Predicting Result Quality in Crowdsourcing Using Application Layer Monitoring

Matthias Hirth, Sven Scheuring, Tobias Hoßfeld, Christian Schwartz, Phuoc Tran-Gia
Chair of Communication Networks
University of Wuerzburg
Germany
Email: matthias.hirth@informatik.uni-wuerzburg.de

***Abstract*—Crowdsourcing has become a valuable tool for many business applications requiring to meet a certain quality of the results generated by the workers. Therefore, several quality assurance mechanisms have been developed which are partly deployed in commercial crowdsourcing platforms. However, these mechanisms usually impose additional work overhead for the worker, e.g. by adding test questions, or increase the costs for the employer, e.g. by replicating the task for majority decisions. In this work, we analyze the applicability of implicit measurements to objectively estimate the quality of the workers' results. First efforts in this area have already been made by investigating the impact of the task completion time. We extend this research by deploying an application layer monitoring (ALM), which enables monitoring the workers' interactions with our task interface on a much more detailed level. Based on an exemplary use case, we discuss a possible implementation and demonstrate the potential of the approach by predicting the quality of the workers' submission based on our monitoring results. This ALM provides a new way to identify low quality work as well as difficulties in fulfilling the formulated tasks in the domain of Crowdsourcing.**

I. Introduction

The easy and cost effective access to huge and scalable computation power provided by machine clouds has been the driver for the development of many new services and fostered the growth of numerous new startups. In contrast to computational power, human workforce is still a sparse resource, especial for smaller companies. Here, Crowdsourcing, leveraging the possibility to access a huge number of people via the Internet, can help to create a human cloud, which offers a scalable and easily accessible human workforce [1].

An essential part of this approach are Crowdsourcing platforms which act as mediator between the users submitting work (*employers*) and the users willing to complete the submitted work (*workers*) for a monetary compensation. Compared to traditional forms of work organization, the work units here are much smaller. The workers can usually complete the *tasks* within a few minutes to a few hours and are usually paid a few cents to a few dollars [2].

In contrast to machine clouds, the quality of the results obtained from human clouds can vary significantly. The main reason for this is the fact that Crowdsourcing tasks are completed remotely by anonymous workers without any supervision. This anonymity fosters cheating among the workers, i.e. they try increasing their income by intentionally using malicious techniques. Studies have shown that even small incentives encourage this behavior [3]. Here, various counter measures have been developed, based on majority decision [4], iterative approaches [5], control and gold standard questions [6], [7], or by designing the tasks in a "cheat-proof" way.

Besides intentional cheating, issues caused by the task design can also result in low quality results [8]. As a direct interaction between workers and employers is usually not possible, identifying problems, e.g. misleading instructions, is hard to achieve. However, even if a direct supervision is impossible, the employer usually has an idea about how the interaction between the worker and the task interface should look like. This interaction pattern might include simple constrains, e.g., there is a minimum time it requires a user to read the instructions, or that the worker has to follow complex workflows if the task requires a sophisticated interface. Monitoring these interactions and comparing them with the expected ones defined during the task design can be used as an indicator if a worker is performing a task seriously and correctly.

In this work, we demonstrate how such user interactions can be analyzed using an application layer monitoring (ALM) approach and demonstrate its applicability to estimating a worker's performance. Modeling of user interactions is only applicable using the example of a specific task, as the monitoring framework and the expected behavior have to be tailor-made for each application. To this end, we use a simple language test as example for a Crowdsourcing task, which is described in Section II. In Section III, we detail on a possible implementation of an appropriate ALM for this task. This monitoring enables us to derive fine granular temporal information, about how much time the participants spend on specific parts of our test. Section IV discusses the results from the ALM monitoring and their interpretations in terms of worker behavior. Using these results, we show that it is possible to predict a worker's performance in Section V. Section VI concludes the paper.

II. Example Use Case: Language Test

To illustrate a possible implementation and benefits of ALM in a Crowdsourcing environment, we use an English language test as example task. Such a qualification test is not necessarily a common Crowdsourcing task, however English language comprehension is a very essential qualification on Crowdsourcing platforms. Khanna et al. [8] have shown that non-comprehension of instructions on Crowdsourcing platforms can be a severe issue, especially in countries with low educational standards, which possibly yields poor task results.

A. Test Design

The test consists of 5 texts with 5 multiple choice questions for each text, resulting in a total of 25 questions on one single web page. In order to increase the difficulty to share any solutions of the test, the order of the texts, the questions and answers was randomized. Additionally, one text production question was added at the end of the test, where the worker was asked to state which text he liked best and why. Workers were only able to complete the task, if all questions were answered. After these mandatory questions, the worker could leave optional feedback on a separate page.

The test texts were based on slightly modified articles from the Simple English version of Wikipedia articles. The texts' topics include science, celebrities, pop culture as well as recent history. Every text contained approximately 200 words. Although the topics were rather common, we made sure that the texts contained very specific information hardly any candidate could answer due to prior knowledge.

We constructed the questions according to Day et al. [9] who give detailed advise on how to design language comprehension tests. Two questions were aiming at *literal comprehension*, i.e. the required information could explicitly be found within the text. One question was aiming at *reorganization*, i.e. extracting and combining several pieces of explicit information from the text is necessary. The two remaining questions were aiming at *inference*, i.e. the required information is only implicitly stated in the text and needs to be inferred. The literal comprehension questions are rather easy to solve, because the answers are explicitly given in the text. In contrast the reorganization question are more difficult, as a deeper understanding of the text is required. The inference questions are assumed to be the most difficult question type for most of the workers, because abstract thinking is required here.

For each question, the worker is given four possible answers. In order to derive additional implicit feedback from the participants, we deploy a special answer scheme. Two of the answers can actually be found within the text, but only one of them makes sense regarding the questions and is correct. This can be used to distinguish between participants who have read the text, but did not understand the question. The other two answers sound possible regarding the question, but cannot be found within the text. These answers were intended to capture people who may very well have understood the question, but may have skipped the text to save time.

To derive the participants' score, we deployed a very simple scoring system that assigned 1 point per right answer and 0 points for each wrong answer of the multiple choice questions. The total score is then calculated as the sum over all points. Consequently, the lowest score that could be reached amounted to 0 points in total, while the highest score was 25 points. The text production question is not considered in this scoring system, because it is not possible to evaluate it an objectively. However, it can used as an indicator how serious a worker is taking the test, e.g., by considering the length of the answer.

The motivation of the test is to determine whether or not a candidate has the qualification to understand English task instructions. Although our scoring system allows for a graduated assessment of this qualification, the choice of whether or not a candidate will pass the test is a binary one. Therefore, we intended to determine a suitable *Qualification Threshold* that needs to be reached in order to pass the test.

Multiple choice tests tend to foster the cheat pattern of satisficing [10], where candidates simply try to fill out the form as quickly as possible. For the subsequent considerations, we assume that these people will pick answers in a uniform distributed fashion. However, due to the fact that each candidate will receive a uniform distributed random sequence of answers, we also covered people that would use a deterministic answering pattern, i.e., for instance always picking the first answer. We calculate the probability of a candidate randomly passing the test, which we also refer to as the probability of false positive qualification. The probability of randomly reaching k points can be modeled using a Binomial distribution, with $p = \frac{1}{4}$ as the probability of randomly selecting the correct answer and $n = 25$ questions. Thus, the probability of reaching $k = 25$ points by chance would amount to $P(X = 25) \approx 8.9 \cdot 10^{-16}$.

Our desired sample size was in the order of $n \approx 10^2$ up to $n \approx 10^3$ individuals. Therefore, we decided that a probability of false positive qualification of approximately 10^{-3} would suffice to make sure that (almost) none of our candidates would pass the test by chance. This probability can be reached for $k > 12$ yielding in $P(X > 12) \approx 1.7 \cdot 10^{-3}$. For the sake of simplicity, we will henceforth normalize the maximum score to 100 percent and set the qualification threshold at 50 percent.

B. Test results

For our study, we recruited 215 test candidates on the Microworkers platform[1] in February 2013. The payment for the test task amounted to 0.10 USD, which is comparable to that of similar studies [10].

As we are conducting a language test, we firstly have a closer look at the origin of the participants. Some demographical information about the users is available at the Microworkers page, amongst others the users' home country. This information cannot be changed after the registration at Microworkers and since a valid mailing address is required for payment it can be assumed to be correct for most of the workers. Our candidates came from 22 different nations, however, the ten most frequent countries make up about 90% of the participants. Most of the participants came form Bangladesh (41%), followed by Nepal (10%) and Sri Lanka (10%). Besides India and Pakistan in Asia, several workers from Eastern Europe participated. About 2% of the participants are native speakers, from the United Kingdom and the USA. This indicates the presence of a distinct interest in countries for that we would expect English language skills to be comparatively low, while English-speaking workers appear to be reluctant to prove their linguistic abilities in a test like this.

Figure 1 depicts the workers' test results as a complementary cumulative distribution function (CCDF), with the normalized score on the x-axis. The qualification threshold is included as a vertical dashed line within the plot. The curve starts with a score of 0.08 and a probability of 100%, indicating that no candidates had less than 2 correct answers. On the contrary, 18% of the candidates had achieved a maximum

[1] http://microworkers.com Accessed: Feb. 2014

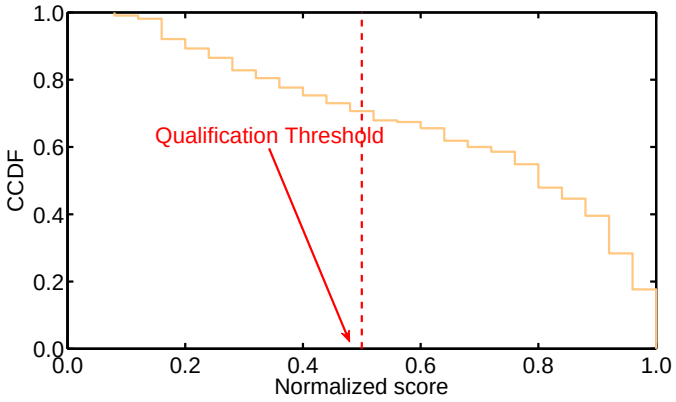

Figure 1. CCDF of the normalized scores from the test participants

score. The curve intersects with the qualification threshold at a probability of 71%.

In the following, we analyze the interactions of the users with our test and demonstrate how these interactions can give first insights about the expected scores of the particular user.

III. Application Layer Monitoring

In our approach, we assume that the user interface of the Crowdsourcing task is implemented as a web application. This enables us to add the monitoring using common web development techniques and guarantees the preservation of the worker's privacy, because we are only able to monitor his interactions with the tasks interface. This is similar to a regular workspace, where the supervisor can monitor the employees.

A. General Approach

Our approach gathers information about the user on both, client and server side. In general, server side measurements enable monitoring the accessed resources of the web application and the time of the request. In our use case, our application only consists of one single web page, therefore we use the server logs to analyze when and how often a users accessed our page. Furthermore, the time of the submission of the form can also be derived from the server side information.

However, more of the information about the users' working behavior can be derived from their interaction with the application interface itself. Using JavaScript and DOM application events, each interaction with HTML elements, such as buttons, text fields, etc. can be monitored with milliseconds precision. It is also possible to monitor a limited range of interactions with the browser itself, such as changing or closing the application window as well as switching to another browser tab. This allows us to study the user's interaction behavior on a very fine-grained interaction level including, leaving and entering the application, click behavior, mouse movement, scroll movement, manipulating interface elements and text inputs, and text selection.

B. Use Case Implementation

As mentioned before, ALM has to be implemented on a per task basis, i.e. the expected behavior of a user has to be known in advance to be able to identify suspicious user interactions. In order to model this expected work behavior, we considered the steps that were necessary to solve the test. Naturally, the user will start by reading the instructions at the top of the test. In order to either get to a text or to get to the questions, the user would have to scroll. He would then be reading a text or be engaged in answering questions, i.e. in finding and picking the right answer. To get to the next text passage of the test, the user would then be scrolling again and so forth. We were interested in the sequence and duration of these steps as well as the details of what the user is doing in them. In order to determine the periods in which the user would remain in a certain step, we relied on two measurements:

1) The user's vertical scroll position which was supposed to help us reconstruct the user's current field of sight (measured synchronously in intervals of 10 sec)
2) The user's interaction with the answering elements of the survey (measured asynchronously, event-based). This particularly included:
   - The clicks on radio buttons for multiple choice questions.
   - The selection and de-selection of the text box for the text production question.

Further, we consider how the users interact with the test while answering a particular question.

IV. Application Layer Measurements

Next, we review the potential of ALM metrics. We show that ALM can be used to analyze the work behavior we would expect from users. Therefore, we deploy several variables to determine different aspects of the work strategy. We will investigate the completion time, working phases, and consideration time.

A. Completion Time

Previous work has shown that the completion time of a task can be used as a indicator for quality of task results [11], [12]. Furthermore, by defining time thresholds, low performing workers can be detected [13], [14]. In order to derive such a threshold for our task, we assume an ideal worker with the following properties:

1) The ideal worker is familiar with speed reading techniques, which allow him to read $2000\,^{\text{words}}/_{\text{min}}$. Thus it would take $t_r = 1700\,\text{words}/2000\,^{\text{words}}/_{\text{min}} = 51\,\text{sec}$ to read all the text within the test.
2) The ideal worker is able to answer any question within $t_q = 5\,\text{sec}$.

As a result, the *Plausibility Threshold*, i.e., minimum completion time would yield in $t_{pt} = t_r + 25 \cdot t_q = 176\,\text{sec} \approx 3\,\text{min}$.

In our test, the completion time varied between 1.03 and 55.95 min with a median completion time of 20.13 min. We have observed that 7.6% of our participants have completion times below our plausibility threshold. In the following we analyze the coherence between the workers' scores and the task completion time, which is visualized as a scatter plot in Figure 2. The x-axis describes the completion time in minutes, while the y-axis denotes the score normalized to 100%. Each

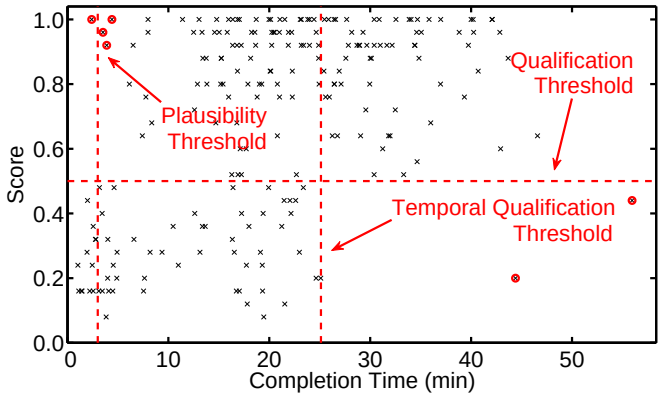

Figure 2. Scatter plot of the completion time and the test score per participant

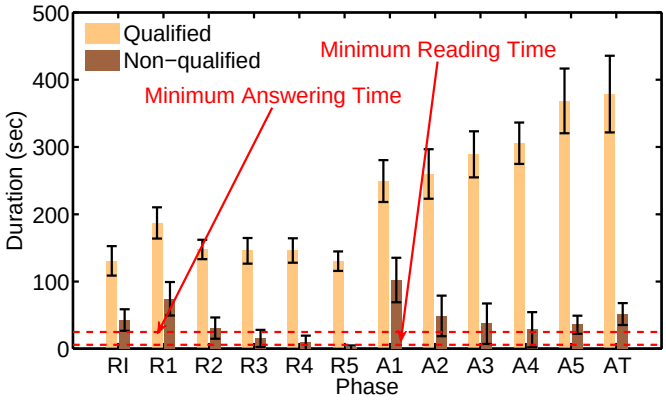

Figure 3. Average durations of the phases in the English language test

data point represents the performance of a single worker. The qualification threshold is included as a horizontal dashed line, the plausibility threshold as a vertical dashed line.

We observe that almost all people with a completion time below the plausibility threshold also drop below the qualification threshold. Nevertheless, our test sample includes four participants, who completed the test approximately within our expected minimal time and still achieved scores between 92% and 100%. A closer analysis of these users shows that all of them accessed the test at least half an hour prior to their submission. It is likely that the users copied the text to familiarize themselves with the test, respectively completed it offline in advance.

The plot also reveals that almost all users with a completion time above 25:07 min qualified in our test. Thus, this value could be regarded as the *Temporal Qualification Threshold* for our test. Nonetheless, we observe two outlier workers, one at 44:24 min with a score of 20% and another at 55:57 min with a score of 44%. Our analysis did not show any further abnormalities for these candidates, so we are not able to determine reasons for their low performance.

We conclude that for the completion time of a task, temporal thresholds are well-suited in order to give a first assessment of quality respective qualification in our case. The two temporal thresholds subdivide the plot into three horizontal segments. The plausibility threshold can predict the non-qualification of candidates in the first segment, i.e. below the threshold. The temporal qualification threshold on the other hand can be used in order to predict the qualification of workers with completion times in the third segment. However, none of the thresholds can make predictions for the second segment, which includes most of the workers and we see that the predictions tend to show a classification error in some cases. Moreover, the temporal qualification threshold can only be be estimated if the results of the test are already evaluated.

In this analysis, we only considered the duration of how long workers worked on the tests. What still remained unclear at this point is what the workers actually did within our test. In the following we shed light on this issue by investigating their low-level interactions with our application.

B. *Working Phases*

Instead of considering the time it takes the users to complete the whole test, we now consider the time he spends on reading and answering. While completing our test task, we assume the worker to be either reading the instructions or the texts, or answering a multiple choice question or the text question. To analyze the time the workers spend in these phases we use the following estimators:

(E1) "Reading Instructions" (RI) describes the time the instructions were visible to the user, i.e. the time the user had the chance to read them. The estimator "Reading Text x" (Rx) works in a similar fashion for each of the texts.

(E2) "Answering the Questions for Text x" (Ax) calculates the time difference between the timestamp of the first time the user could have possibly seen the questions about text x and timestamp of the last answer given for these questions. The estimator "Answering the Text Question" (AT) works in a similar manner, but uses the timestamp of the last de-activation of the text box as end time.

Monitoring the workers' interaction with radio buttons or the text field is rather easy using JavaScript. Here, every change triggers an event which can be monitored. However, large parts of our test include reading texts. During this time, the worker does not explicitly interact with the web page. In order to analyze these reading phases, we estimate the currently visible area. This information can be retrieved using the current scroll position in conjunction with the browser window height, which can be determined using JavaScript. A non-responsive CSS layout which ensures a fixed size of the web page independent of the workers' device resolution enables us to recalculate the position of the visible elements of the web page at any point in time. However, most of the time the user has the chance to see several elements belonging to different phases. Therefore, the aforementioned estimators are constructed in a fashion which allows the different phases to overlap.

Figure 3 visualizes the average time the workers spend in the different phases, including the 95% confidence intervals. Note that the y-axis denotes the absolute duration in seconds. The nomenclature of the phases follows the one introduced for the estimators, whereas the numbers indicate the text or questions position. $R3$ for example refers to the time, the

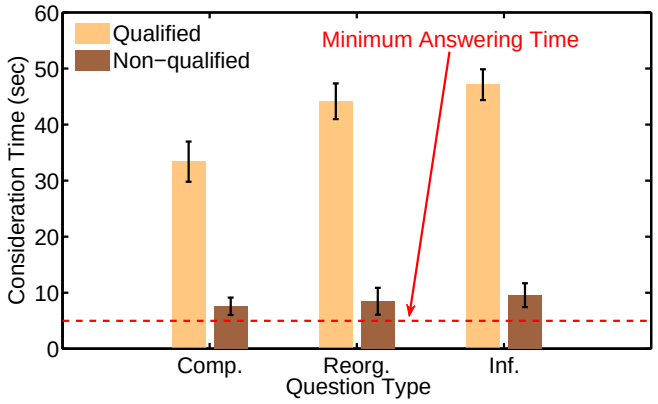

Figure 4. Consideration times per question type

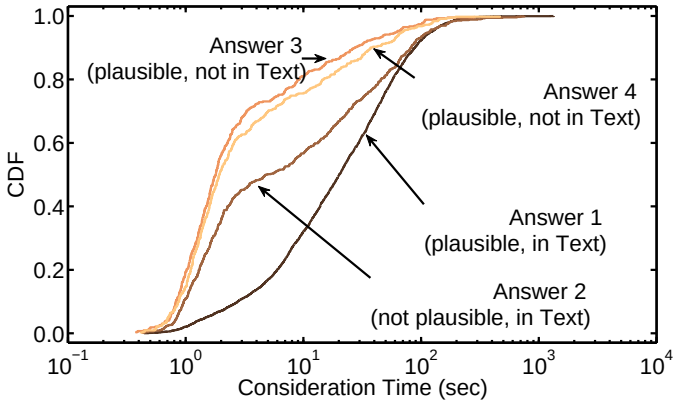

Figure 5. Consideration times per answer type

workers spend on reading the third text. We also included two threshold values visualized as two red dashed lines. The estimated minimum reading time of 6 sec and the minimum answering time of 25 sec. Note, that the minimum answering time refers here to the time it takes to completion of all 5 multiple choice questions per text.

We can observe that none of the qualified workers drops below one of the thresholds, but spends on average between 130 and 187 sec on reading a text and even more time on answering the questions. For our experiments, we see a very surprising tendency for the answering behavior of qualified users, as the duration of the answering phases increases with each text. This would indicate that they would work more diligently towards the end of the test. Note however, that the confidence intervals for all answering phases overlap, except for the first and the fifth phase indicating that this behavior need not be typical for all of the users. The average phase time of the text production question is even higher than the answering phase times for the questions.

In contrast, non-qualified workers tend to fall below the thresholds. Moreover, we can observe that both reading and answering phases tend to decrease with each text. Interestingly, however, the duration of the reading and the answering phase for Text 1 is factually higher than the corresponding duration for Text 5. This indicates that these users' motivation might have dropped during the test. Surprisingly, they tend to spend more time on the text question than on regular question blocks. This might suggest that the candidates assume that this question would have a larger impact on their chance of passing the test compared to the remainder of the test.

Analyzing the working phases of the participants allows us a much stricter distinction between qualified and non-qualified workers than the analysis of the task completion time. Using the same approach of temporal thresholds but on a finer granularity, we can clearly distinguish between workers working diligent and workers only trying to complete the test as fast as possible.

We have already analyzed the overall completion time of our test. In the analysis of the working phases, we considered only the time the workers spend on answering all five questions related to one text. In the next paragraph, we have a closer look at the answering process of the individual questions and analyze which information we can derive from this information.

C. Consideration Time

For the analysis how much time the test participants spend on the single questions, we use the variable *Consideration Time*, which is the time between the first time a user saw a question and the time the user changed his answer for this question for the last time.

Figure 4 visualizes the mean consideration times including the 95% confidence intervals for the different questions types described in Section II-A. The red dashed line indicates the 5 sec threshold for the estimated minimum answering time.

For the qualified workers, we observe the consideration times are above the expected threshold. Furthermore, there is a tendency that questions with a higher level of difficulty cause higher consideration times. This is also intuitive, as simple questions based on literal comprehension require less effort than reorganization or inference questions, therefore they can be answered more quickly. In contrast to this, the difficulty of the questions do not have a significant impact on the consideration times of the non-qualified workers. Even if the mean consideration times for the questions are also higher than the minimum answering time, it is clearly visible that the non-qualified workers spend significantly less time on answering the questions than the qualified workers.

Next, we examine the consideration time with regards to different answer types which are depicted as a CDF in Figure 5. The x-axis denotes the consideration time on a logarithmic scale in seconds. The curve indicating Answer 1 shows the consideration times for questions that were answered *correctly*, while the remaining curves are incorrect answers. We can observe the overall tendency that picking the correct answer requires significantly more consideration time than picking a *wrong* one. Interestingly, Answer 2 appears to take up more consideration time than Answers 3 or 4. This is what we would have expected as this answer can still be found within the text. Thus, people who did not understand the question would at least try to look for this answer within the text. It also becomes apparent that the Answers 3 and 4 indeed tend to capture "lazy" people that may simply skip the text.

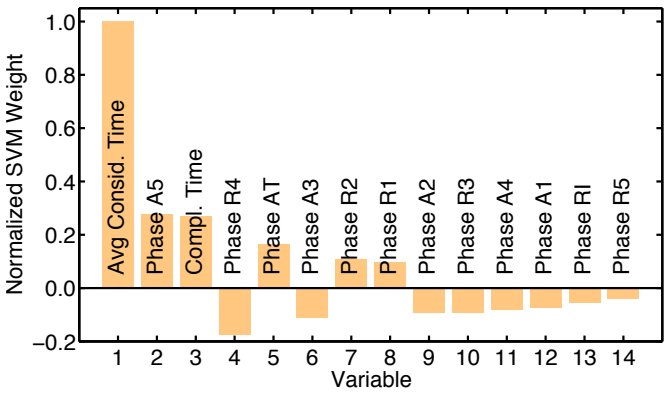

Figure 6. SVM weights indicate importance of features for quality prediction

The results from the consideration times indicate that even on a very low level of interactions, worker behavior can be monitored and suspicious behavior can be detected. In the next section, we analyze to which extent the gathered information about the worker behavior can be used to predict a workers test score.

V. ALM for Estimating Worker Qualification

For the evaluation of our test, we used a qualification threshold to assign each test participant to the categories *qualified* or *non-qualified*. We now want to analyze if it is possible to predict the assigned category solely by using the ALM measurements. In order to achieve this, we use supervised machine learning to train a Support Vector Machine (SVM) using the features, *task completion time*, the different *phase times*, and the *average consideration time*. Furthermore, we use cross-validation to avoid over-fitting due to our relatively small sample size of training data.

Figure 6 shows the different features of the SVM on the x-axis and their weights, normalized by the maximum value, on the y-axis. We can observe that the average consideration time is the most important feature in the classification process. It is likely that the time the participants spend on finding the right answers is a good indicator for the workers diligence. More surprisingly, also the time the worker spends on the answering the last question is a good indicator for the overall quality. This behavior might indicate that the worker works diligently even at the end of the test and implies a good work quality throughout the whole task. As already shown in previous studies, the overall completion time also offers a first indicator on the result quality.

The overall accuracy of the SVM amounts to 88.67% meaning that in 88.67% of all cases its predictions are correct. The class precision for qualified candidates amounts to 93.06%. In contrast, this means that we predict that 10 candidates would be qualified that are truly non-qualified. For 18 workers, we predict that they are non-qualified, although in reality they were indeed qualified.

VI. Conclusion

In this work, we presented an implementation of an Application Layer Monitoring approach for Crowdsoucing tasks and demonstrated that the obtained information can be used to predict the quality of the workers' results. ALM is based on the idea of monitoring a worker's behavior during task completion and comparing it to the expected behavior. We used an English language test as concrete example, however, ALM can also be applied to other Crowdsourcing tasks, if the behavior model and the resulting estimators are adapted.

Using the results of the ALM, we analyzed the test participants' interactions with the task interface. We demonstrated that the interactions can be monitored at different levels of detail and that the observations can be mapped to expected user behavior patterns. Furthermore, we showed that these different user behaviors have an impact on the user's test result. Using this information, a machine learning approach was used to train an automatic classifier to predict the user's test result solely on the ALM data.

Acknowledgement

This work is supported by the Deutsche Forschungsgemeinschaft (DFG) under Grants HO4770/2-1 and TR257/38-1. The authors alone are responsible for the content.

References

[1] P. Tran-Gia, T. Hoßfeld, M. Hartmann, and M. Hirth, "Crowdsourcing and its impact on future Internet usage," *it - Information Technology*, vol. 55, Jul. 2013.

[2] T. Hoßfeld, M. Hirth, and P. Tran-Gia, "Modeling of crowdsourcing platforms and granularity of work organization in future Internet," in *International Teletraffic Congress*, San Francisco, USA, Sep. 2011.

[3] S. Suri, D. G. Goldstein, and W. A. Mason, "Honesty in an online labor market," in *Human Computation Workshop*, San Francisco, USA, Aug. 2011.

[4] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? Improving data quality and data mining using multiple, noisy labelers," in *Conference on Knowledge Discovery and Data Mining*, Las Vegas, USA, Aug. 2008.

[5] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller, "Exploring iterative and parallel human computation processes," in *Workshop on Human Computation*, Washington DC, USA, Jul. 2010.

[6] D. Zhu and B. Carterette, "An analysis of assessor behavior in crowdsourced preference judgments," in *Workshop on Crowdsourcing for Search Evaluation*, Geneva, CH, Jul. 2010.

[7] D. Oleson, A. Sorokin, G. P. Laughlin, V. Hester, J. Le, and L. Biewald, "Programmatic gold: Targeted and scalable quality assurance in crowdsourcing," in *Workshop on Human computation*, San Francisco, USA, Aug. 2011.

[8] S. Khanna, A. Ratan, J. Davis, and W. Thies, "Evaluating and improving the usability of mechanical turk for low-income workers in India," in *Symposium on Computing for Development*, London, UK, Dec. 2010.

[9] R. R. Day and J.-s. Park, "Developing reading comprehension questions," *Reading in a foreign language*, vol. 17, 2005.

[10] A. Kapelner and D. Chandler, "Preventing satisficing in online surveys," in *CrowdConf 2010*, San Francisco, USA, Oct. 2010.

[11] F. Ribeiro, D. Florêncio, C. Zhang, and M. Seltzer, "CrowdMOS: An approach for crowdsourcing mean opinion score studies," in *Conference on Acoustics, Speech and Signal Processing*, Prague, CZ, May 2011.

[12] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, "Quantification of YouTube QoE via crowdsourcing," in *Symposium on Multimedia*, Dana Point, USA, Dec. 2011.

[13] W. Mason and S. Suri, "Conducting behavioral research on Amazon's Mechanical Turk," *Behavior research methods*, vol. 44, 2012.

[14] D. E. Difallah, G. Demartini, and P. Cudré-Mauroux, "Mechanical cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms," in *CrowdSearch*, Lyon, FR, Apr. 2012.