Managing the Quality of Large-Scale Crowdsourcing

Jeroen B. P. Vuurens
Delft University of Technology
Delft, The Netherlands
j.b.p.vuurens@tudelft.nl

Arjen P. de Vries
Centrum Wiskunde & Informatica
Amsterdam, The Netherlands
arjen@acm.org

Carsten Eickhoff
Delft University of Technology
Delft, The Netherlands
c.eickhoff@tudelft.nl

ABSTRACT

Crowdsourcing can be used to obtain relevance judgments needed for the evaluation of information retrieval systems. However, the quality of crowdsourced relevance judgments may be questionable; a substantial amount of workers appear to spam HITs in order to maximize their hourly wages, and workers may know less than expert annotators about the topic being queried. The task for the TREC 2011 Crowdsourcing track was to obtain high-quality relevance judgments. The quality of obtained annotations is improved by removing random judgments and aggregating multiple annotations per query-document pair. We conclude that crowdsourcing can be used as a feasible alternative to expert annotations, based on the estimated proportions of correctly judged query-document pairs in the crowdsourced relevance judgments and previous TREC qrels.

1. INTRODUCTION

Evaluation of IR-systems generally uses known ground truth for every query-document pair. Ground truth is commonly obtained from expert annotators who manually judge relevance for each pair. Obtaining ground truth through experts is an expensive and time-consuming process [1].

Relevance judgments can be crowdsourced on the Internet by using anonymous web users (known as workers) as non-expert annotators [1]. Through the use of crowdsourcing services like Amazon's Mechanical Turk (AMT) or CrowdFlower, it is relatively inexpensive to obtain judgments from a large number of workers in a short amount of time. Typically, several judgments are obtained per query-document pair. A *consensus* algorithm is used to aggregate the judgments into a single outcome per pair [2].

The use of crowdsourcing for relevance judgments comes with new challenges. There have been several reports of workers spamming questions [3], [4], [5]. The random votes these workers produce can seriously affect consensus, especially at increased spam rates. Attempts to suppress random votes in a consensus algorithm showed mediocre results [6]. Therefore, an elimination strategy is used to detect spam and take it out of the dataset before determining consensus.

Section 2 discusses the importance to remove random judgments, while leaving room for difference in opinion. Section 3 gives the design of the used HIT, spam detection and management tool for obtaining results for Task 1. In Section 4, an adapted approach is given for computing consensus over the data for Task 2. The results for both tasks are described and analyzed in Section 5. In Section 6 we conclude that the results are comparable to those of expert annotators, at lower costs.

2. FRAMEWORK OF REFERENCE

2.1 Quality of relevance judgments

There is a distinct difference between random judgments and differences of opinion. Differences of opinion are inherent to subjective information needs. Voorhees compared the variance of relevance judgments created by different assessors and the intersection between assessors [8]. She concluded that different relevance assessments, created under disparate conditions, produce essentially comparative evaluation results. This study shows that differences of opinion do not affect the usefulness of qrels for evaluation. Random judgments on the other hand are useless for the evaluation of IR systems; a perfect IR system is expected to obtain the same score as a random machine, regardless of the measure used.

While there is no need to resolve differences in opinion amongst crowdsourcing workers, random judgments increase the variance of evaluation measures, and can render a test set useless if they are too abundant. We expect that the quality of relevance judgments can be increased by decreasing the proportion of random judgments.

2.2 Consensus for relevance judgments

The relevance judgments obtained from anonymous crowdsourcing workers are of unknown quality. Only part of the workers may use *ethical* behavior, as they follow instructions and aim to produce meaningful results [6]. A common approach is to obtain several judgments for each query-document pair, and combine these with a consensus algorithm [1]. The redundant information helps to filter out errors in judgment and cheat attempts.

2.3 Random judgments

Search results often contain duplicate documents, which contain the same content but have different URLs. In previous TREC datasets, duplicate documents that were retrieved for the same topic were judged by the same assessor. Scholer et al. found that 18% of the duplicate documents found in previous TREC datasets were judged inconsistently, when judgments are converted to a binary scale [7]. Every inconsistently judged duplicate can be seen as a random element within the set of relevance judgments, and will have the same value as random data when used in evaluation.

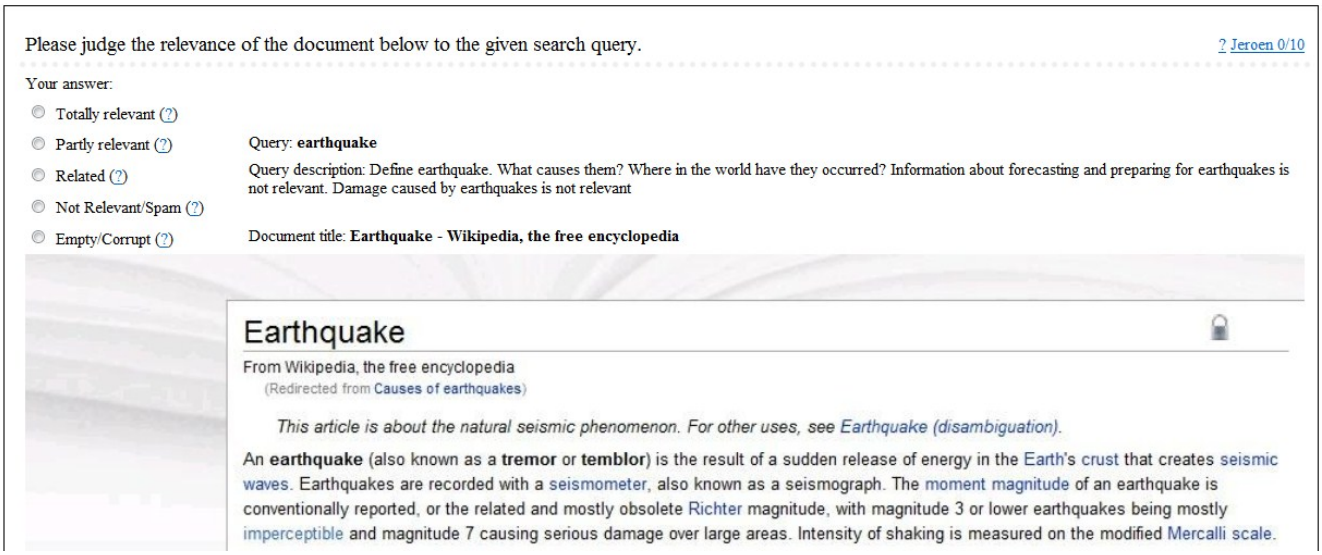

Figure 1: HIT design

In crowdsourcing, when two workers submit inconsistent judgments for the same query-document pair, we also consider this to be a random element. If both judgments would be used as separate judgments in a set of qrels, they would have the same value as random judgments. We expect to find more inconsistently judged duplicate documents via crowdsourcing annotations than expert annotations, because not every worker can be trusted to work the task as required. Previous studies report different types of spammers [4], [6]: *random spammers* try to randomize their responses hoping to stay undetected, *uniform spammers* repeat the same label over and over and *semi-random spammers* switch between ethical and spamming behavior [6]. We also suspect that workers that appear to submit random results, do not always have dishonest intentions. It is possible for workers to have a different understanding of the task, possibly caused by vague or ambiguous instructions, or workers having different frames of reference and abilities (for instance good understanding of English). There are countermeasures that can help to prevent these types of random results, such as clear instructions and the use of qualification tests to determine if a worker is capable of performing the task.

3. TASK 1: OBTAINING JUDGMENTS

3.1 Strategy

The dataset we were given for Task 1 is divided into sets of 5 documents each that are to be judged for a given topic. TREC 2011 requires that every worker has to judge all 5 documents within a set. Each HIT contains 2 sets, so every worker submits a minimum of 10 judgments. Every query-document pair is to be judged by a minimum of 5 workers.

The quality of obtained crowdsourcing results is increased by detecting workers that submit random results and replacing all of their judgments with judgments from another worker, until all accepted workers pass the spam detection filters. In this study, spammers are detected by comparing each worker's judgments to the judgments by other workers on the same query-document pairs, without additional gold set questions or pre-qualification tests.

3.2 HIT design

The HIT design as shown in Figure 1, is focused on clarity, simplicity and worker efficiency. Instructions were formulated as clear as possible to reduce inconsistencies due to unclear labels. The algorithm that is used for detection of random judgments requires judgments to be made on an ordinal scale. The 5 labels used were explained as follows:

- **Totally relevant**: the document completely answers the question.
- **Partly relevant**: the information in the document is relevant to the question but not complete.
- **Related**: the document mentions the subject or holds potentially good hyperlinks to relevant pages, but does not contain any actual information regarding the query itself.
- **Not relevant/Spam**: the document is off topic or spam, not giving information about the subject.
- **Empty/Corrupt**: a document that is corrupt, unreadable or empty.

On the screen, one query-document pair is displayed at a time for the worker to judge. The query is described by the query terms used and an additional query description that was supplied in the dataset, giving additional information on the meaning of relevant, partly relevant and not relevant. The document is represented by the title, which was extracted from the original HTML and a rendered image of the webpage. The rendered pages were cropped to a resolution of 1200x2400 pixels.

3.3 Removal of random results

Detecting random votes is more difficult at higher spam rates, because estimations are blurred by the noise that is present. In a previous study using simulations, we developed a measure for detecting random spammers that has a low false-positive rate in noisy environments [6]. The key idea here is to have relevance judged on an ordinal

scale instead of a binary one, taking advantage of the fact that ethical annotators are more likely to vote closer to each other than random spammers. The *random separator* algorithm uses this characteristic to detect random spammers.

The *uniform separator* algorithm detects uniform spammers by counting the number of errors made in repeating voting patterns, complementing the random separator [6]. Simulations with a worker population containing more than 20% uniform spammers showed that the uniform separator avoids the situation where spammers are not detected by the random separator due to the consensus being affected by those uniform spammers.

3.4 Crowdsourcing Management

The HIT was implemented on AMT as a frame without any static content. Within the frame of the HIT, an external question tag is used to show a page from our own webserver. When a worker views the HIT on AMT, an example of a question is shown, which is not the question the worker is going to answer. When a worker accepts the HIT, the webserver receives the worker ID from AMT and assigns 2 sets of 5 documents each to the HIT, guarding against any possible 'worker-training effects' by ensuring that the worker has not worked these query-document pairs before. The order of sets and documents within each set are shuffled to prevent workers from getting their assignments in the exact same order.

If a user accepts a HIT for the first time, the user is asked to enter his/her nickname, which is displayed in the top-right corner. To judge the assignment on the screen the user clicks a label (Figure 1). After judging the last document in the HIT, a thank you message with a submit button will appear, enabling the user to submit the HIT on AMT in order to get paid. The standard AMT interface then shows a screen that enables a user to continue by accepting another HIT from the same batch.

All judgments are registered on the webserver along with the start and end time. The AMT is only used to attract workers and to pay them afterwards. HITs that are not submitted within 15 minutes or HITs that are returned (i.e. not completed) by workers have their assignments reset so these can be assigned to another user.

The progress of the batch is monitored by an automated tool as follows: (1) The tool generates the assignments needed to obtain 5 judgments per query-document pair. (2) The assignments are uploaded to the webserver, waiting to be completed by workers. (3) Progress is monitored and if all assignments have been completed by workers, the relevance judgments are automatically downloaded and analyzed. (4) The worker(s) that submitted the most random judgments are removed along with all their judgments, but only up to the point where at least 4 judgments for every pair remain, because the replacing votes can affect the spam detection scores for the other workers working the same query-document pairs. (5) If there are query-document pairs with less than 5 judgments, the tool jumps back to (1). Otherwise the batch is complete, having obtained all required judgments by workers that have passed spam detection.

The tool automatically puts the required number of HITs on AMT and monitors these, using the AMT API. Workers that are suspected of spamming are rejected, but the system does require manual confirmation before rejecting the corresponding HITs on AMT. When a batch is completely finished, all accepted work is paid.

4. TASK 2: CONSENSUS

4.1 Strategy

The second task of the TREC 2011 Crowdsourcing track was to aggregate the binary labels obtained from crowdsourcing workers for several query-document pairs. The approach that was used for Task 1 requires that results that are rejected are replaced, that judgments are made on an ordinal scale, collecting 5 judgments per pair and at least 10 judgments per worker to reduce false detection of spammers. The dataset for Task 2 does however not meet these requirements, so while we use the same strategy as for Task 1, and our approach had to be adapted to this specific situation.

The dataset contains workers who exclusively use the exact same label for more than 100 pairs. If we assume the underlying distribution of relevance to be balanced, this indicates the presence of uniform spammers. Instead of the uniform separator applied in Task 1, that has not been tested on binary judgments, we used a simple rule: workers are removed if they cast over 80% of their votes on the same label.

The random separator algorithm does not work on binary labels. Instead the average percentage of inter worker agreement was used to estimate how ethically they work. This measure was applied using the same iteration scheme that is used for the random separator, based on the assumption that the workers below a certain threshold are producing random results and the worker with the lowest inter worker agreement is the most likely random spammer. Iteratively, the most likely random spammer is removed and inter worker agreement is recalculated. Because this estimate is less accurate than using the random separator, detection will be less precise, causing more false-positives. Still, this is expected to remove the majority of random spammers while leaving the majority of ethical workers in the pool. Workers are removed if they have less than 70% agreement with other ethical workers.

The most likely label for a query-document pair is calculated by multiplying the probabilities that accepted workers voting that label are correct and that accepted workers voting the opposite label are wrong. The probability that a worker makes a correct judgment is estimated by their average agreement with accepted workers. The documents were ranked according to the probability that the selected label is correct.

5. RESULTS

5.1 Task 1

In Task 1, relevance judgments had to be obtained for 2,165 query-document pairs, arranged into 433 sets of 5 pairs. The sets were split into a team-specific batch (510 pairs) and the shared batch (1,655 pairs). The size of all HITs was fixed for this experiment to 10 query-document pairs. One extra set of 5 query-document pairs from the team-specific batch was therefore added to the shared batch to make the batch size a multiple of 10.

No restrictions or qualifications of any kind were used to preselect workers. In total, 503 unique workers submitted judgments for our HITs. 40 of them worked on both batches. The submitted work was analyzed afterwards, and workers that worked both batches were evaluated for each batch separately. The 543 worker evaluations (503+40) resulted in 262 acceptances and 281 rejections.

In total, 20,840 judgments were obtained of which 55% (11,510) were accepted and 45% (9,330) were rejected. To determine consensus majority voting was used, and when votes tie an EM model, as described in [6]. On 20.5% of the cases votes tie because an ordinal scale is used instead of a binary scale. The ordinal labels where then converted to binary labels, after which 78.9% of the accepted judgments agree with consensus. In comparison, 55.4% of the rejected judgments agree with consensus. The higher than random percentage for rejected judgments can be explained by incidental rejection of ethical workers and by rejected semi-random spammers; workers who switch between ethical and random voting behavior [6].

The payment was $0.055 per accepted HIT, each HIT consisting of 10 query-document pairs. The workers that were suspected of spamming were rejected along with all the HITs they submitted. The total expenses were $66.85.

On the team-specific batch, it appeared that the uniform separator algorithm did not scale well with the number of worked HITs, causing it to falsely detect 2 workers working more than 20 HITs. The algorithm as presented in [6] was re-normalized to Formula 1. S is a collection of all possible label sequences s with length $|s| = 2$ or 3. We calculate the number of disagreements $disagree_{ij}$ between the workers' judgments $J_{s,w}$ that occur within occurrences of label sequence s, with judgments $J_{j,\bar{w}}$ submitted by other workers for the same query-document pair as judgment j. To reduce false detections of ethical workers $disagree_{ij} = 0$ if $disagree_{ij} < 2$. f_{s,J_w} is the frequency of label sequence s occurring within worker w's time ordered judgments J_w.

$$UniformSep_w = \frac{\sum_{s \in S} |s| \cdot (f_{s,J_w} - 1) \cdot (\sum_{j \in J_{s,w}} \sum_{i \in J_{j,\bar{w}}} disagree_{ij})^2}{\sum_{s \in S} \sum_{j \in J_{s,w}} |J_{j,\bar{w}}|}$$

Formula 1: re-normalized uniform separator.

Using simulation, we found that the worker with the highest UniformSpam score that is above the empirically determined threshold of 1.2 is likely to be a uniform spammer.

Due to technical problems in the crowdsourcing management system, we unintentionally over-obtained a total of 660 accepted judgments on Task 1, which were kept in the dataset. Also, by mistake, the dataset contains 5 judgments that were marked as rejected while the other judgments by the same worker were accepted. These 5 judgments should have been marked as accepted as well.

5.2 Time analysis

The start and end time of every judged query-document pair were registered. Figure 2 shows the average time-per-question vs. the questions in the order that the workers answered them (question sequence number). For Figure 2, only judgments with a time-per-question < 80 seconds were used to suppress the noise from users taking long breaks not actually working the assignments.

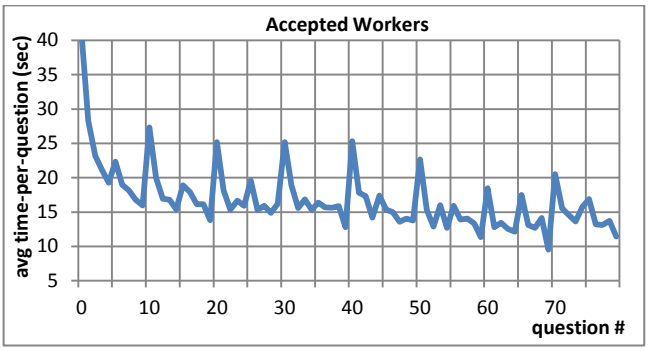

Figure 2: Average time-per-question for accepted workers

For the first assignment of every HIT, workers take more time on average to read the question. Every sixth question the worker is likely to get a different topic to match the next 5 documents to, increasing the average time-per-question as they have to read it. The time-per-question typically decreases after the first and sixth question of every HIT, which is presumably caused by some training effect, gaining a better understanding of the question and having seen other documents to compare to. The overall downward trend in Figure 2 indicates that workers become faster as they work more HITs. Analysis showed that workers who submitted more HITs do not work faster on previous HITs than workers who submitted less HITs.

In Figure 3, a similar time-per-question analysis is done for rejected workers. On the first 20 questions the average time-per-question pattern for ethical and rejected workers is very similar. After working these 3 HITs, rejected workers take more time for the first question of each HIT, but the

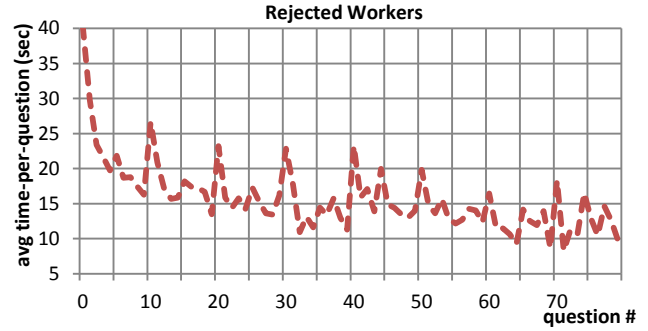

Figure 3: Average time-per-question for rejected workers

time-per-question pattern in between the first questions becomes more random compared to accepted workers. It seems that in this phase rejected workers are less likely to read the new topics that are presented on the screen.

66% of the workers worked only 1 HIT. The spammers amongst these show the same time-per-question pattern as the ethical workers. It seems unlikely that their random votes are caused by not reading the new topic that is presented halfway.

Several reports suggest time-on-task as a criterion to detect spam [3][10]. Although in this experiment the fastest 5 workers have an average time per judgment lower than 5 seconds and were all detected as spammers, 34% of the 56 workers with an average time per judgment lower than 10 seconds were found to be ethical. While time-on-task can reveal unrealistically fast workers as spammers, it appears to be a weak stand-alone measure for general spammer detection.

To get an indication of the volumes that can be processed at the used pay-rate, the pick-up rates were analyzed. The highest pick-up rate was observed during the first iteration of the shared batch, obtaining 8,300 relevance judgments in 12 hours. After the first iteration of a batch, workers were rejected in small amounts to minimize false rejections. The number of HITs in consecutive iterations were therefore small, often 1 to 6 HITs. A repost of 1 HIT on AMT took longer to get picked up than a task that is larger in volume. Presumably, a larger volume is more attractive to workers as it enables them to work more efficiently, repeating a similar task several times, becoming faster as they train, while only having to read the instructions once.

On the team-specific batch, the average pick-up rate after the first iteration was 2 HITs per hour. Just before starting the shared batch, the HIT management tool was altered to fake the number of available HITs on AMT. The tool put three times as many HITs on AMT as were actually available, automatically taking off the excessive HITs as soon as a HIT was accepted by a worker. This increased the average pick-up rate to 6 HITs per hour.

5.3 Comparison of judgments vs consensus

In Figure 4, a grouped bar chart shows the frequency of labels chosen by accepted workers as the primary groups along the x-axis. Within each primary group, the chosen labels are distributed over the consensus on corresponding query-document pairs. The solid bars within each group are the judgments of accepted workers that agree with consensus. The majority of judgments agree with consensus or are on adjoining labels to consensus. While difference of opinion does exist, judgments are less likely to be made on the opposite end of the scale.

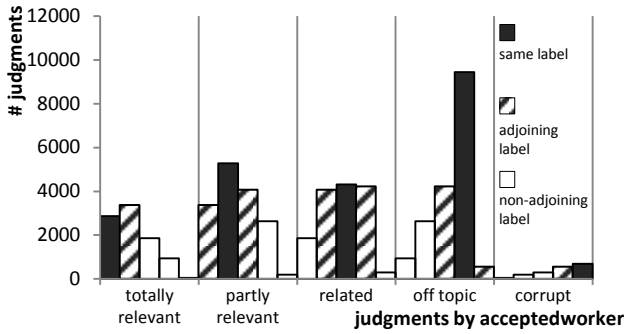

Figure 4: judgment frequencies of accepted workers compared to consensus

Figure 5 shows the same graph for rejected workers compared to the consensus taken from accepted workers. Both accepted and rejected workers rarely used the label corrupt/empty. The characteristic difference of spammers being more likely to vote further away from other workers on an ordinal voting scale is visible as the votes in Figure 5 are distributed more randomly across the remaining four labels, resulting in 35% of their judgments on non-adjoining labels as opposed to 20% by accepted workers.

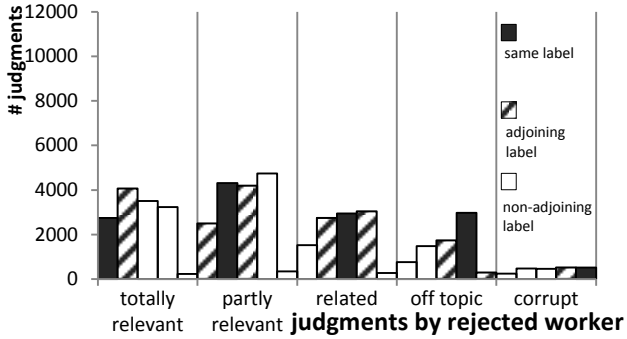

Figure 5: judgment frequencies of rejected workers compared to consensus

Of the 40 workers that worked both batches, 13 were rejected for one batch while being accepted for the other. Without looking at the submitted judgments, we manually inspected the 61 query-document pairs where these rejected workers submitted a judgment at least two labels away from consensus. We agreed with the worker on 36% and with consensus on 64%. For 4 out of 13 workers, we found that they would not have been labeled as spammer if consensus was replaced by our judgments on the query-document pairs we judged. It could be that spam detection settings are a bit too strict, generating false detections. Further inspection showed that these 4 workers primarily voted away from consensus on the topics “Lake Murray Fishing”, “Sudoku” and “DIY audio”, and that these 3 topics may have conflicting instructions asking for references and software, while the general instructions state that information must be on the page that is shown, and promising hyperlinks are to be regarded as not relevant. Creating clear instructions may increase agreement amongst workers and help avoid false rejections.

5.4 Estimation of correct judgments

The inconsistencies reported by Scholer et al. [7] can be used to estimate the proportion of incorrectly judged documents in previous TREC qrels (Table 1). The ternary labels were converted to binary labels (partly relevant becoming relevant). In total, the authors reported 24,327 consistently judged pairs and 5,514 inconsistently judged pairs.

Table 1: estimation of correct judgments

	TREC qrels	crowdsourced
Consistent duplicate	24,327	17,116
Inconsistent duplicate	5,514	8,459
$P_{judgmentcorrect}$	0.897	0.791
#annotators per pair	1	5
Est. correctly judged	89.7%	93.5%

From these numbers we can estimate the probability that a qrel is correct (Formula 2). A pair of binary judgments is consistent when either both judgments are correct or both are incorrect. Using the quadratic formula we find the probability of a judgment being correct, which is 89.7% for previous TREC qrels.

$$P_{consistent} = \frac{24{,}327}{24{,}327 + 5{,}514}$$

$$P_{judgmentcorrect}{}^2 + (1 - P_{judgmentcorrect})^2 = P_{consistent}$$

$$P_{judgmentcorrect} \approx 0.897$$

Formula 2: estimate probability judgment is correct

For an analysis of inconsistencies within the crowdsourced results for Task 1, judgments performed by different workers on the same query-document pair are used as duplicates. Applying Formula 2 results in the estimation that 79.1% of the crowdsourced judgments are correct.

For the crowdsourcing task, we obtained 5 judgments per query-document pair. Sheng et al. estimate the integral quality by using majority voting, assuming uniform worker quality and uniform task difficulty [9]. Using Formula 3 we estimate that 93.5% of the crowdsourced qrels for Task 1 are correct.

$$EstCorrJudged = \binom{5}{0}p^5 + \binom{5}{1}p^4(1-p) + \binom{5}{2}p^3(1-p)^2$$

Formula 3: estimating quality by majority voting using uniform worker quality

A possible weakness in this estimation is the implicit assumption that the probability of inconsistently judged documents is transferable to the probability of incorrectly judged consistent pairs. Consistently judged pairs could be incorrect more often if spammers get the upper hand or if false consensus is being obtained by simply removing workers that agree least.

Spammers can affect consistent pairs more than inconsistent pairs by giving the same judgments on the same query-document pairs. The detection mechanisms that are used guard specifically against the repeated voting patterns of uniform spammers. To guard against the possibility of organized spammers entering the same sequence of votes on the same query-document pairs, the pairs are shuffled within each set of 5 pairs and the sets are shuffled across HITs.

Looking into the possibility that false consensus is being obtained by simply removing workers that agree least, we note that disagreement with consensus alone does not lead to rejection by the algorithms used. Primarily disagreeing judgments that are not on adjoining labels to consensus or disagreements within repeating voting patterns will contribute to rejection, being tolerant on workers that work honestly and have a different opinion. The inter worker agreement between accepted workers is 67%, which is what can be expected under normal circumstances, close to what was found in previous studies [1] and comparable to the average agreement our workers have to the TREC consensus. This verifies that the inter worker agreement was not overestimated by rejecting too many workers that disagreed. The strength of crowdsourcing does not lie in superhuman quality of workers, but the extremely low costs making it feasible to aggregate the results of several ethical workers for every query-document pair.

5.5 Task 2

The dataset that was provided for Task 2 contains 19,033 binary relevance judgments submitted by 762 workers. The relevance judgments are not equally balanced over the query-document pairs; some query-document pairs have been judged by only 1 or 2 workers while others have been judged by over 10 workers.

When first running consensus without removing any workers, the large majority of documents was labeled relevant for their query. Inspection showed there to be twice as many votes on relevant as on irrelevant. In the result set, we found a large number of uniform spammers who judged the vast majority of documents as relevant. Given the high frequency of this happening, it could affect the resulting consensus.

Using a simple rule to replace the uniform separator, 256 workers that judged more than 80% with the same label were rejected, for giving unreliable results.

When all workers that have less than 70% inter-worker agreement were removed, 19.5% of the query-document pairs had zero accepted workers left and 34.8% of the pairs had only one vote by an accepted worker. The threshold was lowered to 62% inter worker agreement, which decreased the number of query-document pairs with zero or one accepted workers to 31%. 148 workers were rejected for having less than 62% inter worker agreement with 'better' workers.

Each worker's probability (Pw) to give a correct answer was estimated by the agreement with accepted workers. The probability of a query-document pair being relevant was

calculated by multiplying the Pw of accepted workers voting relevant and (1-Pw) of accepted workers voting irrelevant. The same was done for the label irrelevant, balancing the sum of the label probabilities to 1. The most likely label was selected, and the probability that the correct label was chosen was used for ranking. For 8% of the query-document pairs, all workers were rejected. In those cases the judgment from the rejected worker with the highest inter worker agreement was taken.

The average agreement between accepted workers with consensus is 85.5%. However, because 31% of the votes were determined by the judgment of only one worker, the inter worker agreement is overestimated. The agreement between rejected workers and consensus is 58%. However, 8% of the query-document pairs have no accepted workers, so the 'best' rejected worker's vote was used, again overestimating agreement with consensus.

6. CONCLUSION

For the TREC 2011 Crowdsourcing track, teams were given the task to obtain high quality relevance judgments from individual crowdsourcing workers. Our strategy was to detect and eliminate workers that submit random votes. 45% of the submitted judgments were rejected.

The average agreement amongst accepted workers, between accepted workers and the TREC participants' consensus and between accepted workers and the gold standard are all close to 67%. To compare the quality of obtained crowdsourced judgments to those of expert annotators, the proportion of correctly judged documents was estimated from an analysis of the consistency of duplicate judgments. We estimate that 90% of the qrels of previous TREC datasets were correctly judged and 95% of the qrels that result from the crowdsourced annotations of Task 1 are correct. The strength of crowdsourcing does not come from the highest performance of individuals, but from aggregating several annotations obtained from ethical workers. The conclusion is that crowdsourcing can be a feasible alternative to relevance judgments submitted by expert annotators.

An analysis of time-on-task revealed a learning curve for workers, needing less time for judgments as they worked more HITs. Rejected workers appear to have the same time-on-task pattern for the first two HITs, indicating that rejected workers in general read the question on the first two hits. The analysis of time-on-task also indicates that rejected workers, after working three HITs, do not use more time when new topics are shown.

The second task for the TREC 2011 Crowdsourcing track was to estimate the most likely true label given a set of obtained relevance judgments obtained via crowdsourcing. The same strategy as for Task 1 was used, to remove workers that most likely submitted random votes before determining consensus. 53% of the workers were rejected. The relevance of each pair was computed by multiplying the probabilities that workers were (in)correct. The results scored well above average compared to other TREC participants, demonstrating that removal of random judgments should have priority over determining consensus.

Both on Task 1 and Task 2, spammers were found to vote more on the relevant labels, perhaps because they expect more documents to be relevant than irrelevant. A skewed label distribution amongst spammers increases their chance to coincide and affect consensus, increasing the necessity to properly filter out spammers.

7. REFERENCES

[1] O. Alonso, D. E. Rose and B. Stewart. Crowdsourcing for relevance evaluation. In *SIGIR Forum*. volume 42(2). pages 9-15. 2008.

[2] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni and L. Moy. "Learning from crowds." The Journal of Machine Learning Research, volume 99: 1297-1322. 2010.

[3] A. Kittur, E. H. Chi and B. Suh. Crowdsourcing user studies with Mechanical Turk. In *CHI 2008*. pages 453-456. 2008.

[4] D. Zhu and B. Carterette. An analysis of assessor behavior in crowdsourced preference judgments. In *Proceedings of SIGIR 2010 CSE Workshop*. pages 21-26. 2010.

[5] J. Le, A. Edmonds, V. Hester and L. Biewald. Ensuring quality in crowdsourced search relevance evaluation. in *Proceedings of SIGIR 2010 CSE Workshop*. pages 17–20. 2010.

[6] J. B. P. Vuurens, A. P. de Vries and C.Eickhoff. How Much Spam Can You Take? An Analysis of Crowdsourcing Results to Increase Accuracy. In *Proceedings of SIGIR 2011 CIR Workshop*. 2011.

[7] F. Scholer, A. Turpin and M. Sanderson. Quantifying Test Collection Quality Based on the Consistency of Relevance Judgments. In *SIGIR 2011*. 2011.

[8] E. M. Voorhees. Variations in Relevance Judgments and the Measurement of Retrieval Effectiveness. In *SIGIR 1998*. 1998.

[9] V. S. Sheng, F. Provost and P. G. Ipeirotis. Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers. In *Proceedings of KDD 2008*. pages 614-622. 2008

[10] G. Kazai. In Search of Quality in Crowdsourcing for Search Engine Evaluation. In *ECIR 2011 - Advances in Information Retrieval*. pages 165–176. 2011.