Summer 10-6-2011

EFFICIENT QUALITY MANAGEMENT OF HUMAN-BASED ELECTRONIC SERVICES LEVERAGING GROUP DECISION MAKING

Robert Kern

Hans Thies

Gerhard Satzger

EFFICIENT QUALITY MANAGEMENT OF HUMAN-BASED ELECTRONIC SERVICES LEVERAGING GROUP DECISION MAKING

Kern, Robert, Karlsruhe Institute of Technology (KIT), Karlsruhe Service Research Institute (KSRI), Englerstrasse 11, Building 11.40, D-76131 Karlsruhe, Germany, robert.kern@kit.edu

Thies, Hans, SAP Research, Blumenbergplatz 9, CH-9000 St. Gallen, Switzerland, hans.thies@sap.com

Satzger, Gerhard, Karlsruhe Institute of Technology (KIT), Karlsruhe Service Research Institute (KSRI), Englerstrasse 11, Building 11.40, D-76131 Karlsruhe, Germany, gerhard.satzger@kit.edu

Abstract

Human-based electronic services (people services) provide a powerful way of outsourcing tasks to a large crowd of remote workers over the Internet. Because of the limited control over the workforce in a potentially globally distributed environment, efficient quality management mechanisms are a prerequisite for successful implementation of the people service concept in a business context. Research has shown that multiple redundant results delivered by different workers can be aggregated in order to achieve a reliable result. However, existing implementations of this approach are highly inefficient as they multiply the effort for task execution and are not able to guarantee a certain quality level. Our weighted majority vote (WMV) approach addresses this issue by dynamically adjusting the level of redundancy depending on the historical error rates of the involved workers and the level of agreement among them. A practical evaluation in an OCR scenario demonstrates that the approach is capable of gaining reliable results at significantly lower costs compared to existing procedures.

Keywords: crowdsourcing, group decision making, quality of service, human-based electronic services.

1 Introduction

Human-based electronic services are a specific form of *crowdsourcing* - a term shaped by Howe (2006). While they formally look like Web services, human-based electronic services are not performed by a computer, but rather by human workforce, typically by a crowd of Internet users. The success of Amazon's Mechanical Turk (MTurk) platform[1] and the growing number of companies that build their business model entirely on that platform demonstrate the potential of this approach (Frei 2009). Here, the platform acts as a broker between requesters who publish micro tasks and workers who perform those tasks in return for a small compensation.

Kern et al. (2009) coined the term *people services* (pServices) for human-based electronic services and define it as "Web based software services that deliver human intelligence, perception, or action to customers as massively scalable resources". As there is limited control over the individual contributors, particular attention has to be paid to the quality of the work results – in fact, quality management is the key to a widespread application of pServices in more business critical scenarios than the ones currently supported. However, a central validation of work results by the requester conflicts with the general idea of people services to provide human workforce in virtually unlimited scalability: if the requester doesn't have the bandwidth for performing the actual work it can be typically assumed that he would neither have the bandwidth for reviewing the result quality in a scalable way. Therefore, it seems obvious that the scalability of the crowd needs to be leveraged not only for performing the actual work, but also for ensuring the quality of the results.

For annotation tasks, Sorokin and Forsyth (2008) distinguish between two generic quality assurance strategies on pServices platforms which use the crowd itself for quality assurance: the collection of multiple annotations and the performing of a separate review task (they call it grading task). More generally, the first approach can be defined as a *majority vote* approach which "introduces redundancy by passing the same task to multiple workers and aggregating the results in order to compute the result with the highest probability for correctness" (Kern et al. 2010).

Existing applications of this approach typically use a fixed degree of redundancy, i.e. each task is performed by the same number of workers. The result with the highest probability for correctness is determined based on the agreement among the workers. Obviously, the level of agreement (and so the expected result quality) varies depending on the error rates of the involved workers. For some tasks, the agreement might be extremely high (e.g. all workers agree on exactly the same result), for others the worker results might be at odds (e.g. half of the workers returns result A, while the other half returns B). Therefore, a majority vote with a fixed degree of redundancy is both highly inefficient and incapable of assuring a certain level of result quality.

In order to overcome these problems, our *weighted majority vote* (WMV) approach dynamically adjusts the required degree of redundancy by taking into account both the actual level of agreement between the workers as well as their historical error rates. After providing an overview on related work, we introduce a model of the WMV. Thereafter, we evaluate its capabilities in an optical character recognition (OCR) scenario implemented on Amazon's Mechanical Turk platform. The paper closes with a summary and an outlook.

2 Related Work

The general use of group decisions has been discussed for a long time. The validity of the majority vote model, e.g. as traditionally adopted by the judiciary, has been mathematically proven by Condorcet's Jury Theorem (Condorcet and Caritat 1785). Under the assumption that one of two outcomes is correct and each decision maker has the independent probability $p > 0.5$ to make the right

[1] www.mturk.com

decision, the probability for a correct group decision is greater than the individual one. Furthermore, the probability of a correct decision increases with the number of individuals in the group. May (1952) examines the fairness of the majority rule under the assumption that every voter has exactly one vote. Lam and Suen (1997) study the application of a majority vote mechanism for the aggregation of results in Pattern Recognition. According to them, the term *majority* usually refers to more than 50 percent of a group.

Surowiecki (2004) illustrated that the aggregation of group responses may lead to better results than the information of any single group member - if the opinions are diverse, independent, decentralized, and an appropriate aggregation mechanism exists. This phenomenon has been described as the *wisdom of the crowds*. Typical applications that leverage crowd intelligence are prediction markets (Gruca et al. 2005), Delphi methods (Rowe and Wright 1999) and extensions of the traditional opinion poll.

In the field of machine learning, Littlestone and Warmuth (1994) developed a weighted majority algorithm that acts as a "master algorithm" and aggregates the answers of several prediction algorithms in order to determine the best prediction possible.

The aggregation mechanism is a vital part of each majority vote model. Depending on the task structure, results can be aggregated in several ways. Xu et al. (1992) suggest to combine multiple classifiers for handwriting recognition, Revow et al. (1996) compare five combination strategies (majority vote, Bayesian, logistic regression, fuzzy integral, and neural network) and arrive at the conclusion that majority vote is as effective as the other, more complicated schemes to improve the recognition rate for the data set used.

The majority vote concept is widely used in the context of people services. Redundant task execution is a basic feature for quality improvement provided by platforms like MTurk. Sorokin and Forsyth (2008) and Snow et al. (2008) have analyzed the effect of the approach based on annotation scenarios. Snow et al. (2008) have investigated how many non-experts out of the crowd are needed in order to achieve better results than one single expert. They consider a basic majority vote mechanism, randomly breaking ties at an equal number of identical answers. They further use the workers' past error rates for bias correction of new votes. They arrive at the conclusion that for many tasks only a small number of non-expert annotators are necessary to outperform a single expert annotator. Depending on the scenario, they report a required number of non-experts between two and more than ten. They also emphasize the high speed and low costs of Mechanical Turk. Whitehill et al. (2009) consider how to integrate labeler's expertise into a majority vote mechanism for image labeling. They propose a probabilistic model and use it to simultaneously infer the label of each image, the expertise of each labeler, and difficulty of the individual image recognition task. As stated by Surowiecki, the aggregation of individual votes is essential to determine a common decision. Only tasks that can be aggregated automatically are suitable for the basic majority vote approach. Kern et al. (2010) refer to this type of task as *deterministic task*. They define the term deterministic task as "a task for which there is a well defined optimal result i.e. for which two workers who perfectly meet the task objective will pass exactly the same results, or for which the responses can at least be automatically transformed (normalized) into a well defined optimal result."

3 The Weighted Majority Vote Approach

In this chapter we introduce the weighted majority vote (WMV) approach which allows for efficient and goal based quality management for pServices. After introducing the overall scenario and the assumptions we describe the approach in detail.

3.1 Scenario and assumptions

The basic scenario of the WMV approach comprises three roles: a requester who wants to offer work, a group of workers who want to work on those tasks and a Web platform that coordinates the process and manages the quality of the work results.

The requester passes tasks to the platform which makes it available to the workers in a task queue. The workers pick the tasks on which they want to work and submit raw results back to the platform. The scenario assumes that a raw result is either correct or not, i.e. it either meets the quality requirements of the requestor or does not. The requester should design and describe the task in a way that allows for a sharp distinction between correct and incorrect results. For each specific type of task, an individual failure rate p_x is attached to each worker x which represents the likelihood of the raw result meeting the quality requirements of the requester. The error rate may change over time, since workers may become careless by the time or they might improve their skills. The term *raw* result indicates that the result is not necessarily correct.

The difficulty of all tasks is assumed to be similar. However, there might be some individual tasks that are harder to solve than others, for example because the task description does not apply to all individual tasks in the same way. As those tasks may be impossible to solve without support of the service requester, they are being sent back to the requester so that he can use this information to improve the task design and resubmit the tasks.

3.2 Model

The goal of the weighted majority vote approach is to dynamically determine for each task how many workers are required to meet the requester's quality requirements (1). The basic idea is to start with a raw result returned by a single worker (2) and calculate based on his individual error rate whether the required minimum probability for correctness has already been met (3). If this is the case, the final result is returned (4). If the required quality has not yet been met, it is checked in step (5) whether a quality improvement can be expected by adding more workers. If that is not the case, the task is escalated back to the requester. Otherwise, the process continues with step (2) where another raw result is gathered from an additional worker. The process is continued until either the raw results delivered by several workers can be aggregated into a reliable result (3) which is returned as the final result to the requester (4) or until the escalation limit is reached in step (5). Figure 1 provides a schematic overview of the scenario.

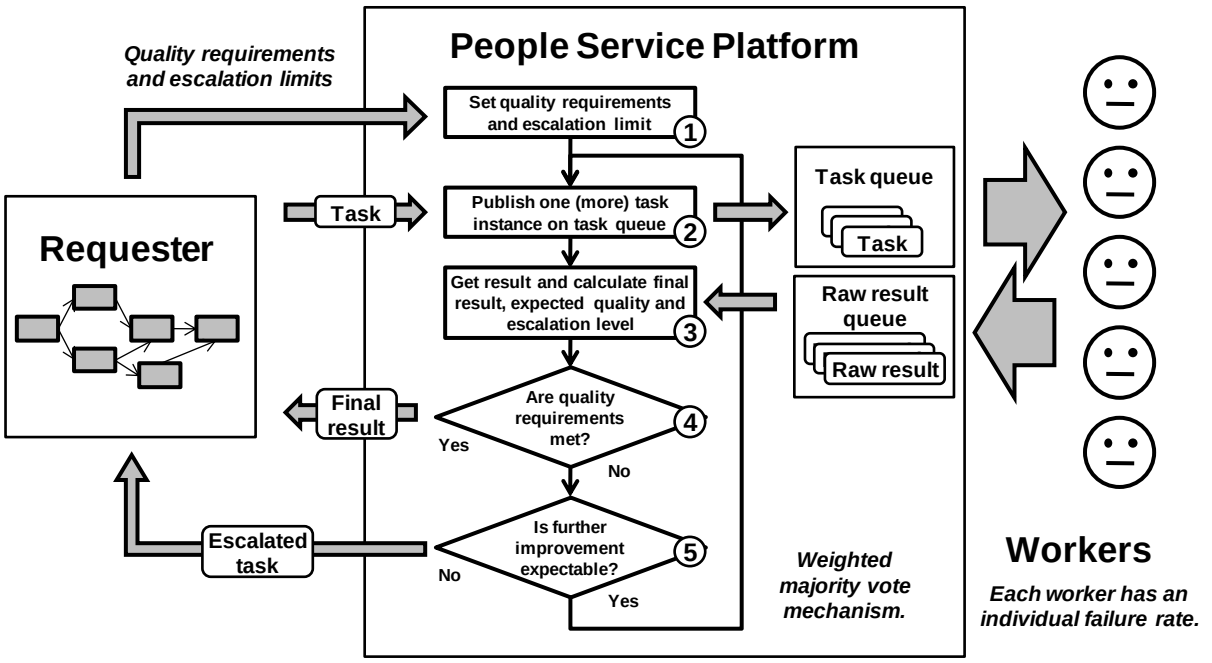

Figure 1. Weighted majority vote (WMV) scenario

We presume that the correctness of the results follows a certain statistical distribution, which can be leveraged in order to aggregate the results. We also assume a fixed payment per task; therefore the overall costs can be minimized by minimizing the total number of tasks.

To summarize, the following parameters are introduced by the scenario:

The requester specifies

- **Desired level of quality φ_{min}** - the desired level of result quality, i.e. the minimum probability that the results returned by the weighted majority vote meet the requirements of the requester.
- **Result distribution D** - the statistical distribution that describes the correctness of the results. It should be tested via statistical hypothesis testing. In this paper, the binomial and the Poisson distribution are used as examples.
- **Maximum escalation level ε_{max}** - the maximum probability that a task gets escalated to the requester. If $\varepsilon_{max} = 0.05$, a maximum percentage of 5% of all tasks might get escalated in the long run even if all of them conform to the task specifications.

The workers x

- return **raw results r_y^x** when working on task y.
- have **individual failure rates p_x** for the given type of task.

From a mathematical point of view, the weighted majority vote approach is consisting of the following steps illustrated by Figure 2:

1. Specify the desired level of quality φ_{min} and the escalation level ε_{max}.
2. Make one (more) instance of task y available to the workers.
3. Retrieve the raw result r_y^x of worker x having worked on task y**.** Among the results already submitted for task y, identify the one with the highest probability of correctness φ_c as well as the escalation level ε_y according to the given result distribution D.
4. Return the result r_c with the highest probability of correctness φ_c in case the probability φ_c exceeds the desired level of quality φ_{min} and update the qualification values q_x of all participating workers, where $q_x = 1 - p_x$.
5. Escalate the task back to the requestor if the overall probability ε_y for getting the result tuple R_y is lower than or equal to the escalation limit ε_{max}, with $R_y = \left(r_y^{x_1}, r_y^{x_2}, .. r_y^{x_N}\right)$ and $x_1 = 1, x_2 = 2.. x_N = N$ being the IDs and N being the number of the workers who have worked on the task.

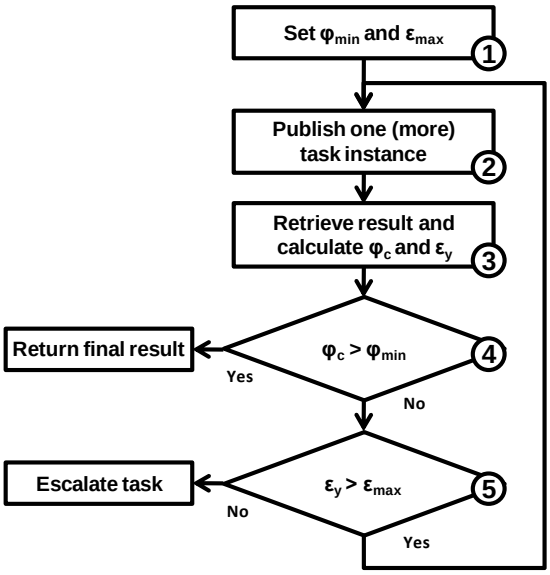

Figure 2. Weighted majority vote (WMV) approach

Steps 2 to 5 are repeated until the final result is returned in step 4 or the task is escalated in step 5. In step 3, the values φ_c and ε_y are calculated using equations (E1) and (E2). Equation (E1) determines the Bayes-conditional likelihood for result r_c being correct under the condition that the result tuple R_y was received. What matters are not the actual values of the results $r_y^{x_1}, r_y^{x_2}, .. r_y^{x_N}$ but the agreement among the workers about a specific result being correct i.e. the number of workers who agree that a specific result is correct in combination with their individual error rates. The parameter k refers to the number of distinct results in the result tuple R_y.

(E1) $$\varphi_c = P(r_c\ correct \mid R = R_y) = \frac{P(r_c\ correct \cap R=R_y)}{P(R=R_y)}$$

(E2) determines the probability ε_y of receiving result tuple R_y. If ε_y reaches or falls below ε_{max}, the task is escalated back to the requestor. Therefore, ε_{max} denotes the approximate portion of tasks being escalated:

(E2) $$\varepsilon_y = P(R = R_y)$$

3.2.1 Example for Binomial Distribution

Assuming that the results follow a binomial distribution, the conditional probability φ_c that result r_c is correct when receiving the result tuple R_y is

(E3) $$\varphi_c = P(r_c\ correct \mid R = R_y) = \frac{P(r_c\ correct \cap R=R_y)}{P(R=R_y)} = \frac{\prod_{\forall r_i=r_c} q_i \prod_{\forall r_i\neq r_c} p_i}{\left(\sum_{j=1}^{k} \prod_{\forall r_i=r_j} q_i \prod_{\forall r_i\neq r_j} p_i\right) + \prod_{j=1}^{N} p_j}.$$

The probability ε_y of receiving result tuple R_y is calculated by:

(E4) $$\varepsilon_y = P(R = R_y) = \left(\sum_{j=1}^{k} \prod_{\forall r_i=r_j} q_i \prod_{\forall r_i\neq r_j} p_i\right) + \prod_{j=1}^{N} p_j$$

3.2.2 Example for Poisson Distribution

In the more complicated case of a Poisson distribution, let d_{ij} denote the distance between two results i and j. The distance to the gold standard is Poisson distributed with individual parameter λ_y. The value p(d_{ij}) denotes the probability that result i has the distance d_{ij} to the gold standard. Then the conditional probability φ_c that result r_c is correct is calculated by:

(E5) $$\varphi_c = P(r_c\ correct \mid R = R_y) = \frac{\prod_{i=1}^{N} p(d_{ci})}{\sum_{j=1}^{k} \prod_{i=1}^{N} p(d_{ji}) + \sum_{r=1}^{\infty} \sum_{j=1}^{k} \prod_{i=1}^{N} p(d_{ji}+r)}$$

The probability ε_y of receiving result tuple R_y is calculated by:

(E6) $$\varepsilon_y = P(R = R_y) = \sum_{j=1}^{k} \prod_{i=1}^{N} p(d_{ji}) + \sum_{r=1}^{\infty} \sum_{j=1}^{k} \prod_{i=1}^{N} p(d_{ji} + r)$$

4 Evaluation

In order to test the capability of the weighted majority vote approach, we implemented and evaluated it based on the existing marketplace Amazon Mechanical Turk. We used an optical character recognition (OCR) scenario that represents a typical task class applied on that platform.

4.1 Experimental design

Data revision, including scenarios such as classifying, tagging, summarizing and revising content or audio and video transcription, and the recognition of (hand) written texts is a class of scenarios which is currently widely used on Mechanical Turk. Like many data revision tasks, optical character

recognition (OCR) of handwritten texts cannot be fully automated yet (Lopresti 2009). Even sophisticated technologies need human assistance in order to achieve satisfying results.

The data used for the evaluation consists of 1,176 single handwritten words. Each task was specified by its particular input data consisting of an image file (JPEG) displaying the handwritten word. The MTurk platform was accessed through the Java SOAP API provided by Amazon.

In order to compare the WMV approach with the traditional majority vote mechanism, the actual evaluation was performed as a simulation on the basis of raw results from two batches, each consisting of multiple redundant instances per task. For simulating the traditional majority vote, a fixed number of raw results were used. For simulating the WMV, a varying number of raw results were used according to the dynamic concept of the approach (Figure 2).

The batches were uploaded to the Mechanical Turk twice: On January 8th, 2010 an initial set of 3 redundant instances of each of the 1176 task was uploaded without a qualification requirement. This first batch was used for parameter calibration. On February 1st, 10 additional instances of each task were uploaded. To prevent fraudulent workers and workers that are not qualified for this task type from participating, a qualification test was introduced. In the qualification test, the workers had to return the digital representation of 10 handwritten words. The actual evaluation used the second batch of 11.760 tasks. It was prohibited that a worker handles the exact same task more than once.

The task payment was \$0.01 per task, with Amazon receiving a service charge of \$0.005 for each task. Consequently a total amount of $1{,}176 \times 3 + 1{,}176 \times 10 = 15{,}288$ data sets has been collected during the evaluation leading to total expenses of $15{,}288 \times (\$0.01 + \$0.005) = \$229.32$.

4.2 Results

The result section consists of two parts. The first part covers the analysis of the raw results from the two batches and comprises the quality of the raw results, the agreement among the workers, the scalability of the MTurk platform and some considerations about the task specific result distribution. The second part covers the performance of the WMV approach.

4.2.1 Quality of raw results

We clearly observed the improvement of quality and the impact of fewer workers participating when introducing a qualification test. 112 workers participated in processing the first batch of $1{,}176 \times 3 = 3{,}528$ assignments, while only 38 did in the second batch, although it was significantly larger, comprising a total of $1{,}176 \times 10 = 11{,}760$ assignments. Of course, the workers in the second batch completed a significantly larger number of assignments each.

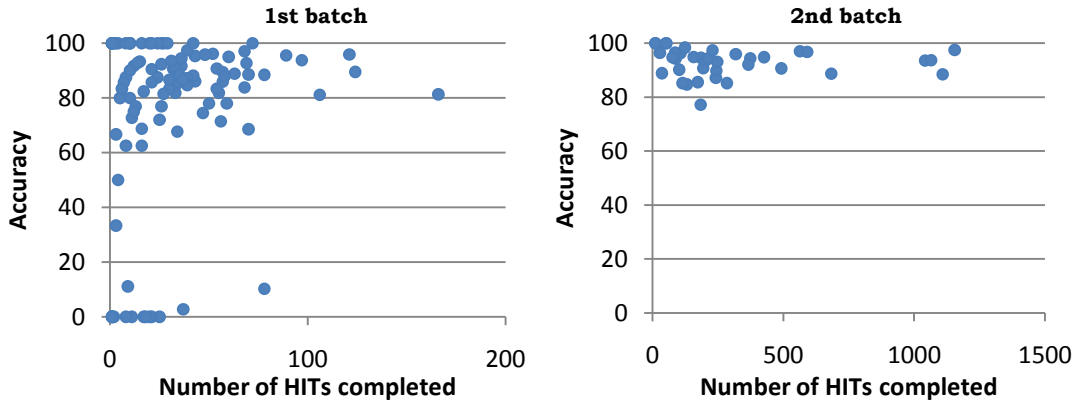

Figure 3. Worker accuracy and number of completed Human Intelligence Tasks (HITs)

The analysis in Figure 3 shows that there are workers with a wide spectrum of individual error rates. Some of them are obviously unable to provide good quality or try to cheat by submitting poor quality on purpose. As the 2nd batch results show, these are eliminated by introducing a qualification test.

To calculate the distance d_{ij} between two answers, we use the Levenshtein distance - a metric for measuring the amount of difference between two character sequences (Damerau 1964). It is defined as the minimum number of edits needed to transform one word into the other. Permissible operations are insertion, deletion and substitution of a single character. We assume the Levenshtein distance d_w^l to the correct answer to be Poisson distributed with parameter λ. The parameter λ is assumed not to change rapidly for the same worker and tasks of the same type. As validated with a χ^2-Test based on the results of the second batches, the hypotheses can be accepted with a significance of 95%.

4.2.2 Worker agreement

Figure 4 illustrates the level of agreement A_y amongst the workers for all HITs y during the second batch. In equation E7, we compute the agreement by dividing the sum of the Levenshtein distance d_{ij}^l of answer i to all other answers j for all assignments by the number of all possible combinations of votes:

(E7) $$A_y = \max\left(1 - \frac{\sum_{i=1}^{M}\sum_{j=1;j\neq i}^{M} d_{ij}^l}{(M-1)M\ z_y}\ ;0\right)$$

M denotes the number of votes participating (which equals 10 for all HITs of the second batch) and z_y denotes the average word length for task y. About 74% of all HITs have an agreement of 95% or better. Less than 5% have an agreement of below 70%. These mainly involve cases where workers submitted empty answers due to not being able to provide the right answer or because of carelessness.

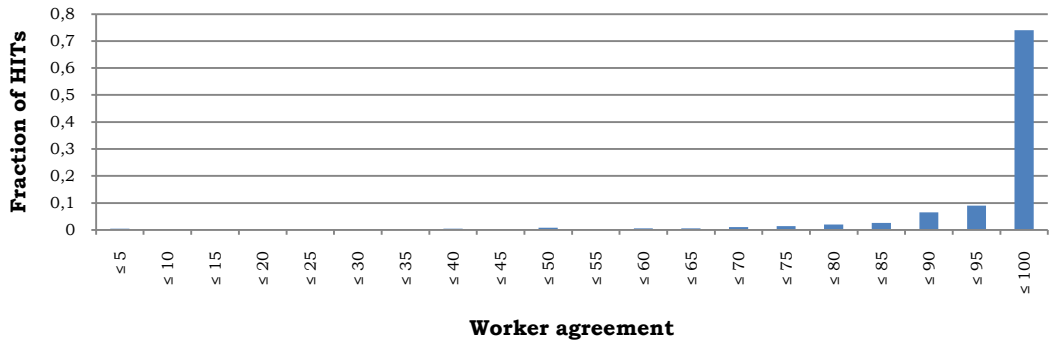

Figure 4: Worker agreement during second batch

We in fact see that the error rate does not seem to be equal for all tasks; especially those ~5% of HITs with an agreement of less than 70% are critical to the assumption of equal task difficulty. The goal is to escalate these tasks in order to provide feedback to the service requester so that he learns which tasks are problematic and what to consider when uploading tasks to the task marketplace in general.

4.2.3 Scalability

Probably the most astonishing result of the experiment was the speed at which the results were submitted. The first result in the first batch of 3,528 tasks was submitted after 22 seconds and the whole batch was handled (except 2 results that were already accepted by workers) in less than 15 minutes at an execution rate of 14,088 tasks per hour or ~126 tasks per worker and hour. During first experiments we even observed total execution speeds up to 3 times as fast, because of more workers participating. We assume that the execution speed besides the payment also depends on the time of day, since most workers are U.S. or Indian citizens (Ross et al. 2010).

Figure 5 illustrates the execution of the whole batch of 11,760 assignments which was executed at a slower execution speed due to the introduction of a qualification test. A similar chart is used by the crowdsourcing service provider CrowdFlower[2].

The first result was submitted after only 59 seconds, with the whole batch being processed after 2:40 hours at an execution rate of 4,410 tasks per hour or ~116 tasks per worker and hour. We also see that workers process tasks more continuously since the ratio of workers participating in assignments per HIT is a lot smaller than in the first case (~3.8 compared to ~37.3). Only few workers leave the process; some even have breaks and return to the process later (e.g. worker No. 16). Additionally we observe very different execution rates for different workers, from 36 assignments per hour up to 900 assignments per hour.

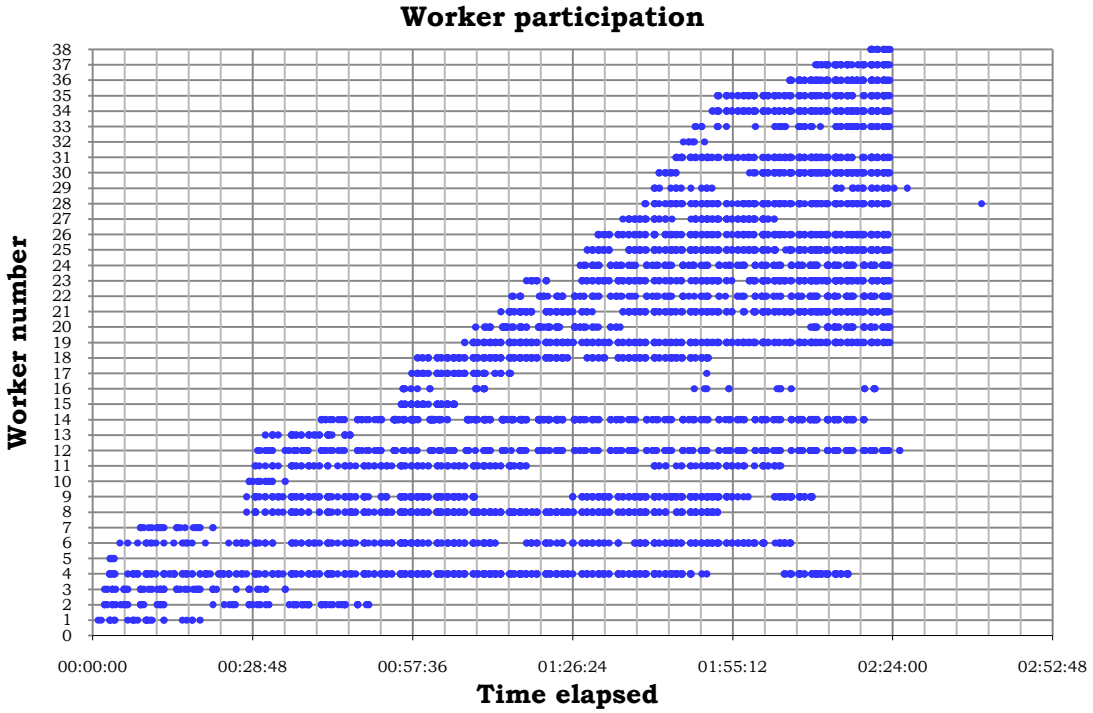

Figure 5. Timeline of HIT execution for the second batch

4.2.4 Impact of a task specific result distribution

As already pointed out before, the weighted (and also the basic) majority vote mechanism can be applied using a binomial distribution or a task specific distribution. In our OCR scenario it is the Poisson distribution. While the Binomial distribution only differentiates between correct results and incorrect results, the Poisson distribution can reflect the proximity between similar responses. For example, if only one character in a word was recognized incorrectly, the result is incorrect but still better than if the complete word was incorrect.

We found that we can use the Poisson distribution in order to approximate the Levenshtein distance of a worker's answer to the gold standard. Using the ordinary majority vote, the answer that occurs most is considered to be the most probable correct result. If several words occur the same amount of times ("tie"), a random choice between these answers is made, as suggested by Snow et al. (2008). In the weighted majority vote mechanism, the qualification values of the participating workers are considered. In this section we try to take a look at how the majority vote mechanism performs without including the worker error rate. In the specific OCR scenario one can think of different aggregation

[2] www.crowdflower.com

methods. Because the answers consist of a set of letters, we suggest measuring the overall Levenshtein distance d_i^l as the sum of all Levenshtein distances d_{ij}^l of answer i to all the other answers j and propose that the most probable correct answer is that with the shortest distance to all other answers (equation E8). Ties are also broken by chance.

(E8) $$d_i^l = \sum_{j=1;j\neq i}^{M} d_{ij}^l$$

The data of the evaluation has been used in order to compare the two aggregation methods. All possible combinations of answers have been used. Figure 6 shows the results of both mechanisms in terms of accuracy and number of votes.

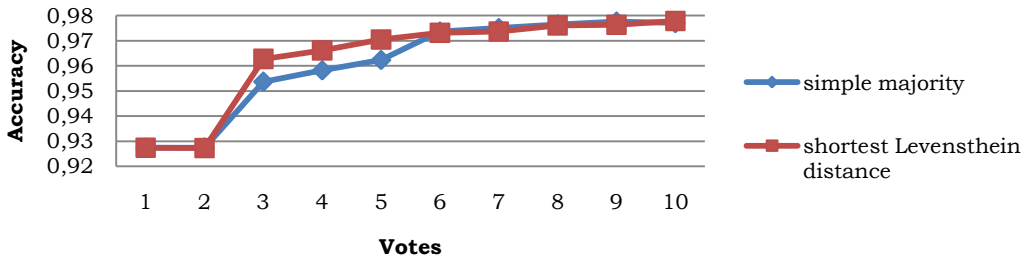

Figure 6. Accuracy of simple majority vote using different aggregation methods

As we see, the aggregation method using the shortest Levenshtein distance indeed outperforms the simple majority vote method, especially when using a low level of redundancy.

4.2.5 Performance of the weighted majority vote approach

Using the weighted majority vote approach, we achieve remarkably good quality and we almost exactly met the quality goal of 0.99. It is not surprising that there is a small deviation because the workers' error rates reflected in the model are only approximations using historic values and therefore not 100% accurate. Table 2 shows the results of Binomial and Poisson majority vote in comparison to the basic majority vote procedure requiring a majority of more than 50% and breaking ties at random. We see that our binomial (98.36%) and Poisson (98.45%) weighted majority vote procedure even outperforms the accuracy of a ninefold simple majority vote (97.76%). That is a remarkable result given that the weighted majority vote is 4 times more efficient as it requires only 2.21 (Poisson) and 2.25 (Binomial) workers per task compared to 9 workers per task for the basic ninefold majority vote approach. In other words: the weighted majority vote approach has reduced the quality management effort by some 75 percent compared to the basic majority vote approach. Figure 7 illustrates this relation based on the data from table 2.

Approach	MV 2	Poisson	Binomial	MV 3	MV 4	MV 5	MV 6	MV 7	MV 8	MV 9	MV 10
Average redundancy	2	2.21	2.25	3	4	5	6	7	8	9	10
Accuracy	0.927	0.985	0.984	0.954	0.958	0.962	0.974	0.975	0.977	0.978	0.977

Table 1. Comparison of the accuracy of different majority vote procedures

We achieve this quality by incorporating the workers' error rates and escalating tasks that lead to uncertain outcomes. As expected, there are only slight differences in terms of quality between binomial and Poisson case, with binomial majority vote achieving an average outgoing fraction defective of ~0.0164 and Poisson majority vote achieving an average outgoing fraction defective of ~0.0156. These values have been evaluated in several runs and differ only slightly.

Figure 8 shows the amount of redundancy leveraged in the majority vote mechanism (binomial and Poisson case). We observe that in the binomial case more tasks end with two votes (88%) than in the Poisson case (85.5%), while the number of tasks with more than 4 votes is higher in the binomial case (10.2%) than in the Poisson case (6.2%). This is not surprising if we take a closer look at the binomial aggregation formula (E3). We see that in almost all cases where the first two workers agree, no additional worker is involved in the process. Given there is one worker who has a result that differs from the others, we almost always need at least 3 workers who agree to outvote this worker, of course depending on the individual error rates. This leads to the result that the Poisson case is not only slightly better in terms of accuracy but also more efficient with an average of 2.21 workers per instance, compared to 2.25 workers per instance in the binomial case.

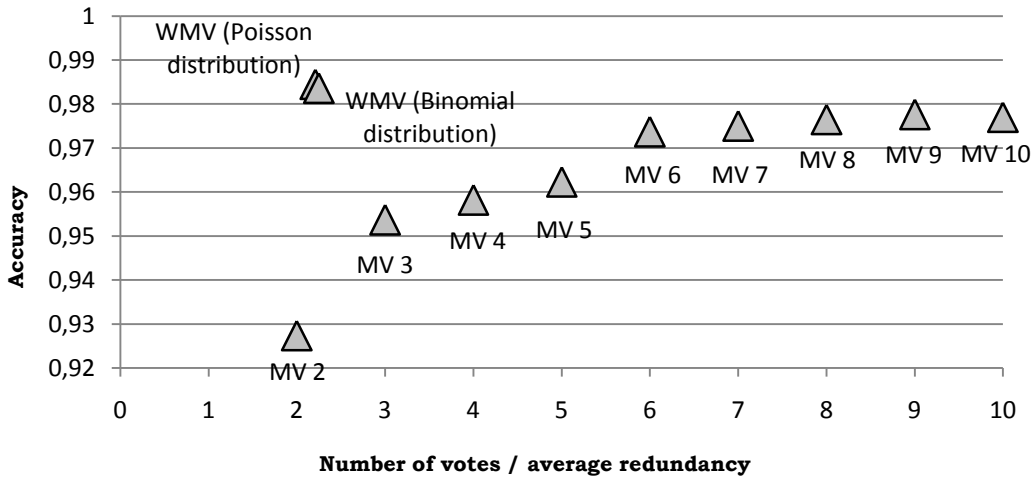

Figure 7. Comparison of the accuracy of different majority vote (WMV) procedures

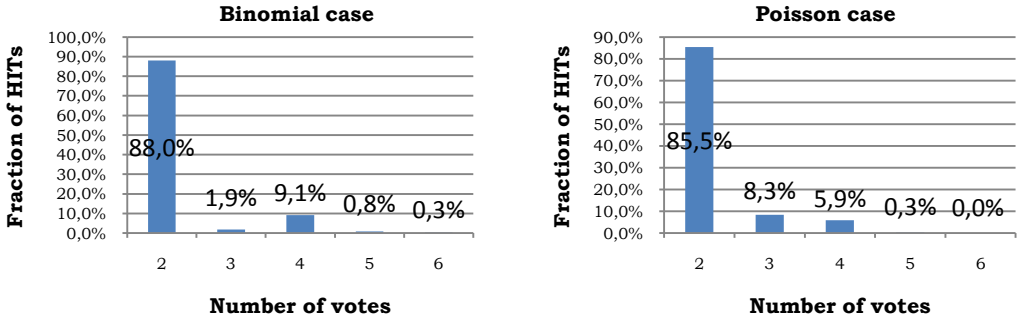

Figure 8. Amount of redundancy in the binomial case and Poisson case

5 Conclusion

We have proposed and evaluated a *weighted majority vote* (WMV) approach that supports goal-based quality management for human-based electronic services (people services). The approach uses redundant task execution by dynamically including additional workers in the decision process until the desired level of quality is reached. An evaluation on Amazon's Mechanical Turk platform based on an optical character recognition (OCR) scenario has shown that the WMV approach is capable of meeting predefined quality requirements almost exactly. Furthermore, it significantly increases the efficiency of the quality management as it reduces the quality management effort by some 75 percent compared to the traditional majority vote approach. In its current form, the WMV approach can only be used for

deterministic tasks, i.e. for tasks for which a well defined optimal result exists. In our ongoing research we are widening the reach of the approach to more complex usage scenarios and extend it by elements of statistical quality control (SQC) in order to be able to reach a predefined level of result quality more precisely while further reducing the overall quality management effort.

6 References

Condorcet, M. le marquis de, and Caritat, A. N. (1785). Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. De l'imprimerie Royale, Paris.

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. Communications of the ACM, 7(3), 171–176.

Frei, B. (2009). Paid Crowdsourcing. Available at: http://www.smartsheet.com/files/haymaker/Paid Crowdsourcing Sept 2009 - Release Version - Smartsheet.pdf. Accessed 2011-04-05.

Gruca, T. S., Berg, J. E., and Cipriano, M. (2005). Consensus and differences of opinion in electronic prediction markets. Electronic Markets, 15(1), 13–22.

Howe, J. (2006). The Rise of Crowdsourcing. Wired magazine, 14(6), 1-4.

Kern, R., Bauer, C., Thies, H., and Satzger, G. (2010). Validating results of human-based electronic services leveraging multiple reviewers. Proceedings of the 16th Americas Conference on Information Systems (AMCIS). Lima, Peru.

Kern, R., Zirpins, C., and Agarwal, S. (2009). Managing Quality of Human-Based eServices. In G. Feuerlicht and W. Lamersdorf (Eds.), Service-Oriented Computing - ICSOC 2008 Workshops, ICSOC 2008 International Workshops, Sydney, Australia, December 1st, 2008, Revised Selected Papers (Vol. 547, pp. 304-309). Springer.

Lam, L., and Suen, C. Y. (1997). Application of Majority Voting to Pattern Recognition: An Analysis of Its Behavior and Performance. SMC, 27(5), 553-568.

Littlestone, N., and Warmuth, M. K. (1994). The Weighted Majority Algorithm. Information and Computation, 108, 212–261.

Lopresti, D. (2009). Optical character recognition errors and their effects on natural language processing. International Journal on Document Analysis and Recognition (IJDAR), 12(3), 141-151. Springer.

May, K. (1952). A set of independent, necessary and sufficient conditions for simple majority decisions. Econometrica, 20, 680-684.

Revow, M., Williams, C. K. I., and Hinton, G. E. (1996). Using Generative Models for Handwritten Digit Recognition. IEEE Trans. Pattern Anal. Mach. Intell., 18(6), 592–606.

Ross, J., Irani, L., Silberman, M., Zaldivar, A., and Tomlinson, B. (2010). Who are the crowdworkers?: shifting demographics in mechanical turk. Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems (p. 2863–2872).

Rowe, G., and Wright, G. (1999). The Delphi technique as a forecasting tool: issues and analysis. International Journal of Forecasting, 15(4).

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. (2008). Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. Proceedings of the Conference on Empirical Methods in Natural Language Processing (p. 254–263). Morristown: Association for Computational Linguistics.

Sorokin, A., and Forsyth, D. (2008). Utility data annotation with Amazon Mechanical Turk. Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on (pp. 1-8).

Surowiecki, J. (2004). The Wisdom of Crowds (1st ed.). New York, USA: Doubleday.

Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., and Movellan, J. (2009). Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise (pp. 2035-2043).

Xu, L., Krzyzak, A., and Suen, C. Y. (1992). Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition. SMC, 22(3), 418-435.