
Article:
Checco, A. orcid.org/0000-0002-0981-3409, Bates, J. and Demartini, G. (2020) Adversarial attacks on crowdsourcing quality control. Journal of Artificial Intelligence Research, 67 (2020). pp. 375-408. ISSN 1076-9757

https://doi.org/10.1613/jair.1.11332

Journal of Artificial Intelligence Research 67 (2020) 375-408 Submitted 02/2019; published 03/2020

Adversarial Attacks on Crowdsourcing Quality Control

Alessandro Checco A.CHECCO@SHEFFIELD.AC.UK
Information School, The University of Sheffield
Regent Court 211 Portobello
Sheffield S1 4DP, United Kingdom

Jo Bates JO.BATES@SHEFFIELD.AC.UK
Information School, The University of Sheffield
Regent Court 211 Portobello
Sheffield S1 4DP, United Kingdom

Gianluca Demartini G.DEMARTINI@UQ.EDU.AU
School of Information Technology and Electrical Engineering
University of Queensland, GP South Building
Staff House Road, St Lucia QLD 4072, Australia

Abstract

Crowdsourcing is a popular methodology to collect manual labels at scale. Such labels are often used to train AI models and, thus, quality control is a key aspect in the process. 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 for which the requester knows the correct answer and, thus, is able to directly assess crowdwork quality. In this paper, we show that such a mechanism is prone to an attack carried out by a group of colluding crowdworkers 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 question detection system, using real-world data from a popular crowdsourcing platform. Our experimental results show that crowdworkers using the proposed system spend more time on signalled gold questions but do not neglect the others thus achieving an increased overall work quality. 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 the 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 incompetent workers, loss of attention, and potential scammers interested in the monetary reward attached to the tasks (Daniel, Kucherbaev, Cappiello, Benatallah, & Allahbakhsh, 2018).

Many solutions that deal with low-quality contributions in crowdsourcing have been proposed so far. The majority are technical solutions based upon monitoring of workers and their outputs at a distance. For example, Snow, O'Connor, Jurafsky, and Ng (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. Gadiraju, Kawase, Dietze, and Demartini (2015) analyse the behavioural patterns of microtask workers to differentiate trustworthy and untrustworthy workers. Only in rare cases do solutions orientate towards more social mechanisms for quality management. For example, Dow, Kulkarni, Bunge, Nguyen, Klemmer, and Hartmann (2011) introduce peer-to-peer feedback systems to encourage worker engagement and high-quality work.

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, Edmonds, Hester, & Biewald, 2010; Huang & Fu, 2013) which are used to validate the accuracy of crowd answers. Similar to other techniques, the use of gold questions is a technical solution based upon monitoring crowdworker activity. Gold question sets should have specific characteristics (Oleson, Sorokin, Laughlin, Hester, Le, & Biewald, 2011):

1. Relatively small size to minimise creation cost as correct answers are typically created by expert editors. Moreover, crowdworkers need to be paid when answering such questions that do not bring new information to the dataset.

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 to prevent priming (Qarout, Checco, & Demartini, 2016).

5. Semantically and structurally similar to the non-gold questions to minimise the possibility of crowdworkers 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, Federico, Moretti, & Paul, 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 the inherent limit on the size of the gold set can be exploited to perform an attack on the crowdsourcing platform: workers can collude to beat the monitoring system by using an inferential system to detect which questions are likely to be gold questions.

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 by 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 & Latorre, 2011).

To simplify the analysis, we will also make the following assumption, which is valid for many crowdsourcing platforms (as we verify in Section 5):

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

We relax this assumption in Section 7, where we test whether showing the gold question using a different distribution can indeed make this attack harder.

This work extends the one of Checco, Bates, and Demartini (2018), where we introduced this kind of attack scheme, with the following improvements and additions: (i) we provide a more detailed literature review and an extended background presentation; (ii) we test an important countermeasure by relaxing Assumption 2; (iii) we expand many parts including extended explanations using more figures and experimental results; (iv) we substantially extend the analysis of the societal implications; (v) we perform a new set of experiments where we analyse the workers' behaviour while using the proposed attack.

The rest of the paper is structured as follows. In Section 2, we discuss previous work in the area of quality control techniques used in paid crowdsourcing platforms as well as existing work about adversarial attacks on these quality control techniques. Then, in Section 3, we introduce our system architecture to automatically identify gold questions in crowdsourcing tasks and we describe, in Section 4, the adversarial attack model used by the system to identify gold questions. In Section 5 we present the results of an experimental evaluation aimed at measuring the effectiveness of the proposed gold question detection technique. We then look at different aspects of our experimental evaluation such as the

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.

behaviour of crowdworkers in a condition where gold questions have been identified (Section 6) and the effectiveness of possible countermeasures to the proposed attack scheme (Section 7). In Section 8 we discuss the implications on the crowdsourcing ecosystem of providing tools to crowdworkers to obtain better visibility on the quality control process. Finally, in Section 9 we draw our conclusions.

2. Related Work

We now summarise the related work in the paid crowdsourcing Computer Science and Social Sciences communities. We will show that the former focused on requester-centric quality control issues, and the latter on crowdworkers labour conditions.

2.1 Quality Control and Gold Questions in Paid Crowdsourcing

Quality control is one of the main challenges in paid micro-task crowdsourcing. Because of the monetary reward available to those who complete tasks, there is a need to detect low quality crowdworkers who complete the task inaccurately with the sole purpose of acquiring the reward attached to the task (Gadiraju et al., 2015). Quality control techniques which are commonly adopted in practice include the use of multiple assignments of the same task to several crowdworkers in order to then aggregate their answers thus removing possible noise and random answers. Several aggregation methods have been proposed in the literature (Ipeirotis et al., 2010; Venanzi, Guiver, Kazai, Kohli, & Shokouhi, 2014) and are used to improve the quality of crowdsourced datasets. The most common technique to control for quality is the use of *gold questions* within crowdsourcing tasks. These are questions for which editorial answers have been created and are used to compare against crowd answers to measure their accuracy. However, the use of gold questions comes with the additional cost of generating ground truth answers for them and thus it is not a scalable approach.

Previous research has looked at means to optimise the creation and use of gold questions. For example, Oleson et al. (2011) looked at how to generate gold questions semi-automatically from a seed of manually created input questions. Such approaches enable the scalability of the gold question quality check approach and reduce its cost. However, it is still prone to adversarial attacks as we show in the following sections of this paper. Work by Buchholz and Latorre (2011) looks at the use of gold questions to identify adversarial workers and to understand their behaviours in order to systematically stop them from participation in the crowdsourcing project. Recently, El Maarry and Balke (2018) proposed personalising the number of gold questions for each individual worker assuming that high quality workers would need fewer gold questions to control their work. Such an approach could lead to savings in the cost of generating gold questions.

In this paper, we present statistical methods that colluding crowdworkers may use to attack the gold question quality control mechanism by sharing questions they observe in the crowdsourcing tasks they complete. This approach re-balances the power in the crowdsourcing ecosystem that currently gives full decision control to requesters about the quality of the work done by crowdworkers. The ability for crowdworkers to attack quality control based on gold questions will require requesters and platforms to re-think quality control and to make it based on trust rather than on an observe-and-act paradigm.

2.2 Adversarial Attacks in Paid Crowdsourcing

In order to quickly complete tasks and to optimise their hourly payment rate, adversarial crowdworkers may use several techniques as individual workers like, for example, filling form input fields randomly or using some logic that can trick simple syntactic quality checks, e. g. Gadiraju et al. (2015). More sophisticated attacks require a group of colluding workers who, for example, agree on which answer to give to questions in a crowdsourcing task in order to trick quality control mechanisms based on majority vote aggregation (Difallah et al., 2012).

Sharing knowledge among crowdworkers is common practice and it is not always performed in an adversarial manner. For example, web forums like TurkerNation are commonly used by many crowdworkers to share tasks they have found worthwhile completing and to discuss crowdwork issues such as hourly rates of pay (Yin, Gray, Suri, & Vaughan, 2016). Other than forums, crowdworkers use reputation mechanisms to identify trustworthy requesters to work for. One such tool is TurkOpticon (Irani & Silberman, 2013) which collects worker-generated reviews of MTurk requesters. Such 'interventionist' tools are an example of a functioning knowledge-sharing system for MTurk crowdworkers.

Social science researchers such as those mentioned in the above paragraph adopt various theoretical stances that guide their work. Research emerging from the Computer Supported Cooperative Work and Human-Computer Interaction communities has tended to develop strongly interpretivist insights into crowdworkers' labour conditions e. g. Martin et al. (2014, 2016), Gray and Suri (2019). In the case of Irani and colleagues, such interpretivist insights have been the basis for their development of online tools such as TurkOpticon and Dynamo in collaboration with crowdworkers and community managers such as Kristy Milland, e. g. Salehi et al. (2015), and in their work towards the development of crowdworker unions. Beyond this body of work, other social scientists have adopted more explicitly critical theoretical lenses drawn from, for example, economic sociology and organisational theory (Lehdonvirta, 2018; Moore, 2017), economic geography (Graham, Hjorth, & Lehdonvirta, 2017), and social theory, e. g. Ettlinger (2016).

Our own research follows the critical stance of this latter body of work to frame worker-requester-platform relations as a complex form of capitalist labour relation that is fundamentally exploitative in nature (Moore, 2017), and in which data-driven computational processes function as systems of control over workers (Deleuze, 1992). However, we also draw inspiration from the 'interventionist' work led by Irani and colleagues – understanding such tools as 'exploits' (Galloway & Thacker, 2007) with the potential to ameliorate and disrupt the exploitative relations baked into crowdwork infrastructures. The approach we propose in this paper follows the same principle of peer knowledge sharing (Yin et al., 2016; Irani & Silberman, 2013), but it has the potential to be used in a more 'adversarial' mode by collaborating crowdworkers than prior interventions. It could be easily integrated in systems like Panda Crazy or Mturk Suite (Kaplan, Saito, Hara, & Bigham, 2018) which have already wide adoption among crowdworkers and could improve their work experience even further.

In relation to these motivations, following the presentation of the statistical methods and experimental work in sections 5 and 6, we draw upon a novel theoretical approach that integrates insights from critical data studies, critical labour/organisational studies and surveillance studies in order to illuminate the power dynamics at play in the monitoring

practices of requesters and consider the societal implications of the proposed intervention. Through adopting this approach we aim to produce a deep critique of the labour relations of quality control in crowdwork, but also open a new space for consideration of how the current infrastructural design can be 'exploited' to force the re-constitution of the labour relation in favour of the workers.

3. Proposed System

We now introduce the system able to perform the described attack scheme on crowdsourcing quality checks using gold questions. We consider the case of a batch of crowdsourcing tasks (i. e. a crowdsourcing platform *job*). We assume that a subset of the workers involved in this job collude to attack the platform. The interaction between workers and a third party server, external to the crowdsourcing platform, is shown in Figure 1. The basic structure and usage

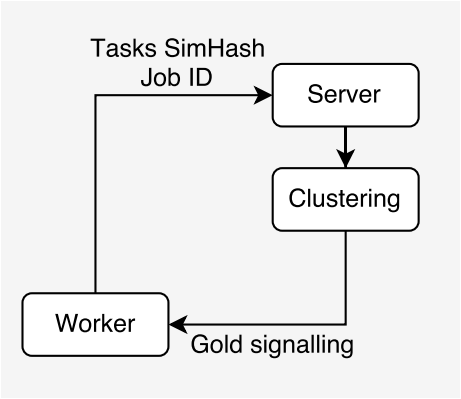


Figure 1: Collaborative gold signalling structure.

are simple: each crowdworker runs a local browser plugin (or a JavaScript bookmarklet) where a set of operations is 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 3.2) to update the information available for that job, and to signal back to the worker the likelihood of each of the current tasks in the job being a gold question. This can be done in mini-batches after a fixed number of new pairs is provided, so no synchronisation is needed.

3.1 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:

Anonymisation 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.

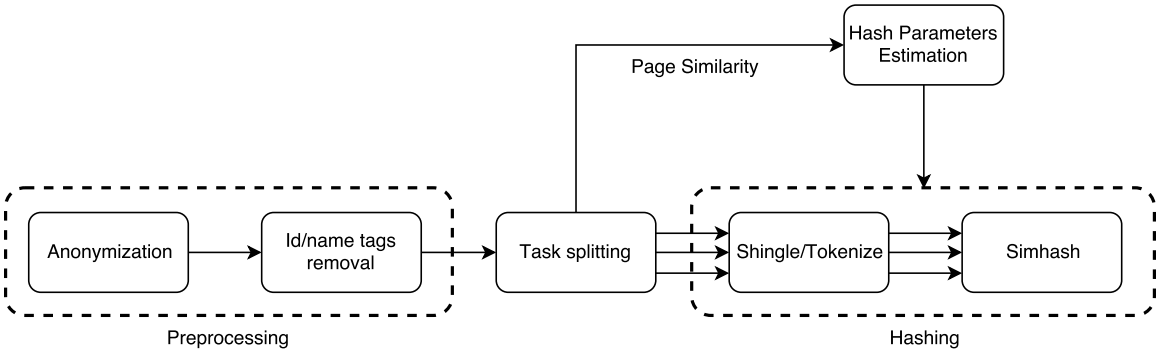


Figure 2: Workflow of the hashing mechanism on the crowdworker side.

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). For example, <**div id**='unique_id' **class**='cls'> would become <**div class**='cls'>. In the rest of this work, a simple heuristic has been devised to perform this task. More advanced methods for this could be developed in future work.

Task splitting The page is split into different fragments, one for each task (see Section 3.3), if needed.

Shingle/tokenization The HTML of each fragment is tokenised with classical 3-gram techniques and then shingled (Damashek, 1995), where the words are either HTML tags or the raw text content. For simplicity, in the rest we chose to use the second approach.

Simhashing For each task (fragment) a simhash (Sadowski & 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 3.5), (ii) it is fast and scalable, (iii) it allows the estimation of 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).

3.2 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[2] 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

2. i. e., differing fragments due to dynamic rendering with CDN/location dependent sources: in our preliminary analysis the same task served to two different workers was never an exact match.

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 5.3, we discuss the effect of the number of colluding workers on the attack performance.

While the architecture is agnostic to the clustering method chosen, we believe the most appropriate one for this context is agglomerative clustering: it works well on non-euclidean distances like the one induced by simhashes, and needs only a distance threshold parameter. Moreover, such a methods can include connectivity constraints induced by the case of multiple tasks per page, as shown in Section 3.3.

3.2.1 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 5, Figure 6. A Gaussian mixture model with two modes will be able to obtain a classification of the current state of the repository of simhashes, together with an estimate of the confidence of the classification accuracy. Such a model is appropriate because (i) the choice of using two components is obtained directly from Assumption 1, (ii) it has a minimal amount of hyperparameters (that guarantees a low transient phase), and (iii) it can capture the potentially unequal variability of the two classes. 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 degrees of freedom to estimate ($p\left[d^2/2 + 3d/2 + 1\right]$), and thus we have to keep the plugin idle for a number of samples of about five times that (Steyerberg, Harrell, & Frank, 2003), 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 & 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 5, 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 it being a gold question. Even after the plugin is active, when a new gold question is shown to the workers for the first few times, the plugin will not signal it as gold, but the workers will still know that the plugin is indeed active and thus gold questions are present in the job: the best behaviour for a worker to maximise quality is to work as usual and use the plugin as confirmation of the presence of gold questions.

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 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.

3.2.2 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 3.3.

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, thus setting their own precision/recall trade-off.

3.3 Multiple Tasks per Page

The operation of task splitting explained in Section 3.1 is not straightforward. It can be achieved in at least two ways:

1. Platform-based heuristic (e. g. Figure Eight 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 Figure Eight (formerly known as Crowdflower) 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 (however, we note that the vast majority of crowdsourcing platforms either uses one task per page, or allows a trivial identification of the splits).

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 method 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.

3.4 Peer-to-peer Implementation

The use of an external server 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[3], to collectively store and retrieve the triples (Job ID; simhash; multiplicity) and locally compute the probability of a task being a gold question.

This approach would significantly increase the attack robustness, as each worker colluding would be a decentralised relay, and would make a countermeasure based on domain banning significantly harder, because no central server would be used.

3.5 Plausible Deniability and Data Security

Crowdworkers 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, potentially sensitive data in the job cannot be reconstructed, even when the third party server is completely compromised, reducing the legal liabilities of the parties involved.

4. 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 a 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 larger 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 a simple model does not allow the estimation of the false positive rate (as will be studied in Section 5 over real data) but it still 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 Figure Eight: 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 Amazon Mechanical Turk by creating qualification tests and by manually distributing gold questions in subsequent tasks.

Moreover, we consider the default Figure Eight aggregation setting: each requested (non-gold) task will be shown to 3 distinct workers. A gold question will not be shown

3. `https://github.com/orbitdb/orbit-db`.

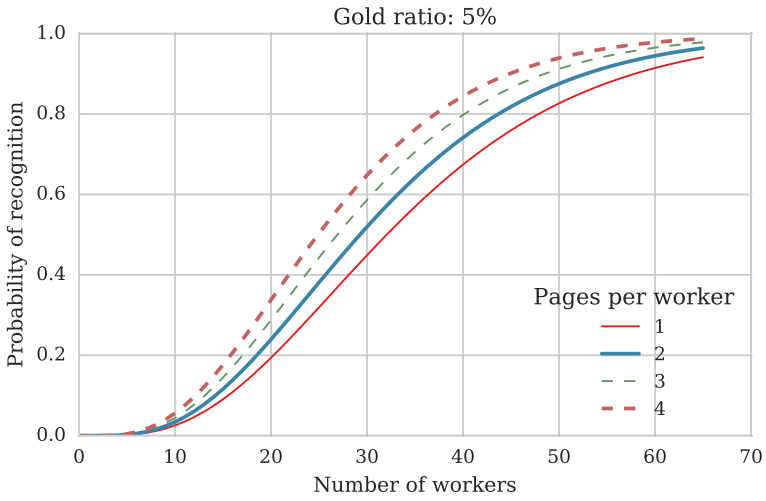


Figure 3: Probability of recognising a gold question varying the number of workers that have already used the system and parametrised by the number of 10-task pages.

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 19 unique gold questions per 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 a certain number of workers had 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 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 judged the same number of pages. In reality, the typical crowd engagement has a power-law distribution: we refer to Section 5 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 crowdsourcing case study.

5. Experimental Analysis of Plugin Effectiveness

In this section, we evaluate the effectiveness of the plugin on real data. Since the effectiveness of the plugin depends both on the power of the inference technique and on the way the workers interact with the signalling system of the plugin (bias, trust, etc.), we focus first on the former, keeping the behaviour of the workers completely controlled (by means of

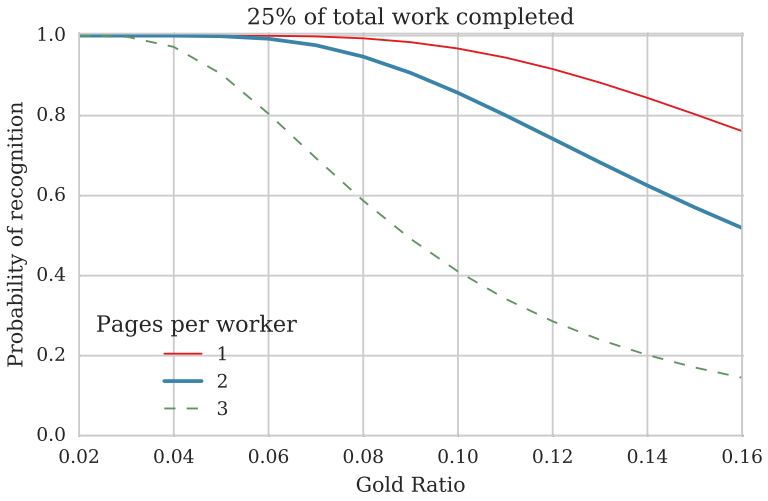


Figure 4: Probability of recognising a gold question after 25 % of the total work, varying the gold questions ratio and parameterised by the number of pages that each worker evaluates.

simulation). Conversely, we refer to Section 6 for an analysis on the workers' interaction with the plugin.

5.1 Plugin Implementation

To perform the rest of the experiment, we did not implement the entire architecture described above. Instead, we simulated a server fast enough to be able to run a clustering every time a new report is provided. On the plugin side, we used a simple heuristic for multi-page splitting, and did not use peer-to-peer functionalities nor (apart from Section 6) reporting/tuning of confidence values. The core functionalities of the plugin to replicate the following experiments are available at `https://github.com/AlessandroChecco/all-that-glitters-is-gold`.

5.2 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, Conway, Lauderdale, Laver, & Mikhaylov, 2016)[4], consisting of crowdsourced annotations of political data. 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, including 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 unusually abundant number of gold questions, and because each non-gold question had been answered by 10 workers with an average of 8.7

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

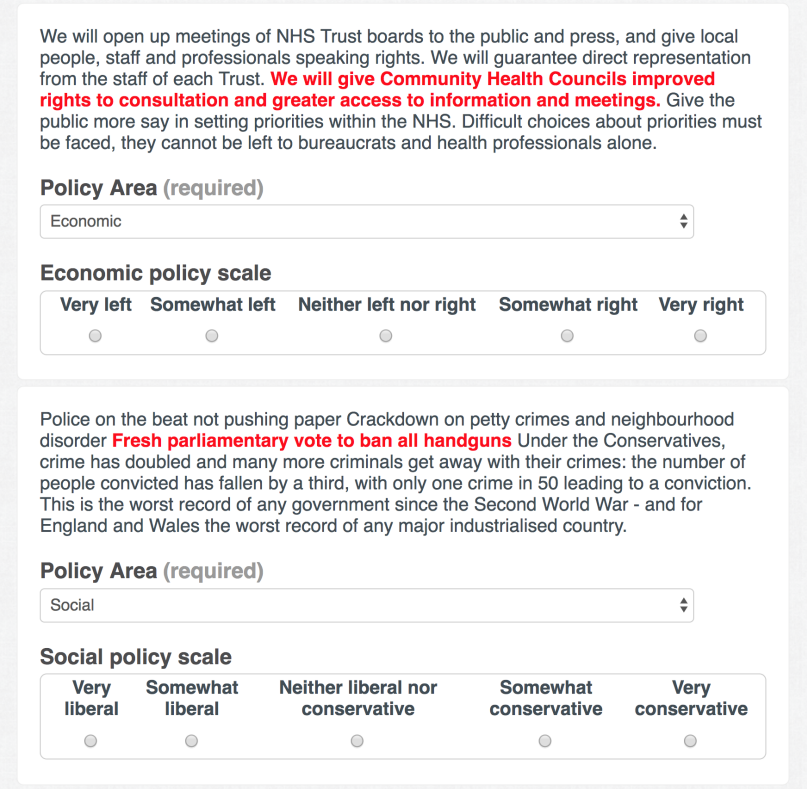


Figure 5: Design of the CSTA task. Workers have to select the most appropriate policy area related to a specific sentence presented to them in its context and the political scale (left/right) where it positions itself.

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

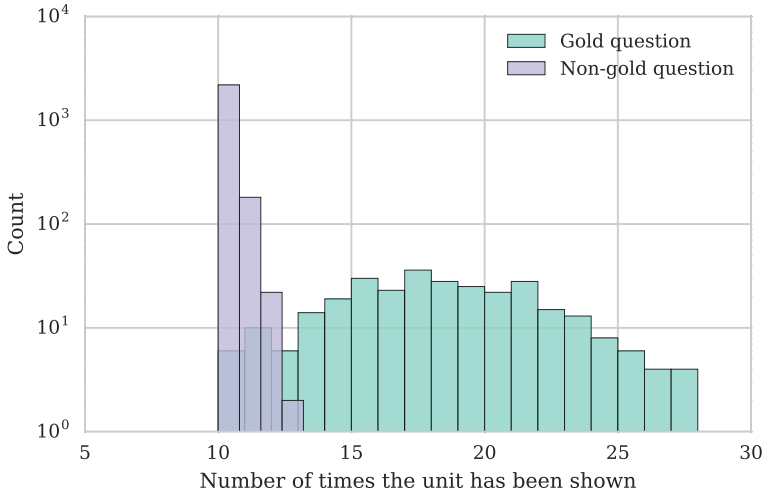


Figure 6: Distribution (log scale) of the multiplicity for gold and non-gold questions in the CSTA task when using 12.4 % of gold questions. Assumptions 1 and 2 are satisfied.

be in the 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 crowdworker original behaviour. The original designer of this job opted for having an initial quiz of 10 questions, 8 of which were gold, and after that to present pages of 10 questions, one of which was 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's ability to answer a gold question is only dependent on their internal state, and not on which gold question they are being shown[5].

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 potentially 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 average time they spent per unit on that page from the log.

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

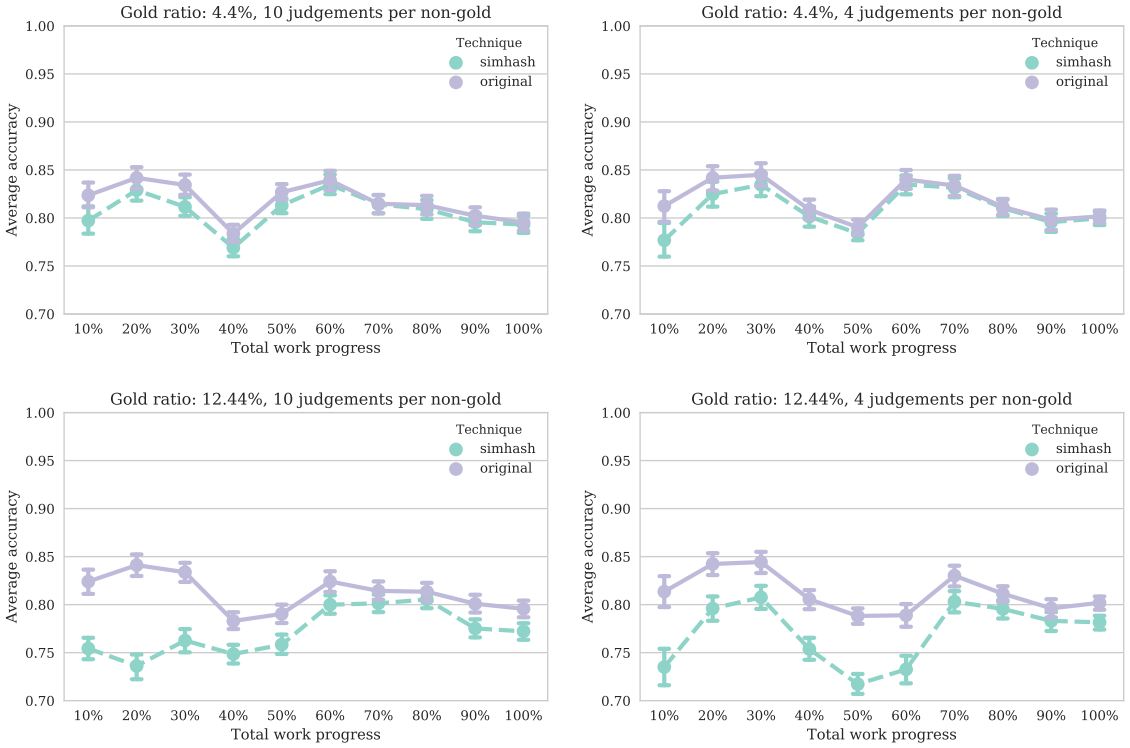


Figure 7: Average worker accuracy for the original logs and for the proposed method. On the left, each non-gold question has 10 judgements, on the right 4. On the top row we used a number of gold questions equal to 4.4 %, on the bottom row 12.4 %.

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 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.

5.3 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 section, 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; in like vein, we will show the case of 4 and 10 judgements collected for each non-gold question respectively. It is important to note that even the cases with low gold ratio and number of judgements may be higher than the usual parameters used in typical crowdsourcing experiments.

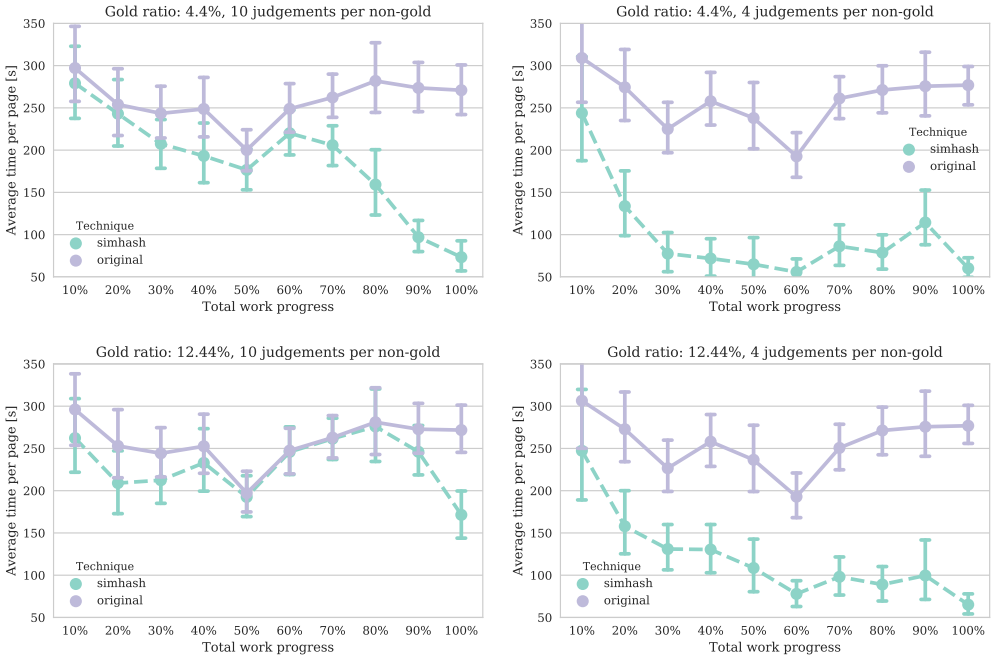


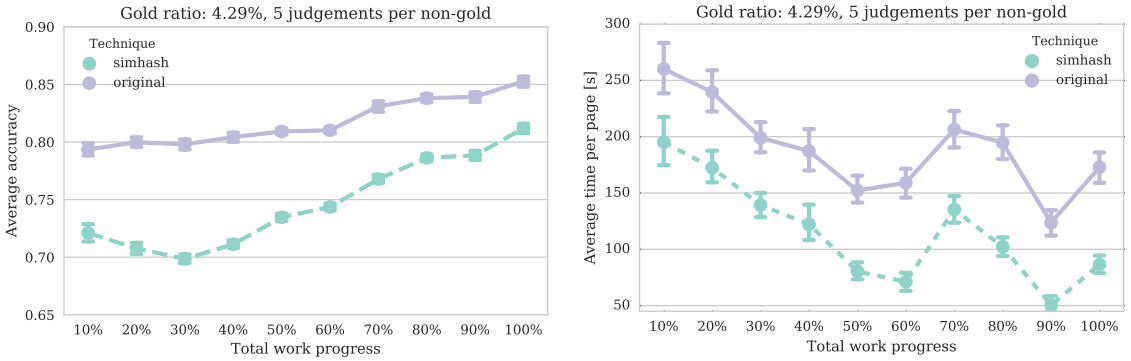
Figure 8: Average time spent per page for original and proposed method on first dataset. On the left each non-gold question has 10 judgements, on the right 4. On the top row we used a number of gold questions equal to 4.4 %, on the bottom row 12.4 %.

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 note that in all cases the accuracy of the workers has 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 the number of judgements per page are low, then 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.



(a) (b)

Figure 9: Average worker accuracy (a) and average time spent per page (b) for the original logs and for the proposed method on the second dataset.

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 13,371 unique questions, 4.3 % of them are gold questions, and each question has been judged five times. We did not perform sub-sampling. Even in this case, the accuracy of the workers never dropped below the threshold of 70 %. After the transient phase, we observed a reduction in accuracy of 4 % (as shown in Figure 9a) and a time saved of about 50 % (as shown in Figure 9b): on average the workers had to answer only half of the questions in the page to maintain an accuracy that allowed them to complete the entire job.

5.3.1 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. For example, if the number of colluding workers is only 10 % of the whole pool of workers, from Figure 7 it is possible to

see that the average accuracy achievable would be only the one of the first data point on the x-axis. Similarly, from Figure 8 it is clear that in this case the time saved by the colluding workers would be negligible. In other words, the number of colluding workers affects in a linear way the speed and extent of the transient phase of the attack. However, the presence of non-colluding workers is not affecting the inference mechanism in any other way.

6. Experimental Analysis of Workers' Behaviour

The analysis in Section 5 was based on some strong assumptions about the behaviour of the workers: a simulated behaviour where worker acted to minimise the time spent on the platform by using the plugin. However, the following research questions remain:

1. How would the workers use the plugin? Would they use it to save time (by ignoring non-signalled questions), to improve quality (by spending more time on the signalled questions), or a combination of both?

2. What is the effect of the plugin's detection inaccuracy on the worker behaviour? Would a loss in trust change the way workers use it?

3. Would workers actually use the plugin?

In this section, we focus on answering the first two questions, by conducting an experiment in which workers interact freely with the plugin while the detection effectiveness of the plugin is artificially controlled. We leave the third question for future work.

6.1 Experiment Design

Workers were required to perform a classification task, where a commercial product (i. e. shoe or garment) and its corresponding customer review was shown. The crowdworker had to decide, for each question, whether the review contained a reference to a size issue, a fit issue, or none of the above. Before the task started, instructions were provided, with a rather convoluted definition of the classification classes and an explanation that a plugin able to signal gold question would have been used in the task.

To minimise the effect of the bias caused by the fact that the workers were aware that this was an experiment (the plugin after all was provided by the designer of the task), we used the crowdsourcing platform's native quality control system to allow the worker to verify that indeed the usual gold set quality control system was in place. Each page was composed by five questions, one of which is a gold question. When a page is submitted, if a gold question is missed the worker will receive a feedback, and the current accuracy level of the worker will be updated. We selected this classification task because it has been annotated in its entirety by domain experts and, from our own experience, it is a rather difficult task. From a pilot run of this experiment we measured the workers' accuracy for each question. We then selected as gold questions a relatively difficult set: in the pilot run, the gold questions had an average accuracy of 80.4 % and a median accuracy of 89 %, while the non-gold question had an average accuracy of 89.7 % and a median of 100 %. The rationale for this choice is that when using the plugin, the only way for the workers to verify the effectiveness of the plugin is to miss a question, because the quality control system of the

platform will in that case notify the loss in accuracy and show the correct answer for that question, revealing that the plugin indeed signalled the correct question from the page, as shown in Figure 12. As an example, if the gold questions were extremely easy, the workers would never miss a gold question, and would never be able to decide whether to trust the plugin. This setup is only necessary in this experiment because the participating workers had never used the plugin before and our goal is to assess their behaviour after trust (or lack thereof) is established.

The plugin signalling was a simple coloured box (red for high confidence, orange for low confidence), that alerted the worker on the possibility that that question might be a gold question, as shown in Figure 11.

We performed three experiments, each with 100 questions and 5 judgements per question, for a total of 1500 tasks involving 244 workers from 46 countries:

Inactive (control): the original task without using the plugin at all. The instruction did not contain information on the plugin either.

Perfect Signalling: all gold questions were signalled with high confidence.

Imperfect Signalling: one third of the gold questions were signalled with high confidence (with a red signal and showing a confidence of 99 %), one third with low confidence (with an orange signal and showing a confidence of 10 %), and one third were missed altogether, with uniform at random selection among these cases[6].

Since the goal of this experiment was to understand the workers behaviour while using the plugin, the following measures were taken to select an appropriate population and to prevent memory biases: At the beginning of the task, a quiz of 5 gold questions was shown, to filter out underperforming workers and bots: workers with less than 40 % accuracy were not allowed to continue. Workers were not allowed to participate in more than one experiment, and additional controls were put in place to prevent workers from restarting a task after gaining knowledge on those questions. Figure 10 shows the instructions provided at the beginning of the task.

Each worker was exposed to the same set of questions and gold questions (selected via uniform sampling), although not all of the workers answered all of the questions. We measured the time spent on each question and the workers accuracy.

6.2 Worker Behaviour with Perfect Signalling

Here we compare the control experiment (i. e. no gold question signalled to workers) with the case in which every gold question has been signalled by the plugin. From Figure 13 we can see that after a phase in which the workers are evaluating the plugin (first 3 pages, which corresponded with testing the plugin 3 times), then the trend of spending more time on the signalled question becomes apparent, compared to the control group.

6. Since the variability of responses in this case was significantly higher, we performed an additional run of this experiment to increase the number of data points.

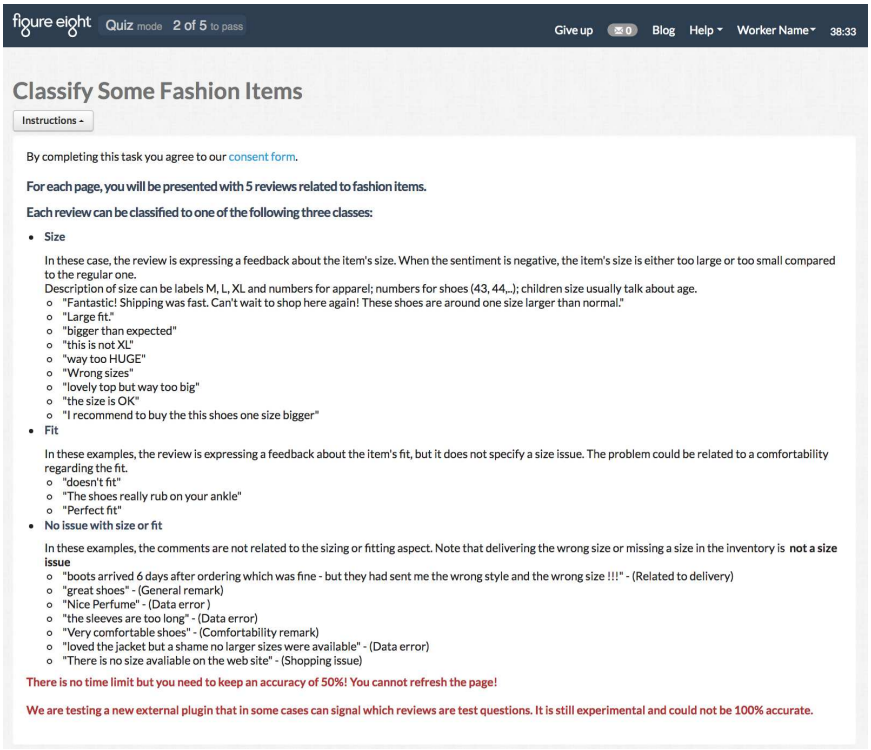


Figure 10: Instructions for the perfect and imperfect signalling experiment.

6.2.1 Intra-page interaction

As shown in Figure 14, there is a significant increase in the time spent on gold questions (t-test, $p < 0.05$), while there is no significant decrease in the worker accuracy for both gold questions and non-gold questions. In fact, perhaps surprisingly, the accuracy on non-gold questions slightly increased, although the difference was not statistically significant.

While a performance improvement on gold question was expected, the performance of workers on non-gold questions is of particular importance for the requester, and thus requires a more careful analysis. As said before, the behaviour of the first 3 pages could be considered as a learning phase for the worker. Removing that part of the dataset, the performance improvement from the control group on the non-gold questions was even more dramatic (as shown in Figure 15): from 63.64 % to 92.31 %, and the improvement was statistically significant (one-tailed t-test, $p < 0.05$). For this reason, a longitudinal study on the same workers is recommended for future work.

6.2.2 Overall Time and Accuracy

A one-way ANOVA on the per-page accuracy (for all pages), with the three plugins as factors, showed that there is a statistically significant difference ($p < 0.05$) on the accuracy: as shown in Figure 16a, the average accuracy using the plugin with perfect signalling increased the accuracy from 79.4 % (control) to 87.3 %, and the median page accuracy from 80 % to

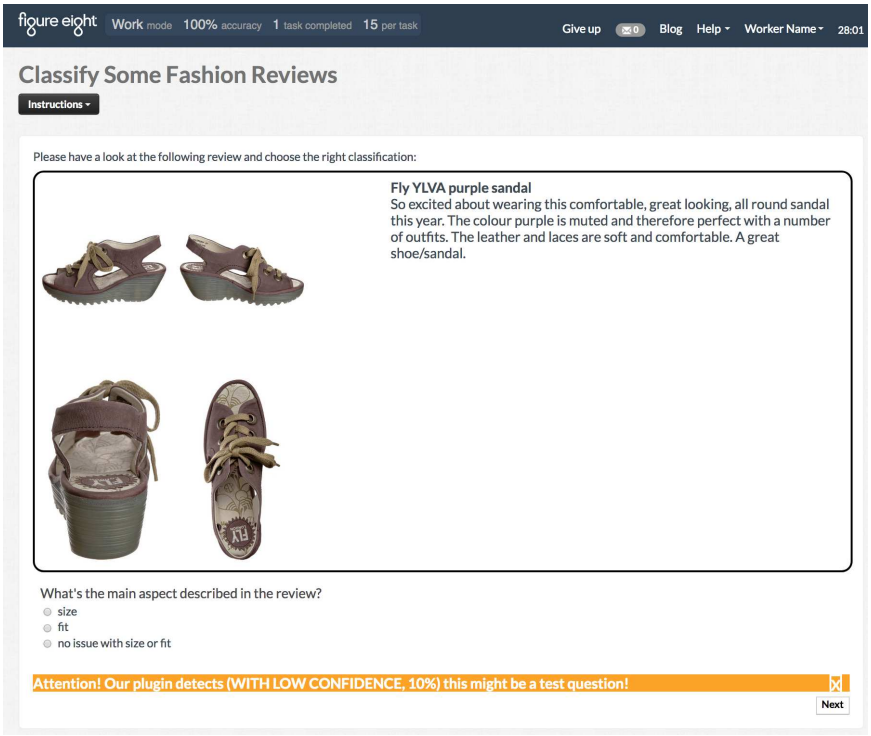


Figure 11: Example of one of the five questions shown on a page: the plugin signalled this question with low confidence.

100 %. On the other hand, the time spent per page showed no statistically significant difference (Figure 16b) between the two experiments.

6.3 Worker Behaviour with Imperfect Signalling

We now focus on understanding the effect of the plugin effectiveness on worker trust. In this experimental setup, one third of the gold questions were signalled with high confidence (with a red signal and showing a confidence of 99 %), one third with low confidence (with an orange signal and showing a confidence of 10 %, as shown in Figure 11), and one third were missed altogether, with uniform at random selection among these cases.

6.3.1 Intra-page Interaction

In Figure 17, while we observe a reduction in time spent on non-gold questions in pages where gold questions were signalled, we can see that the effect is rather reduced compared to the perfect signalling case: there is no statistically significant difference between the times spent on the two different kinds of signalled questions.

6.3.2 Overall Time and Accuracy

Similarly, the overall per-page accuracy and time spent are statistically indistinguishable from the inactive (control) experiment, as shown in Figure 16.

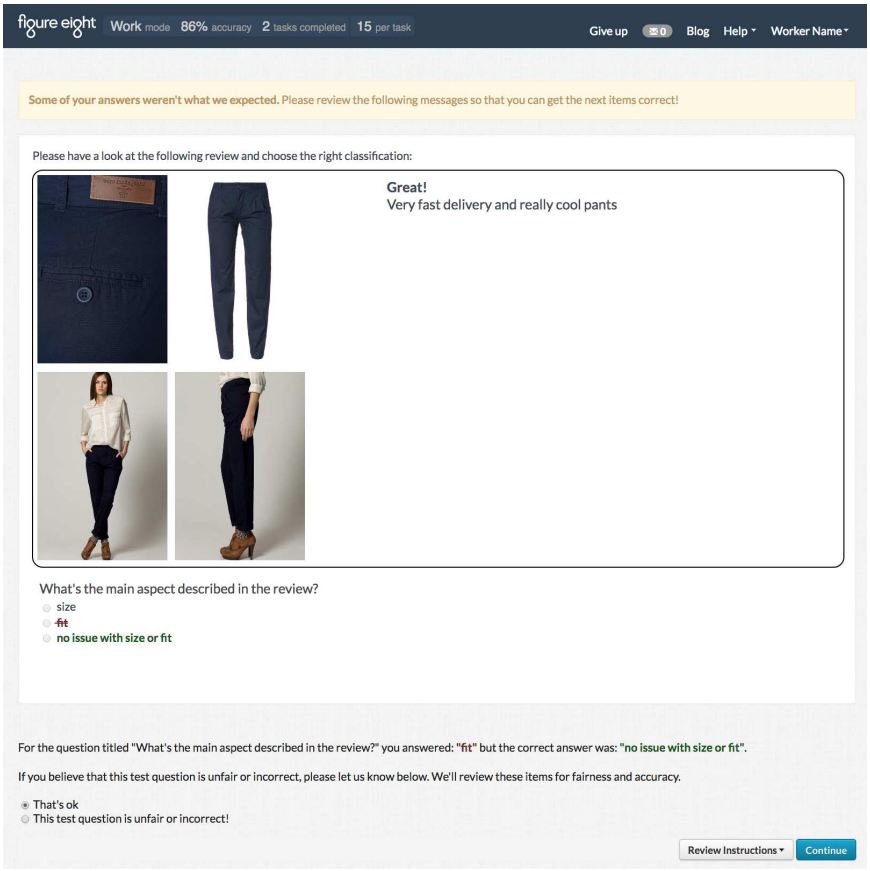


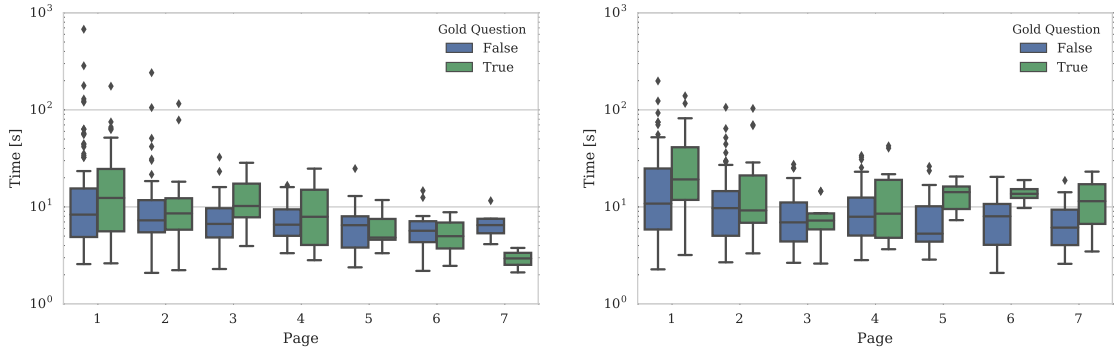
Figure 12: Example of the feedback received after failing a gold question in a page of five questions: the worker is now aware of which question was the gold one. The workers can keep track of their per-page accuracy on the top banner.

6.4 Discussion

In the case of perfect signalling, workers using the plugin spent more time on the signalled questions and less time on the non-signalled ones, while they spent the same time overall in the task. As would be expected, the accuracy on gold questions increased. Perhaps surprisingly, the accuracy on the non-gold (and non signalled) questions also increased (especially after the first pages when workers were still testing the plugin effectiveness). Workers increased their attention on signalled question, while retaining a high (if not often better) ability to work on the rest of the task, perhaps because of the decreased stress/cognitive load caused by the otherwise hidden quality control mechanism.

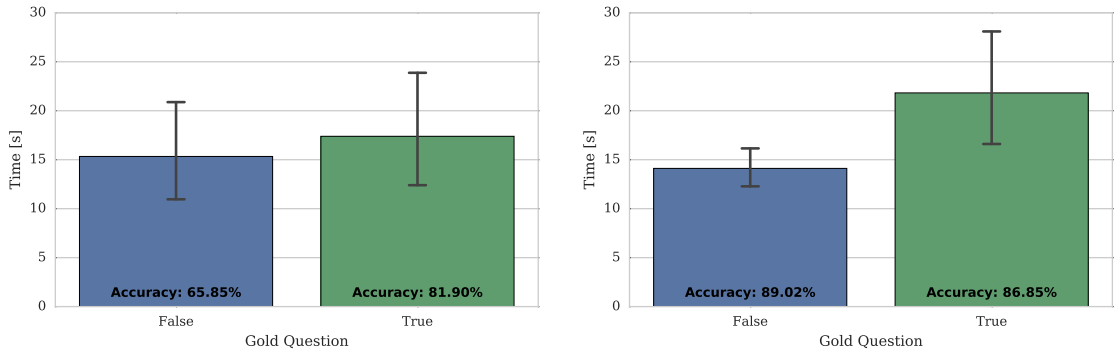
Regarding the imperfect signalling experiments, we can observe that the results are statistically indistinguishable from the control group for both time spent and accuracy: the lack of trust in the plugin made the workers behave in the same way as the control group.

Our approach is also subject to the issue in which the *confidence* of the gold detection classifier is inaccurate. In such a case, the classifier could incorrectly miss some gold questions and label them as being non-gold with high confidence (this problem is often called



(a) Inactive (b) Perfect Signalling

Figure 13: Time spent by workers on each question grouped by page on inactive (a), and perfect signalling (b) experiments. More time is spent on signalled gold questions after the first 3 pages.



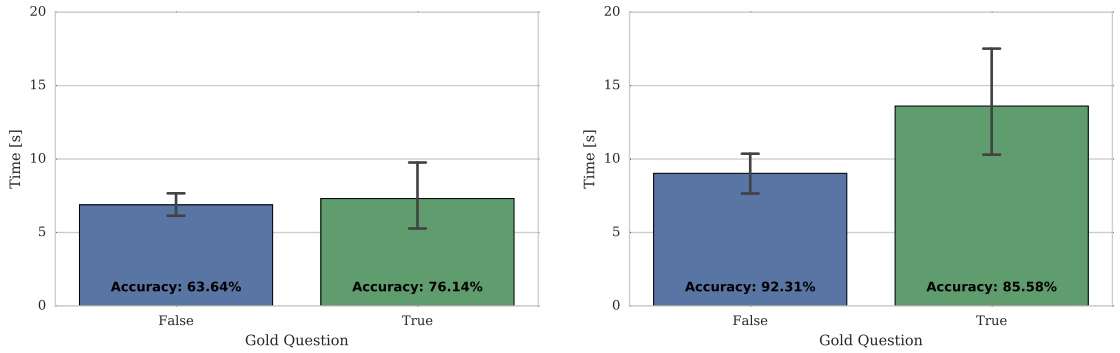
(a) Inactive (b) Perfect Signalling

Figure 14: Accuracy and time spent by workers on gold questions for inactive (a), and perfect signalling (b) experiments. More time is spent on signalled gold questions but accuracy does not decrease for the other questions.

unknown unknowns). In such cases, workers would be likely to miss the unidentified gold questions and potentially damage their platform reputation and get their job rejected. Such issues usually happen when some classes are under-represented in the training data. Thus, in our setting, this could happen when some gold questions are very different from others, and appear very rarely. In summary, the more data is available to the system through the plugin, the less the issue of unknown unknowns should arise and, in the cases when it arises, it would only affect a small minority of gold questions and workers.

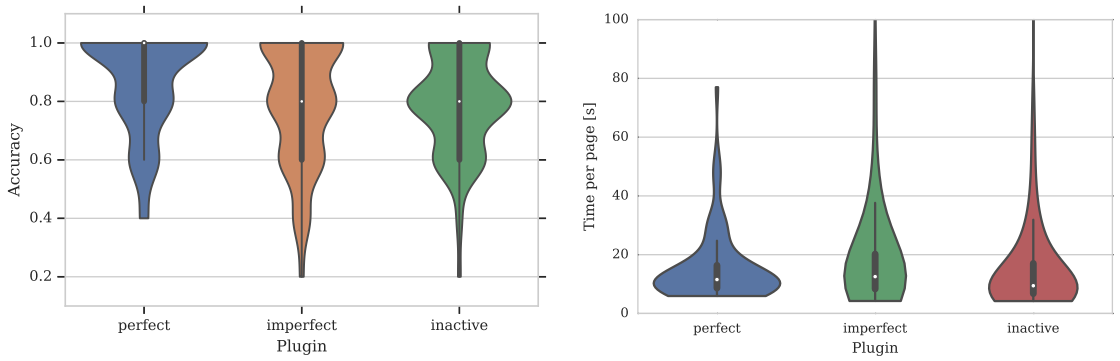
7. Countermeasures

In this section, we describe possible approaches for a requester to mitigate the effects of the proposed attack scheme.



(a) Inactive (b) Perfect Signalling

Figure 15: Accuracy and time spent by workers on gold questions for inactive (a), and perfect signalling (b) experiments, after the first 3 pages. More time is spent on signalled gold questions and accuracy significantly increases.



(a) (b)

Figure 16: Accuracy per page (a), and time per page (b) violin plot distribution for the three plugins. The white dot represent the median.

7.1 Gold Set Size

A potential countermeasure that would make the attack more difficult is to increase the gold set size. This will however significantly increase the overall data collection cost. In (Clough, Sanderson, Tang, Gollins, & Warner, 2013), the cost of generating gold questions for the relevance assessment problem was estimated to be more than 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), an additional 54 % of the crowdsourcing cost already undertaken would be required.

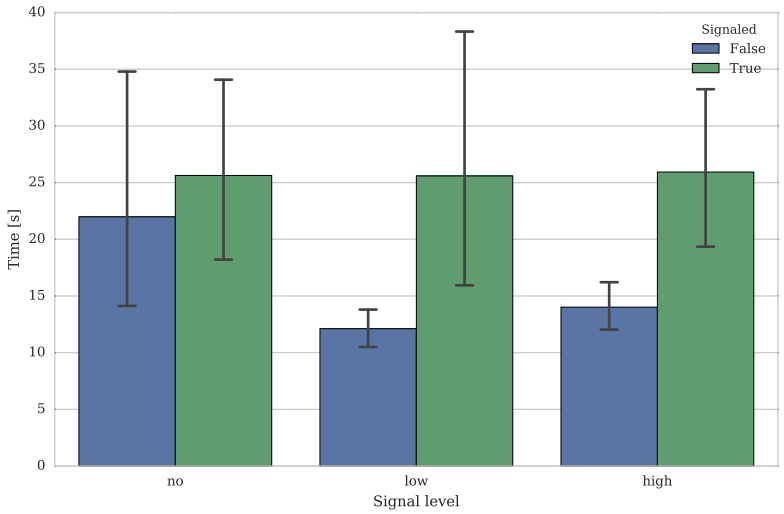


Figure 17: Time spent by workers on each question with varying detection confidence levels for imperfect plugin experiment.

7.2 Number of Judgements

An alternative solution is to increase the number of judgements required per non-gold question. However, from our experiments in Section 4 it seems that the effectiveness of this solution is rather limited. Moreover, there is additional crowdsourcing cost that needs to be taken in account for such an approach.

7.3 Worker Retention

As shown in Section 4, having crowdworkers 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 a few prolific workers. In that case, the total number of gold questions shown will be lower (because of fewer initial assessments) and 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, Catasta, Demartini, & Cudré-Mauroux, 2014).

7.4 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. We tested this hypothesis on the first dataset of Section 5, repeating the simulation after relaxing Assumption 2 in the following way: we serve, at each step, the least seen question from the whole pool of gold questions, while avoiding to show the same gold question twice to the same worker[7]. The results are not encouraging for this countermeasure: the difference in accuracy between this technique and the original technique is less than 2.5 %, while the differences in time spent are of the order of seconds: in both cases the differences are not statistically significant. We believe the reason this countermeasure is not effective is that for relatively small gold set sizes, a uniform serving is statistically indistinguishable from a lexicographical serving. Another approach that should be considered is a countermeasure that exploits potential vulnerabilities in the Gaussian Mixture Model inference method, for example by serving some gold questions a high number of times to make the two modes inference ineffective. We leave for future work an extended analysis of this and other countermeasures.

7.5 Programmatic Gold Questions

Using always different, programmatically 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.

7.6 Inter-worker Agreement

Solutions like the one proposed in (Shah, Balakrishnan, & Wainwright, 2016) would be able to detect the difference in distribution for gold and non-gold questions, potentially identifying workers that answer randomly only on non-gold questions. Attackers could prevent this by colluding on a common rule (e. g. first option) rather than answering uniformly at random, but the study of this different kind of collusion attack is left for future work.

7.7 Time Analysis

Requesters could detect workers that are too fast in answering. This countermeasure could even be refined by analysing the distribution of times for gold and non-gold questions, attempting to identify workers that spend significantly less time on non-gold questions. However, this can increase the number of gold preys: workers that are performing as expected, but who end up failing the quality control because they are faster than the average or that are more careful when they recognise a gold question (Gadiraju et al., 2015). Moreover, when performing the attack, workers could just spend the same time on both types of question by working on multiple tasks in parallel.

7. Assumption 2 required instead to serve the gold questions sampling from the uniform distribution from the pool of gold question not yet seen by the worker.

7.8 Page Obfuscation

A powerful countermeasure would be to obfuscate completely the HTML source, by always serving a seemingly identical source: all images served should appear having same path and all pure HTML text should be substituted by images. However, this would require a major architectural restructuring on the platform side, or a costly effort on the requester side.

8. Societal and Economic Implications

The use of monitoring technologies to control the speed and quality of workers' output is not a new phenomenon. However, within the "precarious, unstable, temporary" working conditions of emergent labour markets typified by crowdwork, such techniques are being used as a means to exert control over – and place accountability and responsibility upon – individual workers struggling to earn an income in increasingly competitive labour markets (Moore, 2017, p. 14).

In a data-rich era, a multitude of possibilities open up for gathering and analysing information about the amount and quality of work completed by employees and contractors. As Moore observes: "data is treated as a neutral arbiter and judge, and is being prioritised over qualitative judgements" (2017, p. 3). In such conditions, workers are becoming accustomed to constant observation and measurement enabled by the use of a variety of monitoring technologies. Stories abound in the media and academic press about the introduction of new monitoring technologies into workplaces, i. e. (Saner, 2018; Yeginsu, 2018; Moore, 2017). As a mechanism to control for quality of paid work, the gold set quality assurance paradigm can thus be defined as a type of computer-enabled monitoring of workers.

It can be argued that such monitoring techniques function as systems of control within the capital-labour relation, and that the gold set quality assurance paradigm promotes similar effects to the classical panopticon effect in the workplace (Vorvoreanu & Botan, 2000; Botan, 1996; D'Urso, 2006; Stahl, 2008). That is, workers' understanding that they might be being observed at any single moment means they ought to feel compelled to self-govern their behaviour at all times in line with the employer's wishes (Foucault, 1991); the ideal workers come to "internalise the imperative to perform ... becoming observing, entrepreneurial subjects ... whilst remaining objectified working bodies" (Moore, 2017, p. 15).

Here, we can turn to the work of philosopher Gilles Deleuze (1992) to theorise the nature of this relation of control. Deleuze explored how we ought to re-imagine the nature of social control in societies whose organisation was becoming more open and dynamic than the institutional settings (e. g. factory, school, prison) that were the focus of Foucault's (1991) work on the panopticon. For Deleuze, workers within capitalism had always been subordinated by machines, however he observed that the shift to the 'control society' was marked, in part, by the introduction of computers as the machines of control (Deleuze, 1992, p. 6).

As Haggerty and Ericson (2000) observe, computer-enabled monitoring works to abstract people from their lived realities. Monitoring data are separated from the people that are observed, and become a series of data flows that are re-assembled in different settings to create "data doubles". These data doubles are then used to inform decisions without the need for any meaningful human engagement. Within the control society, the data double comes to mediate relations between human actors – in our case the relation between requester and worker.

Monitoring technologies allow for the management of labour at a distance. They seem the obvious choice for quality control within the context of crowdwork given the highly distributed, anonymous, undifferentiated and indistinguishable nature of the workforce. In such conditions, the construction of a trust relationship between crowdworker and requester would be very arduous, and monitoring presents itself as a far more efficient solution. In fact, the current crowdsourcing platform architectures, by design, do not allow for different quality assurance mechanisms.

Yet, in the gold-set quality assurance paradigm, each response to a gold question takes on exaggerated significance in the worker's "data double". Each time a worker responds to a gold question there is the potential for significant economic consequences for the worker in terms of payments and the possibility of future work. For a crowdworker, ensuring a flawless "data double" becomes a matter of survival within the highly competitive and individualised crowdwork labour market.

As Holland, Cooper, and Hecker (2015), Snyder (2010), and Knox (2010) have demonstrated, electronic surveillance of this kind is correlated with a reduction of both trust in management and the perceived quality of the workplace relationship. It can also negatively impact work effort, attitudes, and communication in the workplace. In effect, such forms of monitoring tend to alienate workers, rather than establishing meaningful increases in worker quality and satisfaction. Workers subject to such labour conditions will respond in various ways. While some will passively accept the conditions of their labour, others will seek out ways of empowering themselves in relation to the monitoring system.

The architecture of crowdsourcing systems clearly has an important impact on both labour quality and worker satisfaction. While the inherently dynamic nature of crowdwork platforms makes quality control mechanisms potentially prone to abuse against workers, at the same time it exposes many novel techniques for worker self-organisation 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 (2007) argue that in the networked age those that aim to resist these systems of control 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 in this paper is a clear example of this kind of exploit, functioning as a form of "sousveillance" (Mann, Nolan, & Wellman, 2002)– or, watching from below – by turning the gaze back on the requesters who design the gold questions. The experimental findings indicate that not only is such an attack easy to implement and employ, but also that such an attack would be difficult to counter (as shown in Section 5), and moreover that when workers employ such technologies the overall quality and efficiency of their work increases (Section 6). While the reasons for this observed improvement in worker quality are uncertain and the results need further corroboration, the findings suggest that indicators of enhanced trust and sense of worker empowerment in the worker-requester relation, as well as less worry about gold-question monitoring, may prove more effective at enhancing quality than one-way monitoring based systems.

Should the system described in this paper take hold in the crowdworker community, it would negatively impact the effectiveness of the gold question paradigm for quality assurance, forcing a shift towards different quality assurance approaches. The system has the potential to enable a less passive and quiescent labour force (Marx, 2003; Kulynych, 1997;

Salehi et al., 2015), potentially ameliorating some of the digital power imbalance (Cushing, 2013; Sandford, 2006) between workers and requesters. On the other hand, it also has the potential to weaken the competitive advantage of crowdworkers who have invested time and energy in enhancing their reputations within the current frameworks for quality control. The implications for both workers and requesters therefore remain somewhat uncertain.

Reducing rejection risk and building trust is identified as a top priority to improve outcomes for all parties in online labour markets (McInnis, Cosley, Nam, & Leshed, 2016). With our efforts, we encourage the crowdsourcing research community to question the efficacy of technologically enabled monitoring systems as the sole means of quality control. Instead, we push towards more socially-orientated frameworks for enhancing labour quality and satisfaction, such as those seen in the work of TurkOpticon (Irani & Silberman, 2013) and McInnis et al. (2016) on collective dispute resolution mechanisms, and initiatives that aim for more transparency and data portability across platforms (Sarasua & Thimm, 2014).

9. Conclusions

In this paper, we showed that the popular gold question method for quality assurance in paid crowdsourcing is prone to an attack carried out by a group of colluding crowdworkers 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[8]. Integration with existing plugins like TurkOpticon (Irani & 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.

In our experimental evaluation we have observed the effect of the use of the proposed attack scheme on workers behaviours in terms of time spent and effectiveness in answering questions. From real-world crowdsourcing experiments, we saw that when using the proposed method, workers are required to answer 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. We also observed that workers are indeed spending more time on signalled gold questions but they are not neglecting others. The most positive observation we made is that in the presence of gold question signalling, the overall quality of work increases, possibly because of the reduced cognitive load caused by being invisibly monitored. We also observed that workers behaviour is highly sensitive to the plugin effectiveness: this result could also be caused by the fact that the workers used this plugin for the first time during the experiment, so trust establishment played a major role here. This effect should be studied more in depth in future work. 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. The countermeasure of serving gold questions avoiding repetitions among the whole pool also proved ineffective. On the

8. The core functionalities of the plugin are available at `https://github.com/AlessandroChecco/all-that-glitters-is-gold`.

other hand, we noticed that increasing worker retention through, for example, better task design might be a win-win solution that could also make crowdworkers more satisfied and perform better. Other countermeasures that could be explored are the use of programmatic gold questions creation and alternative ways of serving gold questions to interfere with the inference mechanism.

Finally, we discussed the economic and sociological implications of these kind of attacks where we pointed out the positive repercussions on the future of crowdwork 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, & Kamar, 2014; Difallah et al., 2012), and the worker behaviour with respect to the local tuning of the plugin and the consequent risk level. 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.

Acknowledgments

This project is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 732328 and by the Australian Research Council Discovery Project DP190102141.

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*, pp. 901–909. VLDB Endowment.

Benoit, K., Conway, D., Lauderdale, B. E., Laver, M., & 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., & Paul, M. (2011). Getting expert quality from the crowd for machine translation evaluation. *Proceedings of the 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., & Latorre, J. (2011). Crowdsourcing preference tests, and how to detect cheating. In *Twelfth Annual Conference of the International Speech Communication Association*.

Checco, A., Bates, J., & Demartini, G. (2018). All that glitters is gold—an attack scheme on gold questions in crowdsourcing. In *Sixth AAAI Conference on Human Computation and Crowdsourcing*.

Clough, P., Sanderson, M., Tang, J., Gollins, T., & 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., & Allahbakhsh, M. (2018). Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Surveys*, *51*(1), 7:1–7:40.

Deleuze, G. (1992). Postscript on the societies of control. *October*, *59*, 3–7.

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

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

El Maarry, K., & Balke, W.-T. (2018). Quest for the gold par: Minimizing the number of gold questions to distinguish between the good and the bad. In *Proceedings of the 10th ACM Conference on Web Science*, WebSci '18, pp. 185–194, New York, NY, USA. ACM.

Ettlinger, N. (2016). The governance of crowdsourcing: Rationalities of the new exploitation. *Environment and Planning A: Economy and Space*, *48*(11), 2162–2180.

Foucault, M. (1991). *Discipline and punish: The birth of the prison*. Penguin.

Fraley, C., & 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., & 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*, CHI '15, pp. 1631–1640, New York, NY, USA. ACM.

Galloway, A. R., & Thacker, E. (2007). *The exploit: A theory of networks*, Vol. 21. U of Minnesota Press.

Graham, M., Hjorth, I., & Lehdonvirta, V. (2017). Digital labour and development: impacts of global digital labour platforms and the gig economy on worker livelihoods. *Transfer: European Review of Labour and Research*, *23*(2), 135–162.

Gray, M. L., & Suri, S. (2019). *Ghost Work: How to Stop Silicon Valley from Building a New Global Underclass*. Eamon Dolan Books.

Haggerty, K. D., & Ericson, R. V. (2000). The surveillant assemblage. *The British Journal of Sociology*, *51*(4), 605–622.

Holland, P. J., Cooper, B., & 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., & Fu, W.-T. (2013). Enhancing reliability using peer consistency evaluation in human computation. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pp. 639–648. ACM.

Ipeirotis, P. G., Provost, F., & Wang, J. (2010). Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pp. 64–67. ACM.

Irani, L. C., & Silberman, M. (2013). Turkopticon: Interrupting worker invisibility in amazon mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 611–620. ACM.

Kaplan, T., Saito, S., Hara, K., & Bigham, J. P. (2018). Striving to earn more: a survey of work strategies and tool use among crowd workers. In *Sixth AAAI Conference on Human Computation and Crowdsourcing*.

Kittur, A., Chi, E. H., & Suh, B. (2008). Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 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., & 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*, pp. 248–256. ACM.

Le, J., Edmonds, A., Hester, V., & 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*, pp. 21–26.

Lehdonvirta, V. (2018). Flexibility in the gig economy: managing time on three online piecework platforms. *New Technology, Work and Employment*, *33*(1), 13–29.

Mann, S., Nolan, J., & Wellman, B. (2002). Sousveillance: Inventing and using wearable computing devices for data collection in surveillance environments.. *Surveillance & Society*, *1*(3), 331–355.

Martin, D., Hanrahan, B. V., O'Neill, J., & Gupta, N. (2014). Being a turker. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pp. 224–235. ACM.

Martin, D., O'Neill, J., Gupta, N., & Hanrahan, B. V. (2016). Turking in a global labour market. *Computer Supported Cooperative Work (CSCW)*, *25*(1), 39–77.

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., & 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*, pp. 2271–2282. ACM.

Moore, P. V. (2017). *The quantified self in precarity: Work, technology and what counts*. Routledge.

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

Qarout, R. K., Checco, A., & Demartini, G. (2016). The effect of class imbalance and order on crowdsourced relevance judgments. *arXiv preprint arXiv:1609.02171*.

Sadowski, C., & Levin, G. (2007). Simhash: Hash-based similarity detection. *Technical report, Google*.

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*, pp. 1621–1630. ACM.

Sandford, R. (2006). Digital post–colonialism. *Flux*.

Saner, E. (2018). Employers are monitoring computers, toilet breaks – even emotions. Is your boss watching you?. `https://www.theguardian.com/world/2018/may/14/is-your-boss-secretly-or-not-so-secretly-watching-you`. [Online; accessed 2019, Jan 27].

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

Shah, N. B., Balakrishnan, S., & Wainwright, M. J. (2016). A permutation-based model for crowd labeling: Optimal estimation and robustness. *arXiv preprint arXiv:1606.09632*.

Snow, R., O'Connor, B., Jurafsky, D., & 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*, pp. 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., & Frank, E. (2003). Statistical models for prognostication. *Symptom Research: Methods and Opportunities. Bethesda, MD: National Institutes of Health*.

Venanzi, M., Guiver, J., Kazai, G., Kohli, P., & Shokouhi, M. (2014). Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pp. 155–164. ACM.

Vorvoreanu, M., & 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*.

Yeginsu, C. (2018). If Workers Slack Off, the Wristband Will Know. (And Amazon Has a Patent for It.). `https://www.nytimes.com/2018/02/01/technology/amazon-wristband-tracking-privacy.html`. [Online; accessed 2019, Jan 27].

Yin, M., Gray, M. L., Suri, S., & Vaughan, J. W. (2016). The communication network within the crowd. In *Proceedings of the 25th International Conference on World Wide Web*, pp. 1293–1303. International World Wide Web Conferences Steering Committee.