How Much Spam Can You Take? An Analysis of Crowdsourcing Results to Increase Accuracy

Jeroen 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 is used to obtain relevance judgments for query-document pairs. To obtain accurate judgments, each query-document pair is judged by several workers. Consensus is usually obtained by majority voting and spam most commonly reduced by injecting gold set questions. This study puts the performance of gold sets and majority voting to the test. Based on the analysis of crowdsourcing results for a relevance judgment task, we propose an alternative to reduce spam and increase accuracy. Simulations were used to compare performance between different algorithms, inspecting accuracy and costs for different experimental settings. The results show that gold sets and majority voting are less spam-resistant than many believe and can easily be outperformed.

General Terms

Algorithms, Measurement, Design, Reliability, Experimentation.

Keywords

Crowdsourcing, Relevance judgments, Accuracy, Spam, Simulation.

1. INTRODUCTION

Evaluation of IR-systems generally uses known ground truth for every query-document pair. Ground truth is commonly obtained from experts 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 annotations are obtained per query-document pair. These are often combined by majority voting into a single more *accurate* (correct) outcome per pair [2]. One study showed that the accuracy obtained from majority voting over crowdsourced annotations can match or even exceed the quality of annotations done by single experts [3].

The use of crowdsourcing for relevance judgments comes with new challenges. There have been several reports of workers spamming questions [4,5,6]. This has led the research community to question the accuracy of relevance judgments obtained by crowdsourcing. Different approaches to detect spammers have been proposed, sometimes as simple as looking at the time spent per task [4,7], or more commonly by comparing workers' answers on gold set questions to the known correct answer which is already known.

This research examines the effect that spam has on the accuracy of relevance judgments obtained through crowdsourcing. An analysis of crowdsourcing results was done to discover different characteristics between spammers and faithful workers, enabling the design of new algorithms to detect spam. The popular and frequently used gold sets and majority voting will be compared to these new algorithms through simulation. This research aims to reveal how well the accepted measures of quality control perform when facing increased spam rates.

In Section 2 we present a classification of crowdsourcing workers, based on literature and the analysis of crowdsourcing results. In Section 3 we will discuss relevant theories used for this research, and propose additional algorithms to separate workers from different classes. In Section 4 we describe the setup of the simulations. Section 5 analyzes the results of these simulations to evaluate the impact of spam on the accuracy of crowdsourcing results, and how each separate instrument is affected when spam increases. Section 6 shows how combining the proposed algorithms improves accuracy, and how the results compare to gold sets with majority voting. Section 7 discusses the results of this study and present the results of the proposed algorithms in a small field study.

2. ANALYSIS OF CROWDSOURCING

2.1 Demographics

The worker population on AMT is diverse and changes over time. Ross et al. [8] reported a population mainly consisting of females (66%), mid-30 (40%) from the USA (83%) in November 2008. From November 2008 till November 2009 there was an increase in workers from India (5%-36%). While the majority of the initial crowdsourcing population was believed to work crowdsourcing tasks out of curiosity or as entertainment, the change in population increased the number of people who rely on crowdsourcing income for basic end needs (27% India, 14% USA). This may explain the increasing number of reports of spam on crowdsourcing platforms, i.e. workers trying to get paid without performing the task as required, giving useless answers instead.

2.2 Classification of workers

To study spammers in detail, different worker groups were characterized. Workers that contribute to accurate crowdsourcing results are called *ethical workers,* as they follow instructions and aim to produce meaningful results. However, Le et al. mention ethical workers that deliver poor judgments [6]. These poor judgments may be the results of ethical workers misinterpreting the requester's intent of the task, but can also be caused by a different frame of reference or as a consequence of being less capable of performing the task as required. Ethical workers who deliver poor judgments are called *sloppy workers* and ethical workers who produce adequate or better judgments are called *proper workers*.

Zhu and Carterette performed an analysis of voting behavior. One group showed a voting pattern of rapidly alternating labels. These workers may exhibit an advanced cheating behavior by purposely trying to randomize their responses, so that it would be difficult for requesters to discover their dishonesty [5]. This group is likely to have an average *precision* (percentage of correct answers) close to random, making them suspects of spamming. This type is called *random spammers*.

Another type described by Zhu and Carterette uses a fixed voting pattern. These workers seem neither interested in performing the task as intended, nor in advanced cheating, as they mostly repeat the same answer [5]. These workers are called *uniform spammers*. Although the long repeating patterns they produce are often easily detected when manually inspected, automated spam detection may overlook them as they increase the chance of voting in accordance with other uniform spammers over many votes, or to get more answers right on a skewed label distribution.

To confirm the classification found in literature, crowdsourcing results were obtained on 10 topics from the TREC 2010 Web Track ad-hoc task. 10 HITs were put on the AMT at 5 cents per HIT, to judge 100 query-document pairs on an ordinal 5-point scale. The 850 votes obtained from 33 workers were manually analyzed, identifying differences between worker types, comparing every vote to the majority vote for each query-document pair. The results in Table 1 show 55% of the workers to be classified as a proper worker, producing judgments with an average precision of .75 and a minimum precision of .60. 9% of the workers are categorized as a uniform spammer as they mainly repeated 2 out of the 5 possible labels with only few label switches. 30% of the workers are suspected of random spamming. Typically, they frequently vote far away from the correct label while avoiding the extremes of the scale and produce results of poor quality. We did not observe the rapidly alternating patterns described by Zhu and Carterette [5].

Interestingly, 3 of the workers that show the characteristics of a random spammer, answered with an average precision of .52, making it unlikely that these workers spam all questions. They possibly switch to proper voting on some questions, for instance on easy questions or when they expect some type of qualification. These workers are called *semi-random spammers*. The remaining 9 random spammers have an average precision of .20.

The remaining workers did not show any spammer characteristics. They work with a precision higher than random but not high enough to be categorized as proper workers. These workers are categorized as sloppy workers. They may have good intents, but deliver judgments of an inferior quality.

3. FRAMEWORK OF REFERENCE

3.1 Majority voting

To obtain more accurate judgments of query-document pairs in noisy environments, multiple votes are gathered for each pair. These can be aggregated into a single most likely answer by a consensus algorithm. A commonly used consensus algorithm is majority voting [2]. Although it can give good results for relevance judgments in crowdsourcing, majority voting is also criticized. A possible weakness is the implicit assumption that all annotators are equally good [2], while worker quality is considered to be diverse (as confirmed in Table 1).

3.2 Worker error-rates

Several studies have reported successes using different approaches to determine consensus [9,10,2]. Considering that the quality of workers varies, so should the weight of their votes, possibly resulting in an outcome different from the simple majority vote. A complicating factor is that a worker's quality is not easy to determine.

When collecting relevance assessments on an ordinal scale, it is possible to take each assessor's ability to score the different relevance grades into account. This situation is analogous to the medical diagnosis scenario described by Dawid and Skene [11]. Specifically, Dawid and Skene describe a case where 5 anesthesiologists diagnose 45 patients. The diagnosis is done on an ordinal four-point scale. They introduce a (straightforward) statistical model of this scenario:

- T: the probability that a label is the correct answer for a unit.
- π: the probability that a worker votes a label, given the probability of correct answers T.
- P: the prior probability for each label

Dawid and Skene use an expectation maximization (EM) algorithm to estimate the individual error-rates of the workers. The EM algorithm converges towards a local optimum, performing the following 2 steps iteratively: (1) Estimate the correct label T to each question, using multiple answers, taking the quality of each worker into account. (2) Estimate worker

Table 1. Classification of workers with observed percentage

Type	Description	Proportion[1]	Average precision[1]
Proper worker	Performs the tasks as requested. Reads the question and data and judges sufficiently precise.	55%	.75
Random spammer	Gives useless answers without reading the question or data, trying to hide their cheating behavior.	21%	.20
Semi-Random spammer	Gives useless answers on the majority of questions but answers a few questions properly, hoping to avoid spam-detection.	9%	.52
Uniform spammer	Chooses one label to primarily vote, sometimes switching labels on different types of questions.	9%	.35
Sloppy worker	Views the question and data, but may be insufficiently precise in their judgments.	6%	.45

[1] The proportions and average precisions reported here are the result of a field experiment discussed in Section 7.

quality π by comparing given answers to the estimated correct answer. For the first iteration the worker quality is unknown, therefore the first estimation of correct labels has to be seeded differently. In this research T was seeded with the majority vote.

The worker error-rate obtained by the EM-algorithm can be used as weights when aggregating votes for each query-document pair.

3.3 Gold sets

In crowdsourcing, gold sets are used to assess each worker's quality on questions that are representative of the actual task and have known answers [4]. These gold set questions are hidden in the actual task. Workers that fail on too many gold set questions are rejected. This method of qualification is often used to filter out spammers and retains the majority of proper workers.

Gold sets are easy to understand, explain and implement, which explains their popularity. But gold sets are also a coarse instrument, since workers may be qualified over a few questions. As a result, the use of gold sets alone may not identify all spammers [5], and a sizeable amount of proper workers may be rejected just for being unlucky on the gold set questions. Gold sets leave no room for difference in opinion, making them less suitable for less factual questions. Furthermore, spammers are aware of the use of gold set questions and the qualification mechanism. As a result they may switch voting behavior on the unambiguous easy questions they suspect are gold set questions.

3.4 Removal based on random error

The 'incidences of error probabilities' are presented by Dawid and Skene (Table 2), to display the probability that a worker gives an observed response, given the true (estimated correct) response [11]. Dawid and Skene describe the diagonal of correct allocations that is marked in Table 2, containing the cells for which the observer gave the same response as the estimated correct response. In the 5 error of incidences tables that Dawid and Skene present for the 5 anesthesiologists, it can be seen that when experts err, they err on labels adjoining the diagonal of correct allocations. This was also observed in the crowdsourcing results mentioned in section 2.2, where ethical workers consistently voted on and around the diagonal of correct allocations, while spammers made errors further away from the correct answer on an ordinal scale.

Table 2. Diagonal of correct allocations [11]

		Observer 1			
Observed response:		1	2	3	4
True response	1	.36	.04	.00	.00
	2	.03	.37	.02	.00
	3	.00	.04	.07	.00
	4	.00	.00	.04	.03

For the separation of random spammers from ethical workers, we introduce the term *random error* ε_r which is equal to the ordinal difference between the label a worker answered and the (estimated) correct answer. The *RandomSep* function is designed to separate random spammers and ethical workers, based on the average squared random error present in the collection of votes V casted by each worker.

$$RandomSep = \frac{\sum_{v \in V} \varepsilon_r^{\,2}}{|V|}$$

Figure 1 illustrates the use of RandomSep with an example of a worker population, before RandomSep is applied. In this figure, the median RandomSep for ethical workers is 1 and for random spammers it is 3. The workers with the highest RandomSep score (righthand side) are random spammers. Separation of random spammers and ethical workers is hindered by the overlap between the classes. This overlap is presumed to be caused primarily by the noise from random votes, clouding the prediction of correct labels. When the random spammers are removed from the right, their random votes are also taken out. Consequently, the prediction of correct labels becomes more accurate, causing the RandomSep score for random spammers to increase and the score for ethical workers to decrease, seperating the classes further, 'pushing' more random spammers out to the right. The hypothesis is that by taking sufficiently small iteration steps (removing the worker with the highest score and replenishing results before removing the next worker), RandomSep will enable the removal of random spammers at a minimal loss of ethical workers.

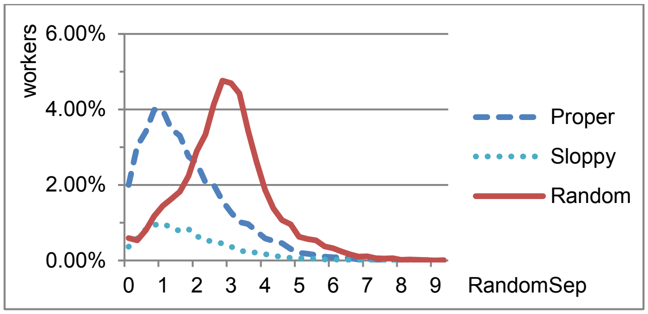

Figure 1. Worker distribution with RandomSep

3.5 Removal based on pattern frequency

Uniform spammers use fixed voting patterns, sometimes as simple as a single label. Against RandomSep they decrease the chance to get detected by choosing the middle label, as the average distance of error is decreased. RandomSep and gold sets are both less capable of detecting uniform spammers on a skewed label distribution, should they choose a more frequently occurring label (in relevance judgments for IR, there are often fewer documents 'totally relevant' than 'not relevant' to a query). Uniform spammers are also more likely to affect correct label prediction than random spammers, because they are more likely to coincide with other uniform spammers choosing the same label over a greater number of questions. This last effect could be counteracted by shuffling the questions performed by workers. The obvious characteristic of long repeating voting patterns can be used to filter out uniform spammers more effectively.

Kouritzin conducted a study on the detection of fake coin flip sequences [12]. The best detection algorithms focus on the variances between coin flips and the preceding flip sequence, which would add up to 0 for genuine coin flips. An analysis of uniform spammers confirmed long regular voting patterns to be repeated frequently. This characteristic is used as the basis for separation. There are however three important differences between vote-sequences and coin flip sequences: (1) Many of the possible voting sequences do not occur in the often short voting sets of crowdsourcing workers. (2) Voting sequences have corresponding correct answers. (3) The extent to which reoccurring patterns are suspicious depends on the amount of errors made in those patterns. We propose the *UniformSep* algorithm is, which is designed to identify uniform spammers based on reoccurring voting patterns. The errors made in reoccurring patterns are regarded as uniform spam errors.

$$UniformSep = \sum_{s \in S} \frac{|s|^2 \cdot (f-1)^2 \cdot \varepsilon^2}{\theta}$$

S is a collection of all possible n-tuples from available labels within a predefined range of lengths. To calculate the total amount of uniform error, all n-tuples $s \in S$ are matched within the ordered sequence of votes casted by the worker. For each s, |s| is the tuple length of s, f is the frequency of s within the casted votes, and ε is the number of casted votes within all matches of s that do not correspond with the estimated correct answer.

In this study, empirical testing showed the UniformSep algorithm to work best with LENGTHS = {2,..,5} and θ = 150·|VOTES|·|LENGTHS|, creating a stable threshold point regardless of average sequence length, worker quality or proportion of spam.

3.6 Qualifying on precision

In crowdsourcing, sloppy workers may have good intent, but produce poor judgments nevertheless [6]. After spam is removed, the worker pool may still contain sloppy workers that mostly disagree with the estimated correct answers, which becomes more visible in a logarithmic plot such as Figure 2. The spike at 0% precision is caused by rare occasions in which the last worker only works 1, 2 or 3 questions, increasing the chance of a worker failing on all questions.

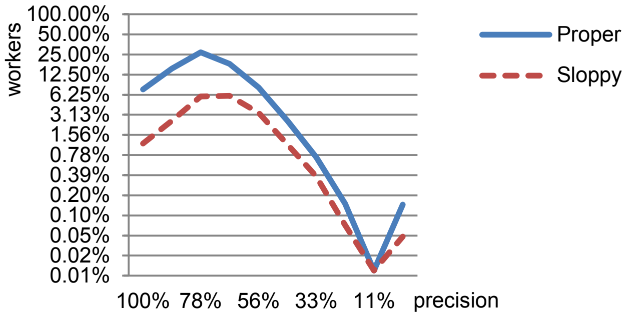

Figure 2. Worker precision

It depends on the task at hand whether a low score on precision should be considered poor judgment or disagreement. On interpretative tasks, different backgrounds, interests, and theoretical perspectives can lead to different interpretation of the data and to a substantial variation between annotators [13,14]. In general, asking for opinions or interpretations must hold more tolerance for disagreement than strict factual questions.

Judging the relevance of a document to a query is an interpretative task, subject to the annotators' frame of reference, ambiguity of language, etc. However, since relevance judgments for IR evaluation are usually considered to have a single best answer, systematic disagreement may indicate sloppy work, poor understanding of the data or incapability to perform the task. Removal of the most extreme underperformers seems justifiable in the same way the use of gold set questions is. However, it remains difficult to decide where to place the threshold.

4. EXPERIMENT SETUP

4.1 Crowdsourcing management

To compare gold sets and majority voting to the algorithms proposed in this paper, they will have to be tested over a large set of query-document pairs on a crowdsourcing platform. This is easy to do for consensus algorithms. However, the spam detection algorithms require votes by rejected workers to be replenished, to meet a predefined target result (e.g. 5 votes per query-document pair). The replacement of rejected workers with new workers may lead to more rejections, increasing the number of iteration cycles. This may easily surpass manual feasibility. CrowdFlower is a good example of a service that handles the crowdsourcing management, but cannot be used with user-defined algorithms. In order to use the algorithms proposed in this paper, a crowdsourcing management tool has to be created.

4.2 Simulation

As a pre-study, simulations were used to experiment with the designed algorithms, before eventually testing them in the real world. The simulation model used is based on three types of entities: units (query-document pairs), workers and votes. The simulations process the following steps:

(1) Units are created with a correct label chosen out of all possible labels, the required number of votes to obtain and a degree of difficulty, influencing the probability of an ethical worker making a correct judgment.

(2) As long as there are units that have not received the required number of votes, new workers are created. Upon creation, each worker is assigned to a worker class (Table 1), using a predefined worker class distribution over the whole population. The attributes for each worker are chosen from distributions, such as the maximum number of votes to cast and for ethical workers the ability (probability) to make correct judgments.

(3) The worker is requested to vote on a unit he has not previously voted on. Each worker continues to cast votes until he reaches his limit, or until there are no more units left for him to vote on.

(4) When all units have received the required number of votes, the configured algorithms, like gold sets, RandomSep and/or UniformSep, are executed. Workers that do not meet the requirements are rejected together with all of their votes. If at the end of this step there are units that do not have the required number of accepted votes, the simulator jumps back to step (2).

(5) A consensus algorithm, like majority voting, aggregates the votes cast per unit into one answer. Accuracy is calculated by comparing the consensus for a unit to the original correct label assigned to the unit.

The simulation model allows parameters, such as worker-class distribution, worker-ability distribution and minimum number of votes per query-document pair, to be changed.

4.3 Workers

Ethical workers are given a prior probability to answer a question correctly. This probability is distributed according to a (skewed) Gaussian distribution over the whole population. Ethical workers with a prior below .6 are labelled sloppy and above proper. As ethical workers were observed to mostly err close to the correct answer, the error distance from the correct label also follows a Gaussian distribution, decreasing the probability to vote further away from the correct label.

Random spammers pick a random label each time. Semi random spammers answer 40% of the question as an ethical worker and 60% as a random spammer.

Based on the analysis of crowdsourcing results, uniform spammers choose 2 labels (possibly the same). One label is used, having a 10% chance to switch labels after each vote and a 10% chance to put in a random label.

5. RESULTS OF COMPARISONS

5.1 Ideal world

To create a point of reference for future results an ideal world scenario was used, using only proper workers, to give an indication of the accuracy that can be obtained with given settings. To facilitate comparison, basic settings such as task difficulty and worker abilities were chosen to remain the same across all experiments.

For this ideal world scenario, ethical workers were generated with a mean ability of .65 of answering a question correctly. This scenario contained no spam and all workers with an ability less than .60 were removed prior to simulation. This results in an accuracy of 84% when majority voting is used. The consensus obtained with EM (83.7%) is significantly (one-tailed t-test, α = .01) less accurate than majority voting in this scenario.

5.2 Impact of spam on consensus

The hypothesis was that EM (as described in Section 3.2) provides more accurate results than using the majority vote, when spam is present. This was tested on populations with different percentages of spam. Between spammers the distribution used was 40% random workers, 20% semi-random workers and 40% uniform spammers. The workers that were not spammers were ethical workers with a mean probability of .65 to judge a query-document pair correctly. This population is referred to as *mixed spam* and used in other experiments as well.

Figure 3 shows both consensus algorithms to be affected when spam increases. There is a crossover in accuracy around 60% mixed spam. Separate paired t-tests (α = .01) showed that EM is significantly more accurate when mixed spam is 60% or less and majority voting is significantly more accurate when spam is more than 60%.

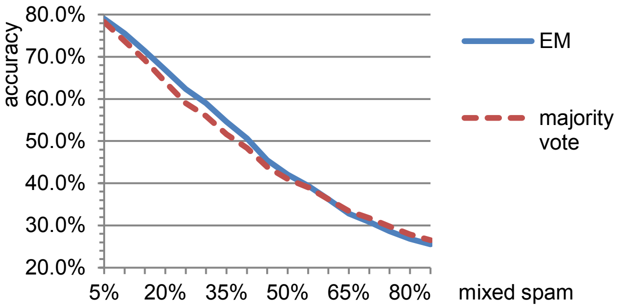

Figure 3. Consensus

The hypothesis that EM provides more accurate consensus when spam is present did not prove correct. Interestingly, it is majority voting that gives more accurate results when spam is abundant. Experiments with different mixes of workers showed EM to perform better than majority voting when the population contains more uniform spammers or sloppy workers. This can be explained by the fact that both types often fail with the same combination of correct label and given label, allowing EM to decrease the weight of those combinations.

5.3 Combined consensus

A detailed analysis of the simulation results that were generated comparing majority voting and EM (Section 5.2), revealed majority voting to tie on 12% of the questions, using 5 votes per question. If a decision is forced when majority voting ties, accuracy will not exceed 50% if no other evidence is used. This indicates that in cases where majority voting does not tie, majority voting is likely to outperform EM. We propose to combine the two algorithms, favoring majority voting and using EM when votes tie. The hypothesis is that this *combined consensus* algorithm is more accurate than majority voting and EM.

The combined consensus was compared to the best of majority voting and EM on a mixed spam population. Combined consensus gave significantly better accuracy across all spam percentages (paired t-test, α = .01): 0.2% better than majority voting when 85% is spam, and 0.9% better than EM when 5% is spam.

5.4 Removal of random spammers

To remove random spammers we proposed the RandomSep algorithm in Section 3.4. The hypothesis is that RandomSep can remove random spammers more effectively than gold sets. The comparison was done over a mixed spam population with different proportions of spam. When gold sets were used, 30% gold set questions were injected into the set, of which 50% had to be answered correctly.

The results in Figure 4 show that RandomSep detects close to all random spammers. RandomSep also removes other types of spammers when the proportion of spam does not exceed 55%. However, if uniform spammers are abundant, they are more likely to coincide with other uniform spammers, increasing the chance to avoid detection by affecting the consensus that RandomSep depends upon.

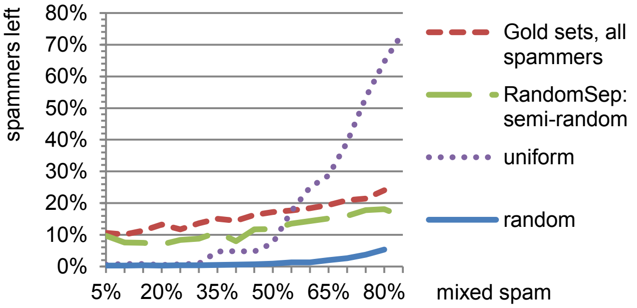

Figure 4. Remaining Random spammers

To rule out the possibility that the better algorithm simply removed more workers in total, the percentage of removed proper workers is given in Figure 5. On a population with up to 50% spam, RandomSep removes fewer proper workers, showing that removal of more spam is not caused by removing more workers overall.

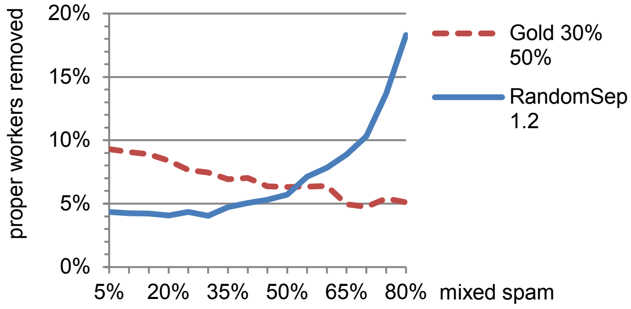

Figure 5. Proper workers removed

Comparing the accuracy after the use of gold sets and RandomSep, it appears that removing more spam while retaining more proper workers improves accuracy by up to 2% in Figure 6. However, beyond 50% mixed spam the accuracy of RandomSep drops, being incapable of dealing with an abundance of uniform spammers.

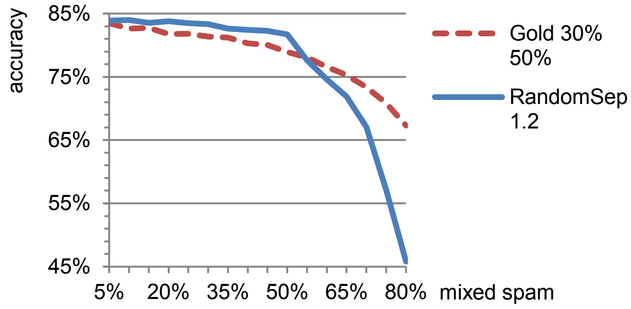

Figure 6. Accuracy between gold sets vs RandomSep

Note that this research did not study the full extent of different configurations for both algorithms. For instance, the use of more gold set questions is assumed to improve accuracy at greater monetary costs.

5.5 Removal of uniform spammers

In Figure 4 we already presented the extent to which RandomSep detects uniform spammers. Although RandomSep gives good results when less than 15% of the population is a uniform spammer, RandomSep progressively fails detecting them when there are more. Figure 4 also gives the average detection of spammers by gold sets. The hypothesis is that UniformSep can remove uniform spammers more effectively than RandomSep and gold sets. The performance of UniformSep was measured over a mixed spam population. The results in Figure 7 confirm the ability of UniformSep to remove uniform spammers. It also shows that UniformSep is incapable of removing all spammers.

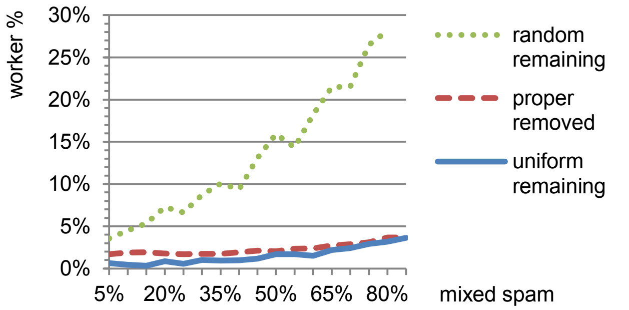

Figure 7. UniformSep

In this test, UniformSep on average removed 2% of the proper workers. Other experiments with 50% spam in the population showed that UniformSep can also be used with a more tolerant threshold, resulting in the removal of 95% of the uniform spammers while only incidentally (<0.1%) removing a proper worker.

5.6 Removal of sloppy workers

As proposed in Section 3.6, we hypothesize that removing ethical workers with poor precision (have most questions wrong) improves accuracy. However, because the precision of each worker is estimated using the consensus amongst all workers, spammers have to be removed in order to minimize bad predictions. Removal of spammers was done using RandomSep on a 50% mixed spam population. The precision algorithm uses a fixed value as the required minimum precision, and rejects all workers that do not meet this requirement. The algorithm follows the same iteration scheme that RandomSep and UniformSep use; eliminate the worker with the lowest precision first and replenish before eliminating the next. In case a replenished worker is a spammer, after each cycle RandomSep is checked with a higher priority than the precision algorithm. The precision algorithm is only executed when RandomSep is done removing workers.

Figure 8 shows that accuracy on the left y-axis increases when workers below a fixed *precision threshold* on the x-axis are removed. However, using a precision threshold also rejects more proper workers, resulting in more votes needed. It seems that no general threshold can be given for precision; the tradeoff between accuracy, costs and tolerance for difference in interpretation, is case-dependent.

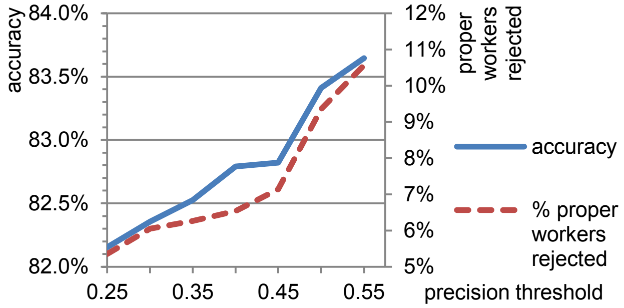

Figure 8. Qualifying on precision

6. THE NEXT LEVEL

6.1 Combining RandomSep and UniformSep

RandomSep and UniformSep are two spam detection algorithms designed for specific types of workers. RandomSep appeared incapable of handling abundant uniform spammers (as shown in Figure 4), whereas UniformSep performs more consistently against uniform spammers (as shown in Figure 7). The hypothesis is that the two algorithms combined can detect spammers more effectively.

This test was carried out on a mixed spam population with UniformSep being executed before RandomSep. The details of this test show that fewer than 2% of the random and uniform spammers got detected when 85% of the workers are spammers. The results in Figure 9 show that UniformSep counteracts the negative effect of abundant uniform spammers on RandomSep, resulting in consistent accuracy.

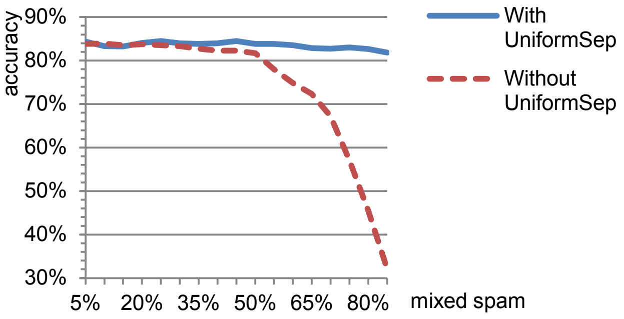

Figure 9. Combining RandomSep and UniformSep

6.2 Selectively obtaining more evidence

An analysis of simulation results showed frequent disagreements between EM and the majority vote, for example a mean probability of .65 amongst ethical workers to judge query-document pairs correctly, results in 15% disagreement between these consensus algorithms. The idea of selective repeated-labeling [15] inspired us to obtain more evidence when majority voting and EM do not agree on the outcome. The hypothesis is that gathering more votes on questions where the consensus algorithms do not agree, will increase accuracy.

The *Resolve disagreement* algorithm compares the outcome of majority voting and EM. For each question the initial number of votes to obtain was set to 5, which is increased for questions where the two consensus algorithms do not agree. The maximum number of votes for any question is given as a parameter to restrict the number of iterations. The test was performed on a mixed spam population with 50% ethical workers. Spam was removed with RandomSep before applying Resolve disagreement.

The results of the test show that accuracy is likely to increase when more votes are obtained for questions on which the two consensus algorithms disagree (Figure 10). If 3 more labels are allowed, accuracy was increased by 3.5%, exceeding the accuracy obtained in the ideal world scenario (Section 5.1). Obtaining these extra labels required 4% more votes on the total amount of obtained votes.

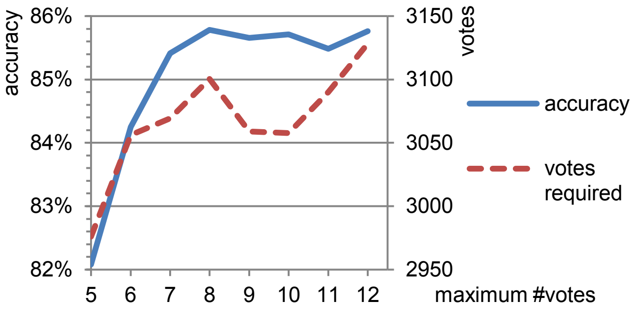

Figure 10. Resolving disagreement

6.3 How much spam can you take?

The final test reveals the difference in accuracy between the commonly used combination of gold set questions and majority voting, and the combination of algorithms presented in this paper. The test was performed on a mixed spam population. Ethical workers were drawn from a distribution with a mean ability of .65 of voting the correct out of 5 labels. 200 query-document pairs were judged by a minimum of 5 workers. The concrete parameters used in this experiment can be found in Appendix A.

The results in Figure 11 reveal that gold sets and majority voting are more susceptible to spam. When 50% of the workers are spammers, the proposed algorithms improve accuracy by 9% over gold sets and majority voting.

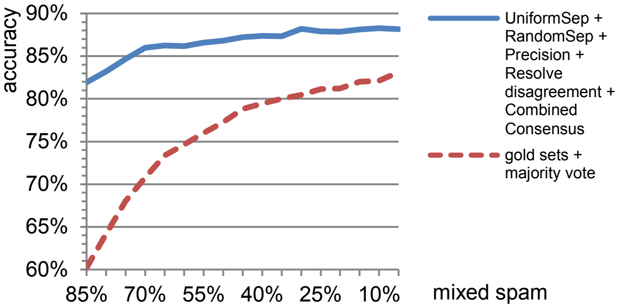

Figure 11. How much spam can you take?

Figure 12 shows the average number of votes obtained per query-document pair. Initially 5 votes per query document pair were required, and additional extra votes were required to replenish the votes of rejected workers, for resolving disagreement and for gold set questions. The proposed algorithms require fewer votes, except when spam becomes more than 80%.

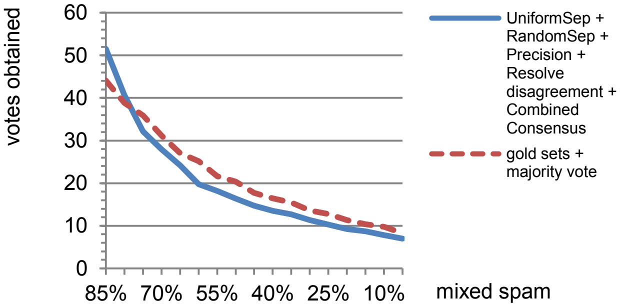

Figure 12. Average votes obtained per query-document pair

7. DISCUSSION

Crowdsourcing can be used to obtain quality relevance judgments. However, large amounts of spam may have to be dealt with. In this study, crowdsourcing results were analyzed to classify workers based on characteristics. These characteristics were used to select theories and design algorithms to separate the workers from different classes.

Majority voting was compared to a consensus were votes are weighted by worker quality, determined by an EM algorithm. EM only outperformed majority votes when uniform spammers or sloppy workers were present and less than 60% of the workers are spammers. However, a combined consensus algorithm, favoring the majority vote and using EM when tied, consistently gave the most accurate predictions of relevance judgments.

Increasing the quality of the workforce, by removing spammers and workers with poor precision, increases accuracy of relevance judgments. The algorithms presented in this paper outperformed gold sets in filtering out spammers and workers with poor precision, while rejecting fewer proper workers and needing fewer votes overall. Results were further improved by obtaining more votes on undecided questions. The combination of gold sets and majority voting and the combination of our proposed algorithms were compared over different proportions of spam. The combination of gold sets and majority voting is susceptible to spam, giving poor results when spam is abundant. The algorithms proposed in this paper show to be potentially more spam-resistant, consistently outperforming gold sets and majority voting. On a population containing 50% spam, the combined algorithms proposed in this paper provided 9% more accurate results.

A simulation was used to compare the proposed algorithms with the commonly used gold sets and majority voting. Simulation proved to be a fast and cheap instrument for a pre-study on the effect of spam on different algorithms. There is however always a concern of simulations not sufficiently reflecting the real situation.

As a prelude to real world tests, the crowdsourcing results already mentioned in Section 2.2 were obtained to verify the classification in Table 1, as well as for testing the proposed algorithms on real data. Table 1 reports per worker type on the proportion of the population and the measured average precision of their work. Out of a total of 33 workers, RandomSep detected 12 spammers and UniformSep detected 9 spammers, overlapping on 8 workers and resulting in 13 workers getting rejected. The 13 rejected workers have an average precision of .31, while the 20 accepted workers have an average precision of .72. This indicates the removed workers are most likely spammers and the accepted workers are not. The average RandomSep score for rejected workers was 2.6 and for accepted workers 0.6, supporting the hypothesis that ethical workers indeed err closer to the correct answer than spammers. Comparing the votes of accepted workers with the estimated correct relevance judgment, we found 70% of the votes to agree with the consensus.

8. FUTURE WORK

To support the outcome of this study, the algorithms will have to be further tested in real world crowdsourcing experiments. To prepare real world testing the proposed algorithms have been normalized to give comparable outcomes with the same parameters in different situations (Appendix A).

For real world testing, a crowdsourcing management tool has to be constructed. Several algorithms depend on small iteration cycles to get the best results. As the number of cycles can easily exceed 50, manual management of crowdsourcing activities does not seem feasible. A crowdsourcing management tool can automate the publishing of HITs, fetching and processing of results and iterate until all requirements are met. This can be fully automated by using the AMT API.

9. 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] O. Alonso and S. Mizzaro. Can we get rid of TREC assessors? Using Mechanical Turk for relevance assessment. In *Proceedings of SIGIR 2009 Workshop on the Future of IR Evaluation*. pages 15–16. 2009.

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

[5] 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.

[6] 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.

[7] J. S. Downs, M. B. Holbrook, S. Sheng and L. F. Cranor. Are your participants gaming the system?: screening mechanical turk workers. in *Proceedings of CHI 2010*. pages 2399-2402. 2010.

[8] J. Ross, L. Irani, M. Silberman, A. Zaldivar and B. Tomlinson. Who are the crowdworkers?: shifting demographics in mechanical turk. In *Proceedings of CHI 2010*. pages 2863-2872. 2010.

[9] C. Callison-Burch. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of EMNLP 2009*. pages 286-295. 2009.

[10] P. G. Ipeirotis, F. Provost and J. Wang. Quality management on amazon mechanical turk. In *KDD-HCOMP 2010*. pages 64-67. 2010.

[11] A. Dawid and A. Skene. "Maximum likelihood estimation of observer error-rates using the EM algorithm." Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1): 20-28. 1979.

[12] M. A. Kouritzin, F. Newton, S. Orsten and D. C. Wilson. On Detecting Fake Coin Flip Sequences. In *Markov Processes and Related Topics*. pages. 2006.

[13] K. Krippendorff and M. A. Bock. The content analysis reader. Sage Publications, Inc. 2009.

[14] S. Nowak and S. Rüger. How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In *Proceedings of MIR 2010*. pages 557-566. 2010.

[15] V. S. Sheng, F. Provost and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD 2008*. pages 614-622. 2008.

APPENDIX A. PARAMETERS

In real world use, the distribution of worker types and worker attributes are unknown. The parameters for the proposed algorithms should therefore be as environment-independent as possible. The algorithms to which this applies were normalized to give comparable outcomes with the same parameters over different situations. This enables a set of parameters, with an agreed upon relative tradeoff between costs and accuracy, to be used regardless of the distributions of worker types, as seen in Figure 11.

The proposed algorithms did not appear very sensitive to changes in parameters when the population consists of 70% spam or less, making the algorithms suitable to use in field studies. However, simulation did show the algorithms to become more sensitive to parameter changes when spam exceeds 75%.

For reproduction purpose, the parameters used for Figures 11 and 12 were:

- Gold set: 30% injected, 50% required correct answers.
- UniformSep: maximum allowed = 1
- RandomSep: maximum allowed = 1.2
- PrecisionThreshold: minimum precision = 0.4
- Resolve disagreement: max. votes when no consensus = 8