processing - 13th International Conference, CICLing 2012, New Delhi, India, March 11-17, 2012, Proceedings, Part I. pp. 591–602 (2012)
11. Liu, Q., Ihler, A.T., Steyvers, M.: Scoring workers in crowdsourcing: How many control questions are enough? In: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (eds.) NIPS. pp. 1914–1922 (2013)
12. Liu, Q., Li, H.: Cheaper and better: Selecting good workers for crowdsourcing. In: HCOMP. p. 20–21 (2015)
13. Mavridis, P., Gross-Amblard, D., Miklós, Z.: Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. In: Proceedings of the 25th International Conference on World Wide Web. pp. 843–853. WWW '16, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2016)
14. Moayedikia, A., Yeoh, W., Ong, K.L., Boo, Y.L.: Improving accuracy and lowering cost in crowdsourcing through an unsupervised expertise estimation approach. Decision Support Systems (2019)
15. Prabhakaran, V.e.a.: Statistical modality tagging from rule-based annotations and crowdsourcing. In: Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics. pp. 57–64. ExProM '12, Association for Computational Linguistics, Stroudsburg, PA, USA (2012)
16. Rangi, A., Franceschetti, M.: Multi-armed bandit algorithms for crowdsourcing systems with online estimation of workers' ability. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. p. 1345–1352. AAMAS '18, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2018)
17. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. J. Mach. Learn. Res. **11**, 1297–1322 (Aug 2010)
18. S. Lyu, .: Learning representations for quality estimation of crowdsourced submissions. Inf. Process. Manage. **56**(4), 1484–1493 (2019)
19. T. Finin, e.a.: Annotating named entities in twitter data with crowdsourcing. In: Proc. of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk. pp. 80–88. CSLDAMT '10 (2010)
20. Yang, P., et al.: Identifying the most valuable workers in fog-assisted spatial crowdsourcing. IEEE Internet of Things Journal (2017)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50417-5_38