The Sixth AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2018)

All That Glitters Is Gold — An Attack Scheme on Gold Questions in Crowdsourcing

Alessandro Checco
University of Sheffield, UK
a.checco@sheffield.ac.uk

Jo Bates
University of Sheffield, UK
jo.bates@sheffield.ac.uk

Gianluca Demartini
University of Queensland, Australia
g.demartini@uq.edu.au

Abstract

One of the most popular quality assurance mechanisms in paid micro-task crowdsourcing is based on gold questions: the use of a small set of tasks of which the requester knows the correct answer and, thus, is able to directly assess crowd work quality. In this paper, we show that such mechanism is prone to an attack carried out by a group of colluding crowd workers that is easy to implement and deploy: the inherent size limit of the gold set can be exploited by building an inferential system to detect which parts of the job are more likely to be gold questions. The described attack is robust to various forms of randomisation and programmatic generation of gold questions. We present the architecture of the proposed system, composed of a browser plug-in and an external server used to share information, and briefly introduce its potential evolution to a decentralised implementation. We implement and experimentally validate the gold detection system, using real-world data from a popular crowdsourcing platform. Finally, we discuss the economic and sociological implications of this kind of attack.

1 Introduction

Crowdsourcing is a growing solution to perform human computation and to collect manual annotations, especially useful in case of large-scale data and complex labelling tasks on which machine-based algorithms still struggle.

Crowdsourcing has the capability of achieving high quality labelling, but it requires specific quality assurance mechanisms to deal with potential scammers interested in the monetary reward attached to the tasks, incompetent workers, and loss of attention (Daniel et al. 2018).

Many solutions that deal with low-quality contributions in crowdsourcing have been proposed so far. For example, Snow et al. (2008) propose a bias-correction and averaging scheme to improve annotation quality. Ipeirotis, Provost, and Wang (2010) use EM procedures in bias correction to detect scammers. In the case of subjective tasks, Kittur, Chi, and Suh (2008) propose the injection of verifiable questions, arguing that they be included into otherwise subjective tasks and answering the verifiable part correctly should take as much effort as doing the whole task. Dow et al. (2011) introduce peer-to-peer feedback systems to encourage worker engagement and high-quality work. Gadiraju et al. (2015) analyse the behavioural patterns of microtask workers to differentiate trustworthy and untrustworthy workers.

The most commonly used technique for quality assurance in crowdsourcing is the use of *gold questions*: a small set of questions with known ground truth answers (Le et al. 2010; Huang and Fu 2013) which are used to validate the accuracy of crowd answers. Such set should have specific characteristics (Oleson et al. 2011):

1. Relative small size to minimise its creation cost as correct answers are typically created by expert editors. Moreover, crowd workers need to be paid when answering such questions that do not bring new information to the job.
2. Limited (or absent) repeated exposure of gold questions to minimise the likelihood a worker will recognise them.
3. Objective, non-ambiguous true answers.
4. Even distribution amongst the possible answers in the case of multiple-choice questions.
5. Semantically and structurally similar to the non-gold questions to minimise the possibility of crowd workers recognising them.

Point 1 is in direct contrast with points 2–5: generating a collection of gold data with such requirements is costly because it needs to be tailored to the specific crowdsourcing job and it needs to have a relatively large size (Bentivogli et al. 2011). In some cases, it is possible to reduce this cost by generating gold sets in a programmatic way (Oleson et al. 2011). In any case, building a gold set requires a compromise between points 1 and 2: there is an inherent trade-off between the size of the gold set and the cost of performing gold questions in a batch. Thus we can make the following fundamental assumption:

Assumption 1 The size of the gold set is notably smaller than the size of the set of non-gold questions.

In this paper, we show that this inherent limit on the size of the gold set can be exploited to perform an attack to the crowdsourcing platform: workers can collude by using an inferential system to detect which questions are likely to be gold questions.

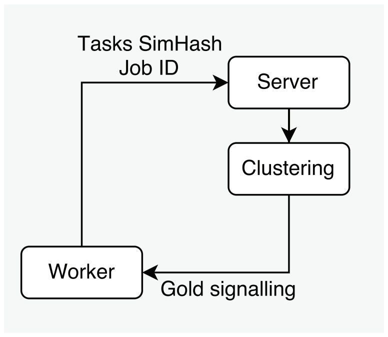


Figure 1: Collaborative gold signalling structure.

Such a system should have the following capabilities:

1. Ability to signal the likelihood for a task of being a *gold question*[1].
2. Anonymous (the worker is not identified by the system) and secure (the content of the tasks is not circulated).
3. Ability to be used on any crowdsourcing platform where the requester checks worker quality with a gold set considerably smaller than the set of tasks to be evaluated, and even if the gold questions are dynamically generated, e.g., 'solve the following arithmetic equation', 'answer the captcha', or more complicated solutions like the ones presented in (Oleson et al. 2011).
4. Ability to support workers even in the relatively sporadic cases when quality assurance mechanism can use intrinsic metrics like transitivity checks (if $A > B$ and $B > C$ then $A > C$) (Buchholz and Latorre 2011).

To simplify the analysis, we will also make the following assumption, that is satisfied in many crowdsourcing platforms (as we verify in Section 4):

Assumption 2 Gold questions are shown to the worker sampling uniformly at random from the gold set, with the additional constraint each gold question can be shown only once to each worker.

2 Proposed System

We now introduce the system able to perform the described attack scheme to crowdsourcing quality checks using gold questions. We consider the case of a batch of crowdsourcing tasks (job). A subset of the workers involved in this job are assumed to collude to attack the platform. In Figure 1, a sketch of the interaction between workers and a third party server, external to the crowdsourcing platform, is shown. The basic structure and usage are simple: each crowd worker runs a local browser plugin (or a javascript bookmarklet) where a set of operations are executed to create a fingerprint of the tasks currently displayed in the browser. Then, the pair (job ID, page hashes) is sent to the external server. The server will adopt an inference technique (explained in Section 2.2) to update the information available for that job, and to signal back to the worker the likelihood of each of the current tasks being a gold question.

[1]The system is not aiming at providing the actual answer of a signalled gold question, but just to identify them so that workers can focus on answering them accurately.

2.1 Client Workflow - Simhash

Figure 2 shows the pipeline of the operations executed by the browser plugin of each colluding worker. Every time the webpage Document Object Model (DOM) is updated, the following operations are executed:

Anonymization The worker ID and other identifying information are stripped out from the webpage. This is necessary to guarantee plausible deniability, to protect from watermarking, and to improve the possibility of identifying similar gold questions.

ID/name tags removal Session IDs and name tags are stripped out from the webpage (excluding tags containing attributes like "href" and "src", to preserve the ability to distinguish tasks).

Task splitting The page is split into different fragments, one for each task (see Section 2.3) if needed.

Shingle/tokenization The HTML of each fragment is tokenized with classical 3-gram techniques, where the words are either HTML tags or text content: in the latter case the text is shingled (Damashek 1995).

Simhashing For each task (fragment) a simhash (Sadowski and Levin 2007) is generated.

Using simhashes is the ideal solution for our problem because: (i) it allows a secure export of the task fingerprints, without the risk of leaking the task online (as explained in Section 2.5), (ii) it is fast and scalable, (iii) it allows to estimate similarity (by simply comparing the Manhattan distance of the simhashes) even for near-miss cases, enabling the system to recognise gold questions that can differ by a small part, e.g., captchas, arithmetic questions, or programmatic gold (Oleson et al. 2011).

2.2 Server Workflow - Clustering

The server, described in the previous section, keeps a repository of triples (Job ID; simhash; multiplicity), where multiplicity is the number of times a simhash appears in the collected data. The Manhattan distance matrix between the bit representation of the simhashes can be used to generate a clustering. Detecting similarity rather than exact matches is important because of the potential presence of noise in the task HTML collected from workers (i. e. differing fragments due to dynamic rendering), and because gold questions whose hashes differ for only few bits (like captchas and arithmetic questions) belong to the same cluster in our framework even if they are not exact matches. After this process, each cluster will represent a specific question, and have a multiplicity that is equal to the number of times that question has been posed to the participating workers. We will initially assume that all workers are colluding. In Section 4.2, we discuss the effect of the number of colluding workers on the attack performance.

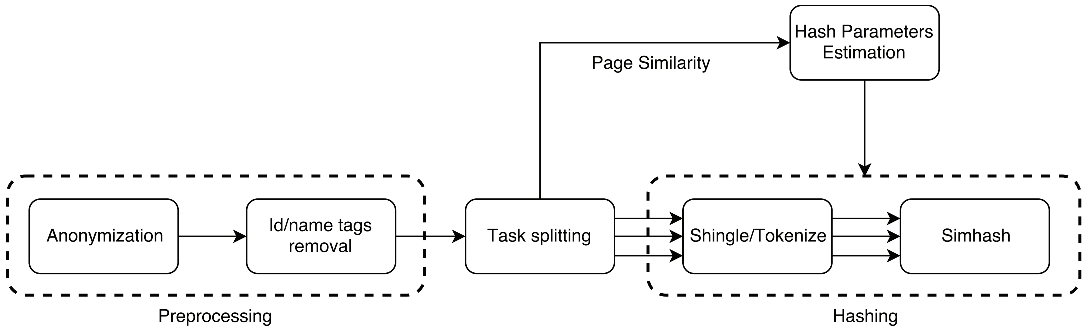


Figure 2: Workflow of the hashing mechanism on the crowd worker side.

Gold Questions Inference. If Assumptions 1 and 2 are satisfied, we expect that the multiplicities of clusters will have a bimodal distribution, where clusters corresponding to gold questions will have a higher mean multiplicity, as shown for a real case in Section 4, Figure 6. A simple Gaussian mixture model with two modes will be enough to obtain a classification of the current state of the repository of simhashes, together with an estimate of the confidence of the classification accuracy. The plugin has two states:

Idle state: If the model goodness of fit is low, the plugin shows that there is not enough information: any question could be gold.

Active state: When the goodness of fit is high, the plugin signals which questions in the page are likely to be gold, together with a probability score for each of them.

When only few samples are provided, overfitting can be a problem: since we have only one dimension d (frequency) for each cluster of simhashes, and two modes p, we have six parameters to estimate ($p(d^2/2 + 3d/2 + 1)$), and thus we have to keep the plugin idle for a number of samples of five times (Steyerberg, Harrell, and Frank 2003) the parameter size, i. e. 30 samples. After that, the Bayesian Information Criterion (BIC) of the model compared with the corresponding one-component model will be used to establish the state of the plugin (Fraley and Raftery 1998). The performance of the Gaussian mixture model (and thus the plugin state) depends on the difference between the means of the two distributions (and in some cases the two means can be very close, as shown in Section 4, Figure 6). However, when the plugin is active it means that a two-components Gaussian mixture model explains the data better than a fit obtained by a unique population, and for each question the worker is able to visualise the posterior probability of being a gold question.

If the job presents regular patterns, like one gold question per page, then the worker can decide to employ a more aggressive behaviour, by answering carefully only to the questions that are signalled as gold.

The plugin does not need to know the proportion of gold questions used, nor the proportion of colluding workers.

False Positives and Sensitivity. It is worth noticing that there is a risk that non-gold questions are simhash similar, with the consequence of having some of them ending up in the same cluster and thus causing some false positives. However, this event can be detected and corrected when a page contains multiple tasks, as shown in Section 2.3.

Due to the nature of the application, high recall is more important than high precision in gold question detection: from a worker perspective, false positives will lead to additional work, but missing a gold question can potentially disrupt a worker quality score (e.g., approval rate) in the crowdsourcing platform. The server will return a probability of being a gold question for each simhash submitted by a worker. The user is then able, via the browser plugin, to select the desired confidence threshold for the tasks being signalled setting their own precision/recall trade-off.

2.3 Multiple Tasks per Page

The operation of task splitting explained in Section 2.1 is not straightforward. It can be achieved in at least two ways:

1. Platform-based heuristic (e.g., Crowdflower uses a specific HTML class element to identify tasks in a page).
2. Heuristic based on document size: this affects the balance between precision and recall.

In our experiments, which make use of Crowdflower (now called Figure Eight) datasets, we use the former approach. If this is not possible, a more conservative solution (with more fragments) can be used to maximise recall: for example, a very conservative solution could be to split the page at the <div> tag level.

If multiple tasks appear in the same page, the server will be able to use this information to perform a *hash parameter estimation*: the minimum distance between the simhashes belonging to the same page can be used to tune the clustering algorithm and avoid that two different questions end on the same cluster. Moreover, if the clustering algorithm is able to use connectivity constraints, they can be enforced for all simhashes known to be in the same page. It is possible to obtain the same information even when only one task per page is used, by moving the parameter estimation on the client side.

2.4 Peer-to-peer Implementation

The use of an external server where to centrally collect data and perform the similarity computations can be avoided, if required. All operations of clustering and inference are lightweight and could potentially be run locally by the worker in the browser: what is needed is at least an append-only distributed peer-to-peer database system, e.g., OrbitDB (https://github.com/orbitdb/orbit-db), to collectively store and retrieve the triples (Job ID; simhash; multiplicity) and locally compute the probability of a task being a gold question.

2.5 Plausible Deniability and Data Security

Crowd workers are sending only a simhash of the page with identification information removed on the client side. Moreover, they are not sending any information about the actual judgements being performed. This is an important aspect that allows to minimise the risk of worker identification through de-anonymisation (Aggarwal 2005), and thus obtaining plausible deniability for the workers. Moreover, potential sensitive data in the job cannot be reconstructed, even when the third party server is completely compromised.

3 Attack Model - Performance, Cold Start

To understand whether our framework is applicable in a real crowdsourcing platform, we devise a model that allows us to compute the probability of recognising a gold question. Such model assumes that the parameters of the system are known and that the report of a specific question will have exactly the same simhash from all workers (so, in this theoretical model, we do not consider the cases of noisy HTML or gold questions generated programmatically). In order to obtain a closed form solution of the average probability of recognition, we consider a gold question recognised by the system only when it has been already reported a number of times bigger than the (known) multiplicity of the non-gold questions. Clearly, this under-estimates the probability of recognition, because as we will show in the next section, a statistical analysis between the multiplicities can be enough to recognise a gold question. Moreover, such simple model does not allow to estimate the false positive rate, that will be studied in Section 4 over real data, but still it gives us a quick and clean way to explore the effect of the different parameters on the gold recognition probability.

Regarding the parameters of the system, we consider a realistic scenario: a job of 2000 tasks with an additional 5% (100 tasks) of gold questions.

We consider the default automatic behaviour of Crowdflower: 10 gold questions are used at the beginning to train and test the ability of the worker (i.e., a quiz page). After that, pages of 10 tasks are shown to the worker, of which 9 are requested tasks and one is a gold question. To be considered trusted, workers are required, by default, to judge a minimum of four gold questions and to reach an accuracy threshold of 70%. A similar setting can be implemented on MTurk by creating qualification tests and by manually distributing gold questions in subsequent HITs.

Moreover, we consider the default Crowdflower aggregation setting: each requested (non-gold) task will be shown to 3 distinct workers. A gold question will not be shown twice to the same worker: thus, for our setting, a maximum of 9 pages can be shown to each worker, with a total (including the initial quiz page) of 100 unique gold questions per

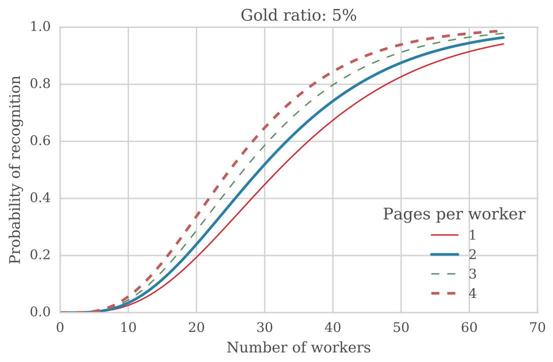


Figure 3: Probability of recognizing a gold question varying the number of workers that have already used the system and parametrized by the number of 10-task pages.

worker. Clearly, different workers may be required to evaluate the same gold questions. In Figure 3, we estimate the probability that a gold question displayed to a worker is correctly detected by our system, after that a certain number of workers already used the system for a specific job. This number depends on how many tasks each worker will complete on average. Even in the most conservative case (each worker completing only 10 tasks) and assuming a uniform at random distribution of gold questions, we can observe that after 50 workers entered the job, the probability of correctly detecting a gold question is above 99.9%. In Figure 4 we can see a drastic fall in the probability of recognition when the number of gold question increases, especially when a small number of more prolific workers are contributing: if the average worker is completing 30 questions and more than 16% of gold questions are available, the probability of recognition is below 20% even when 25% of the total work has been already completed. It is worth noticing that in this simplified model we assumed that all workers judge the same number of pages. In reality, the typical engagement distribution has a power-law distribution: we refer to Section 4 for a more realistic analysis.

These results are promising but are based on the simplifying assumptions of the model. For this reason, in the next section we present the result of implementing and testing our system over a real-world case study.

4 Experimental Analysis

4.1 Experimental Setting

To evaluate the effectiveness of the proposed attack scheme, we simulate the attack over two real crowdsourcing experiments. We use the CSTA datasets and task logs described in (Benoit et al. 2016)[2]. An example of the task design is shown in Figure 5. We will start with the first dataset, consisting of 29,594 judgements from 336 workers and containing the platform logs for the submitted judgements, includ-

[2]We used the jobs in the repository with ID f269506 and f354285, available from https://github.com/kbenoit/CSTA-APSR.

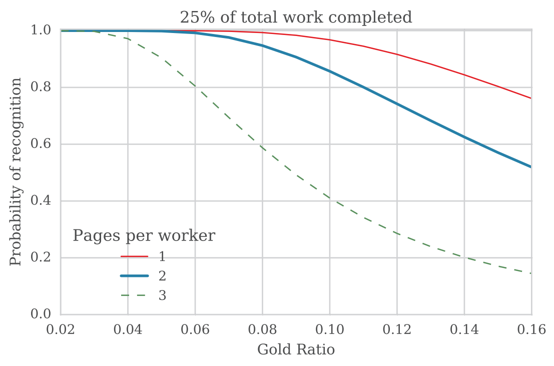
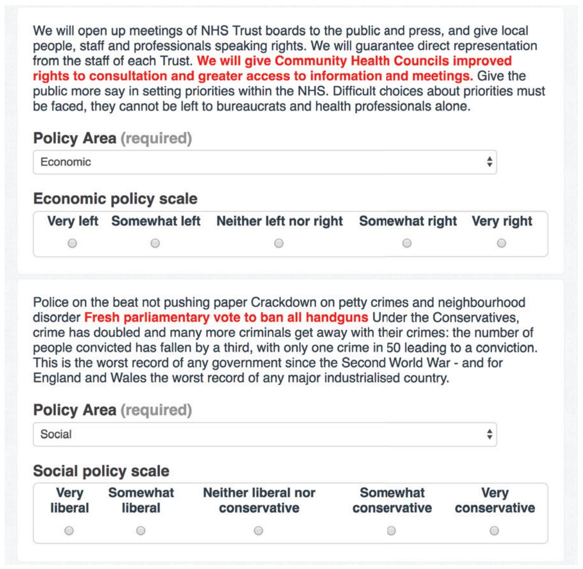


Figure 4: Probability of recognizing a gold question after 25% of the total work, varying the gold questions ratio and parametrized by the number of 10 tasks pages that each worker evaluate.


We will open up meetings of NHS Trust boards to the public and press, and give local people, staff and professionals speaking rights. We will guarantee direct representation from the staff of each Trust. **We will give Community Health Councils improved rights to consultation and greater access to information and meetings.** Give the public more say in setting priorities within the NHS. Difficult choices about priorities must be faced, they cannot be left to bureaucrats and health professionals alone.

Policy Area (required)

Economic

Economic policy scale

Very left	Somewhat left	Neither left nor right	Somewhat right	Very right
○	○	○	○	○

Police on the beat not pushing paper Crackdown on petty crimes and neighbourhood disorder **Fresh parliamentary vote to ban all handguns** Under the Conservatives, crime has doubled and many more criminals get away with their crimes: the number of people convicted has fallen by a third, with only one crime in 50 leading to a conviction. This is the worst record of any government since the Second World War - and for England and Wales the worst record of any major industrialised country.

Policy Area (required)

Social

Social policy scale

Very liberal	Somewhat liberal	Neither liberal nor conservative	Somewhat conservative	Very conservative
○	○	○	○	○

Figure 5: Design of the CSTA task. Workers have to select the most appropriate policy area related to a specific sentence presented to them in its context and the political scale (left/right) where it positions itself.

ing timestamps of each question and whether a gold question has been missed.

In the first CSTA dataset, out of 2700 unique questions, 12.4% of them are gold questions. We selected this dataset because of the unusual abundant number of gold questions, and because each non-gold question have been answered by 10 workers with an average of 8.7 pages per worker, making this example a particularly difficult case for this kind of attack, as predicted by our model and shown in Figure 4. Such an abundance of judgements and gold questions also allows us to simply sub-sample these two parameters keeping the rest of the log unchanged so that we can simulate what would have happened if fewer gold questions or judgements per question were to be used. Moreover, as shown in Figure 5, the gold questions used in this experiment are indistinguishable from the non-gold questions in terms of design and structure: being able to distinguish them is thus only possible through the statistical analysis of their multiplicity. At the end of this section we will show the result for the second dataset, consisting of 76,183 judgements and with a less challenging, more realistic distribution of gold questions. In Figure 6, the distribution (in logarithmic scale) of the multiplicity for gold and non-gold questions is shown: we can see that there is a clear indication of a bimodal distribution for the count of gold and non-gold questions, even if the two distributions overlap, making this dataset a good candidate to understand the limits of our framework. However, the main unknown of this approach is the behaviour of the system in the transient phase of the batch, that is, when many gold questions still have a multiplicity similar to that of non-gold questions (because of the high statistical variability when only a few workers have started the job and shared their tasks). This transient phase is when false positives are more likely to happen. Ideally, the plugin should be in idle state during this transient phase.

Regarding the clustering phase to identify simhashes belonging to the same question, this job did not present significant difficulties, because all simhashes belonging to the same questions had a Manhattan distance of less than 2 bits, even though they were reported by different workers with potentially different DOM.

In order to avoid disrupting a real crowdsourcing job, we decided to run the plugin on the reconstructed HTML obtained from the logs, together with the crowd worker original behaviour. The original designer of this job opted for having an initial quiz of 10 questions, 8 of which being gold, and after that to present pages of 10 questions, one of which being a gold question.

To study how the different parameters affect the proposed system, we keep the behaviour and time evolution of the worker fixed, but we vary the number of gold questions available via sub-sampling (i.e., allowing us to move from 0% to 12.4% of gold questions in the job), assuming that the worker ability to answer a gold-question is only dependent on their internal state, and not on which gold question they are being shown[3].

We are able to compare the original behaviour of the workers with the behaviour they would have by using the proposed collaborative gold signalling technique.

To simplify the analysis, we assume the following behaviour for the workers when the plugin is in *active* state:

Time spent: The worker will spend time in answering only the questions that are signalled as potential gold, spending an amount of time per page equal to $g \cdot T$, where g is the number of signalled gold questions in that page, and T is the time they spent on that page from the log.

Performance: The worker will *answer randomly* to any gold-question that the plugin failed to signal (false negative). On the other hand, the workers will answer any

[3]The error made using this assumption should be mitigated by the fact that all gold questions are sampled uniformly at random.

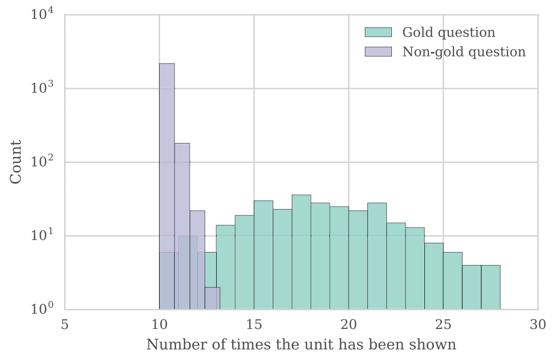


Figure 6: Distribution (log scale) of the multiplicity for gold and non-gold questions in the CSTA task when using 12.4% of gold questions. Assumptions 1 and 2 are satisfied.

correctly signalled question normally (i.e., in the same way they did in the logs, still potentially answering them wrong even when correctly signalled). This is a worst-case analysis: in practice we can expect workers to answer more carefully to signalled questions.

Confidence: The worker will consider as gold all questions with signalled probability of being gold of at least 50%.

This simulation setup guarantees that the individual accuracy of each worker is preserved: indeed whenever the plugin is *idle*, we will use the original performance reconstructed from the logs. We can notice that the likelihood threshold to receive the signals could be modified to reduce the false negative rate, at the expense of more false positives: thus, there is a trade-off between the time saved by the worker and the performance loss. We did not perform such optimisation and leave this aspect for a future study. However, it is worth noticing that the worker is able to set up their level of risk locally in the plugin, because the Gaussian mixture model is able to provide a probability of classification of each question provided, that can be then filtered on the client side.

4.2 Experimental Results

We show the results of the experimental analysis by measuring the worker accuracy and the time spent per page for the proposed method, and we compare such measures with the values from the original platform logs. We modified the logs via sub-sampling, varying the number of gold questions available and the number of judgements performed for each non-gold question. For the rest of the paper, we choose a realistic value of 4.4% of gold questions and compare it with the higher value of 12.4%, more challenging to achieve in practice due to the cost of generating gold questions; with the same spirit, we will show the case of 4 and 10 judgements collected for each non-gold question respectively. It is important to notice that even the cases with low gold ratio and number of judgements may be higher than the usual parameters used in typical crowdsourcing experiments.

In Figure 7, the average worker accuracy is shown. When the gold ratio is high, more time is required for the inferential system to gain precision. However, it is interesting to notice that in all cases the accuracy of the workers have stayed above the threshold of 70%, that was the value under which a worker would have been rejected. Regarding the number of judgements per non-gold question, we do not observe a notable trend on the accuracy of the workers.

On the other hand, we can see in Figure 8 that the time a worker will save by using the proposed system is considerably higher when fewer judgements per non-gold question are used, especially in the transient phase: this is because the number of false positives is higher when the two distributions of multiplicities (of gold and non-gold questions as depicted in Figure 6) have a closer average value.

More importantly, we can see that if either the gold ratio or number of judgements per page are low, the inferential system will allow the workers to complete the tasks in *one fifth* of the original time (after the transient phase): this means that on average, the worker will only need to answer to 2 questions per page, *ignoring 8 questions per page*, without a significant loss in accuracy. This result shows that the proposed system can completely disrupt the gold set paradigm for quality control.

Finally, we repeated the whole experiment on the biggest dataset from the CSTA repository, with ID f354285. The dataset consists of 76,183 judgements from 230 workers Of the unique 13,371 questions, a 4.3% of them are gold questions, and each question have been judged 5 times. We did not perform sub-sampling. The results are shown in Figure 9. Even in this case, the accuracy of the workers never dropped below the threshold of 0.7. After the transient phase, we observed a reduction of accuracy of the 4% and a time saved of about 50%: on average the workers had to answer only to half of the questions in the page to maintain an accuracy that allowed them to complete the entire job.

Number of Colluding Workers. In all experiments in this section, we considered the scenario where all workers are colluding. While an extensive study of the relaxation of this assumption is a major undertaking and left for future work, we can discuss its implications in the simple case in which all workers have equal retention and all enter the job at random times. If not all workers are colluding, some gold questions will not be reported. However, by Assumption 2, this is equivalent to having those gold questions still not reported because of entering early in a batch. The result is that M colluding workers (over a total of N workers) can at best expect to reach, towards the end of the batch, an accuracy equivalent to the one achievable when all workers are colluding and only $\frac{M}{N}$ of the batch has been completed. However, the presence of non-colluding workers is not affecting in any other way the inference mechanism.

5 Countermeasures

In this section we describe possible approaches to deal with the proposed attack scheme.

Gold set size. A potential countermeasure that would make the attack more difficult is to increase the gold size

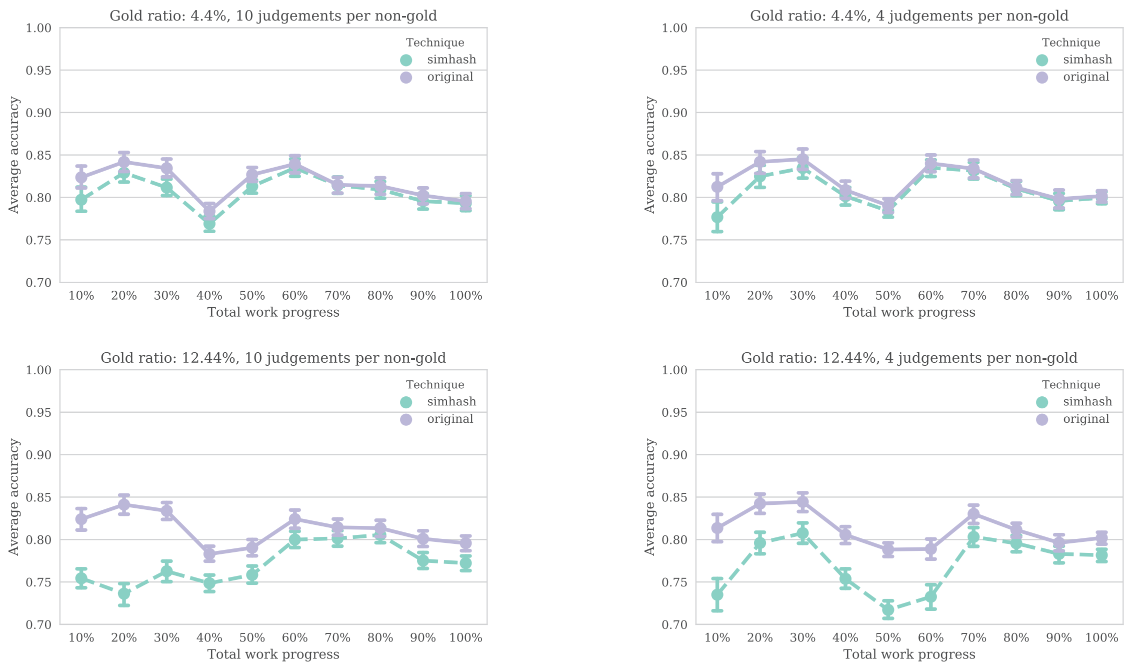


Figure 7: Average worker accuracy for the original logs and for the proposed method. On the left, each non-gold question has 10 judgements, on the right 4. On the top row we used a number of gold questions equal to 4.4%, on the bottom row 12.4%.

set. This will however significantly increase the overall data collection cost. In (Clough et al. 2013), the cost of generating gold questions for the relevance assessment problem was estimated to be above 4 times the minimum wage (for a senior civil servant). This means that in our example, assuming to pay the crowd workers minimum wage, in order to move from a gold set size of 4.4% to 12.4% (as shown in Figure 8), it would be required an additional 54% of the crowdsourcing cost already undertaken.

Number of judgements. An alternative solution is to increase the number of judgements required per non-gold question. However, from our experiments it seems that the effectiveness of this solution is rather limited. Moreover, there is an additional crowdsourcing cost to be taken in account for such an approach.

Worker retention. As shown in Section 3, having crowd workers with high retention will significantly reduce the strength of this attack, because of the reduced initial assessment requirement, and because of the fact that each worker will only see different gold questions. In other words, after a fixed number of tasks are completed, the probability of having gold questions with high multiplicity in the inferential system is low if those tasks have been completed by few prolific workers, because the total number of gold questions shown will be lower (because of fewer initial assessments) and because the probability of having repeated gold questions overall will be lower than the corresponding scenario where many workers completed the same number of tasks. This solution is interesting because increasing retention (e.g., through better task design or reward schemes) can also improve the quality of the work thanks to learning effects on long-standing workers (Difallah et al. 2014).

Non uniform selection from the gold set. Another countermeasure can be to not satisfy Assumption 2: instead of sampling uniformly at random from the pool of questions that have not yet been shown to the worker, there could be a better approach that takes into account the overall sequence of gold questions shown and the possible existence of this attack scheme, potentially mitigating its effectiveness, especially if worker retention is not uniformly distributed.

Programmatic gold questions. Using always different, dynamic generated gold questions, as in (Oleson et al. 2011), that also have sufficiently distant simhashes would be an ideal solution. This could be achieved also by modifying carefully the way the questions are rendered. However, this approach would require a careful design phase again increasing the initial setup cost.

Additional quality controls. Using additional quality controls, like punishing workers that are too fast in answering could make this attack harder. However, this can increase the number of gold preys: workers that are performing as expected but ends up failing the quality control because they are faster than the average (Gadiraju et al. 2015).

6 Societal and Economic Implications

It could be argued that the gold set quality assurance paradigm has similar effects to the classical panopticon effect in the workplace (Vorvoreanu and Botan 2000; Botan

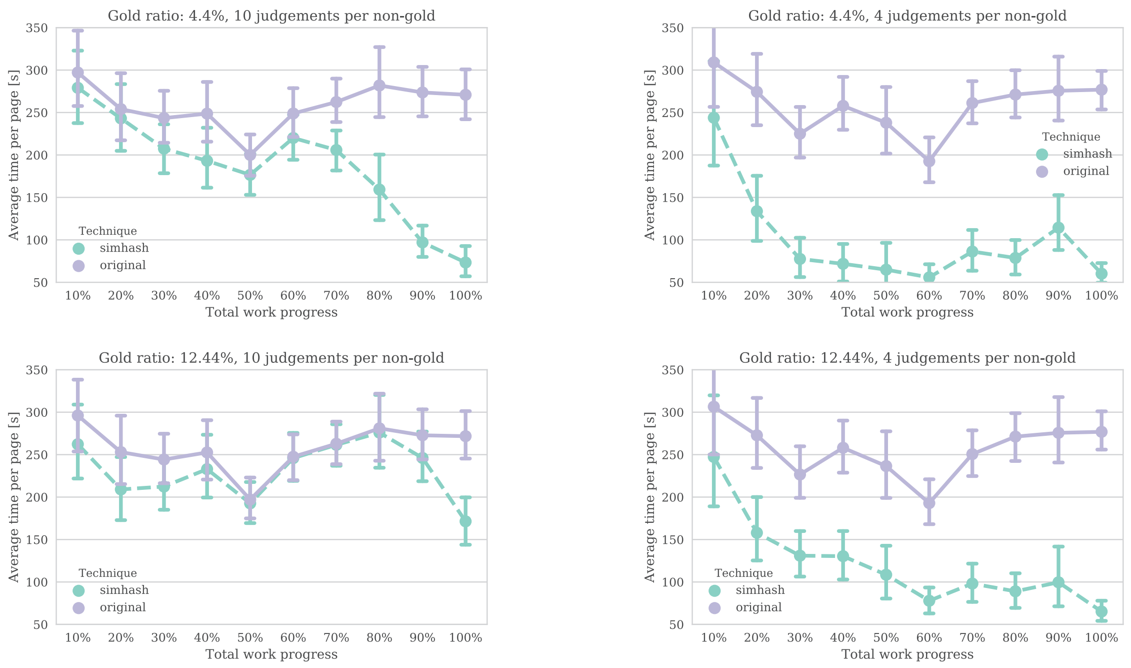


Figure 8: Average time spent per page for original and proposed method on first dataset. On the left each non-gold question has 10 judgements, on the right 4. On the top row we used a number of gold questions equal to 4.4%, on the bottom row 12.4%.

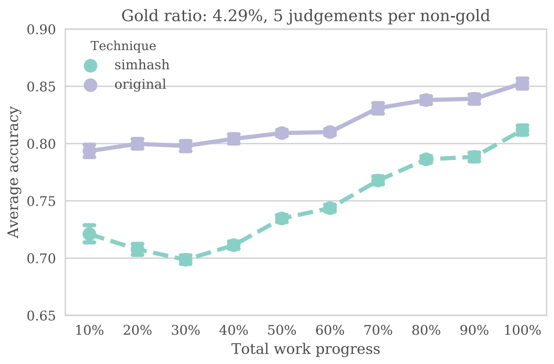


Figure 9: Average worker accuracy for the original logs and for the proposed method on the second dataset.

1996; D'Urso 2006; Stahl 2008). That is, workers' understanding that they might be being observed at any single moment means they feel compelled to self-govern their behaviour at all times in line with the employer's wishes.

As Holland, Cooper, and Hecker (2015), Snyder (2010), and Knox (2010) has shown, electronic surveillance of this kind is correlated with reduction of trust in management, perceived quality of the workplace relationship and can negatively impact work effort, attitudes, and communication in the workplace.

On the other hand, the current crowdsourcing platform architectures, by design, do not allow for different quality assurance mechanisms: workers are anonymous, undifferentiated and indistinguishable, and thus the construction of a trust relationship between crowd worker and requester is very arduous. In (McInnis et al. 2016), reducing rejection risk and building trust towards requesters is identified as a top priority to improve outcomes for all parties in online labour markets.

The architecture of crowdsourcing systems clearly has an important impact on both labour quality and worker satisfaction. The inherently dynamic nature of crowdsourcing platforms make the aforementioned quality assurance mechanisms potentially prone to abuse against workers, but at the same time exposes many novel techniques for worker self-organization and efforts to constitute a more equal power balance between workers and requesters. Reconceptualising the idea of the 'exploit' from hacker culture, Galloway and Thacker argue that in the networked age those that aim to resist dominant power structures must turn their attention to the 'vulnerabilities' embedded within networked infrastructures and leverage these 'exploits' for the purpose of bringing about positive social change. The framework we presented here is a clear example of this kind of exploit, functioning as a form of 'sousveillance' - or, watching from below - by turning the gaze back on the requesters.

Should the system described in this paper take hold in the crowdsourcing community, it would negatively impact the effectiveness of the gold question paradigm for quality assurance, forcing a shift towards different quality assurance approaches. In a way, the proposed system allows an indirect form of horizontal communication, re-balancing in part the digital power imbalance (Cushing 2013; Sandford 2006) between workers and requesters and enabling, rephrasing and adapting to this context (Marx 2003; Kulynych 1997), a less passive and quiescent labour force (Salehi et al. 2015).

With our efforts, we encourage the crowdsourcing research community to push towards a direction, e.g., having more transparency and data portability across platforms (Sarasua and Thimm 2014), or having collective dispute resolution mechanisms (McInnis et al. 2016), where trust between requesters and crowd workers can be achieved (e.g., the direction in which TurkOpticon is going (Irani and Silberman 2013)), removing the need of such granular quality assurance techniques.

7 Conclusions

In this paper, we showed that the popular gold question method for quality assurance is prone to an attack carried out by a group of colluding crowd workers that is easy to implement and deploy. We described an inferential system based on a browser plugin and a server, that can exploit the inherent limited size of the gold set to detect which parts of a crowdsourcing job are more likely to be gold questions. We have also showed how the described attack is robust to various forms of randomisation and programmatic generation of gold questions[4]. Integration with existing plugins like TurkOpticon (Irani and Silberman 2013) is left for future work where we envision implementing a 'traffic light' alert system for signalling potential gold questions to workers similar to the way TurkOpticon signals requester reputation levels.

From real-world crowdsourcing experiments, we saw that when using the proposed method, workers are required to answer to only half (or even a fifth, in some conditions) of the questions presented to them, still maintaining an accuracy level high enough to avoid being excluded from the job.

Regarding potential countermeasures, we observed that increasing the gold set size or the number of judgements per question might be useful but infeasible in terms of cost. On the other hand, we noticed that increasing worker retention through, for example, better task design might be a win-win solution that could also make crowd workers more satisfied and performing better. Other countermeasures that could be explored are the use of programmatic gold creation and probabilistic choice of the order in which the gold questions are posed.

Finally, we discussed the economic and sociological implications of these kind of attacks where we pointed out the positive repercussion on the future of crowd work of the creation of stronger and long-term worker-requester relationships where bilateral trust can be established.

Regarding future research directions, other than exploring the proposed countermeasures in detail, it would be interesting to refine the attack scheme by using locally optimised likelihood thresholds to balance the time saved by the workers and their loss in accuracy. Moreover, it is necessary to study the robustness of the method with respect to the number of workers colluding and coordinated attack times (Lasecki, Teevan, and Kamar 2014; Difallah, Demartini, and Cudré-Mauroux 2012).

More advanced collusion attack schemes may include the sharing of the correct answer for gold questions, that would increase even more the gain in time spent and reduce the risk of being detected.

[4]The core functionalities of the plugin are available at http://github.com/AlessandroChecco.

8 Acknowledgments

This project is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 732328.

References

Aggarwal, C. C. 2005. On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st international conference on Very large data bases*, 901–909. VLDB Endowment.

Benoit, K.; Conway, D.; Lauderdale, B. E.; Laver, M.; and Mikhaylov, S. 2016. Crowd-sourced text analysis: Reproducible and agile production of political data. *American Political Science Review* 110(2):278–295.

Bentivogli, L.; Federico, M.; Moretti, G.; and Paul, M. 2011. Getting expert quality from the crowd for machine translation evaluation. *Proc. MT Summmit* 13:521–528.

Botan, C. 1996. Communication work and electronic surveillance: A model for predicting panoptic effects. *Communications Monographs* 63(4):293–313.

Buchholz, S., and Latorre, J. 2011. Crowdsourcing preference tests, and how to detect cheating. In *Twelfth Annual Conference of the International Speech Communication Association*.

Clough, P.; Sanderson, M.; Tang, J.; Gollins, T.; and Warner, A. 2013. Examining the limits of crowdsourcing for relevance assessment. *IEEE Internet Computing* 17(4):32–38.

Cushing, E. 2013. Amazon mechanical turk: The digital sweatshop. *Utne Reader*.

Damashek, M. 1995. Gauging similarity with n-grams: Language-independent categorization of text. *Science* 267(5199):843.

Daniel, F.; Kucherbaev, P.; Cappiello, C.; Benatallah, B.; and Allahbakhsh, M. 2018. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Comput. Surv.* 51(1):7:1–7:40.

Difallah, D. E.; Catasta, M.; Demartini, G.; and Cudré-Mauroux, P. 2014. Scaling-up the crowd: Micro-task pricing schemes for worker retention and latency improvement. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

Difallah, D. E.; Demartini, G.; and Cudré-Mauroux, P. 2012. Mechanical cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms. In *CrowdSearch*, 26–30. Citeseer.

Dow, S.; Kulkarni, A.; Bunge, B.; Nguyen, T.; Klemmer, S.; and Hartmann, B. 2011. Shepherding the crowd: managing and providing feedback to crowd workers. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, 1669–1674. ACM.

D'Urso, S. C. 2006. Who's watching us at work? toward a structural–perceptual model of electronic monitoring and surveillance in organizations. *Communication Theory* 16(3):281–303.

Fraley, C., and Raftery, A. E. 1998. How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal* 41(8):578–588.

Gadiraju, U.; Kawase, R.; Dietze, S.; and Demartini, G. 2015. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 1631–1640. ACM.

Galloway, A. R., and Thacker, E. 2007. *The exploit: A theory of networks*, volume 21. U of Minnesota Press.

Holland, P. J.; Cooper, B.; and Hecker, R. 2015. Electronic monitoring and surveillance in the workplace: The effects on trust in management, and the moderating role of occupational type. *Personnel Review* 44(1):161–175.

Huang, S.-W., and Fu, W.-T. 2013. Enhancing reliability using peer consistency evaluation in human computation. In *Proceedings of the 2013 conference on Computer supported cooperative work*, 639–648. ACM.

Ipeirotis, P. G.; Provost, F.; and Wang, J. 2010. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, 64–67. ACM.

Irani, L. C., and Silberman, M. 2013. Turkopticon: Interrupting worker invisibility in amazon mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, 611–620. ACM.

Kittur, A.; Chi, E. H.; and Suh, B. 2008. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, 453–456. ACM.

Knox, D. 2010. A good horse runs at the shadow of the whip: Surveillance and organizational trust in online learning environments. *The Canadian Journal of Media Studies* 7:07–01.

Kulynych, J. J. 1997. Performing politics: Foucault, habermas, and postmodern participation. *Polity* 30(2):315–346.

Lasecki, W. S.; Teevan, J.; and Kamar, E. 2014. Information extraction and manipulation threats in crowd-powered systems. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, 248–256. ACM.

Le, J.; Edmonds, A.; Hester, V.; and Biewald, L. 2010. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *SIGIR 2010 workshop on crowdsourcing for search evaluation*, 21–26.

Marx, G. T. 2003. A tack in the shoe: Neutralizing and resisting the new surveillance. *Journal of Social Issues* 59(2):369–390.

McInnis, B.; Cosley, D.; Nam, C.; and Leshed, G. 2016. Taking a hit: Designing around rejection, mistrust, risk, and workers' experiences in amazon mechanical turk. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, 2271–2282. ACM.

Oleson, D.; Sorokin, A.; Laughlin, G. P.; Hester, V.; Le, J.; and Biewald, L. 2011. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. *Human computation* 11(11).

Sadowski, C., and Levin, G. 2007. Simhash: Hash-based similarity detection.

Salehi, N.; Irani, L. C.; Bernstein, M. S.; Alkhatib, A.; Ogbe, E.; Milland, K.; et al. 2015. We are dynamo: Overcoming stalling and friction in collective action for crowd workers. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, 1621–1630. ACM.

Sandford, R. 2006. Digital post–colonialism.

Sarasua, C., and Thimm, M. 2014. Crowd work cv: Recognition for micro work. In *International Conference on Social Informatics*, 429–437. Springer.

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. In *Proceedings of the conference on empirical methods in natural language processing*, 254–263. Association for Computational Linguistics.

Snyder, J. L. 2010. E-mail privacy in the workplace: A boundary regulation perspective. *The Journal of Business Communication (1973)* 47(3):266–294.

Stahl, B. C. 2008. Forensic computing in the workplace: hegemony, ideology, and the perfect panopticon? *Journal of workplace Rights* 13(2):167–183.

Steyerberg, E.; Harrell, F.; and Frank, E. 2003. Statistical models for prognostication. *Symptom Research: Methods and Opportunities. Bethesda, MD: National Institutes of Health.*

Vorvoreanu, M., and Botan, C. H. 2000. Examining electronic surveillance in the workplace: A review of theoretical perspectives and research findings. In *the Conference of the International Communication Association*.