Quality Control Attack Schemes in Crowdsourcing*

Alessandro Checco[1†], **Jo Bates**[1] and **Gianluca Demartini**[2]
[1]The University of Sheffield, UK
[2]The University of Queensland, Australia
{a.checco, jo.bates}@sheffield.ac.uk, g.demartini@uq.edu.au

Abstract

An important precondition to build effective AI models is the collection of training data at scale. Crowdsourcing is a popular methodology to achieve this goal. Its adoption introduces novel challenges in data quality control, to deal with under-performing and malicious annotators. One of the most popular quality assurance mechanisms, especially in paid micro-task crowdsourcing, is the use of a small set of pre-annotated tasks as gold standard, to assess in real time the annotators quality. In this paper, we highlight a set of vulnerabilities this scheme suffers: a group of colluding crowd workers can easily implement and deploy a decentralised machine learning inferential system to detect and signal which parts of the task are more likely to be gold questions, making them ineffective as a quality control tool. Moreover, we demonstrate how the most common countermeasures against this attack are ineffective in practical scenarios. The basic architecture of the inferential system is composed of a browser plug-in and an external server where the colluding workers can share information. We implement and validate the attack scheme, by means of experiments on real-world data from a popular crowdsourcing platform.

1 Introduction

Micro-task paid crowdsourcing is a popular solution to perform manual data labelling at scale, but it requires specialised techniques to deal with under-performing or potentially malicious annotators [Daniel *et al.*, 2018]. The most popular technique for quality control in crowdsourcing, adopted for its relative ease of deployment and its effectiveness, is the use of *gold questions*: a set of questions with known ground truth answers [Le *et al.*, 2010; Huang and Fu, 2013] that are used to monitor the accuracy of crowd workers and to identify low quality ones that can be potentially blocked from future labelling tasks. The cost of building such gold set is often non-negligible: they need to be tailored to the specific dataset (to be indistinguishable from non-gold questions) and they should not repeat [Oleson *et al.*, 2011] across multiple tasks not to be easily identified by workers. Moreover, this cost should be added to the cost of rewarding crowd worker participation.

Quality control in crowdsourcing is a well-known challenge. Other than the use of gold questions, several techniques have been proposed to improve the quality of crowdsourced labels. Most techniques aim at aggregating labels for the same data item collected from multiple workers by learning how to correct low-quality labels [Snow *et al.*, 2008; Ipeirotis *et al.*, 2010]. Another class of approaches looks at worker behaviours and peer feedback to incentivize high-quality contributions [Gadiraju *et al.*, 2015; Dow *et al.*, 2011]. In the area of gold questions, methods have been proposed to automatically create gold questions and to adapt the number of gold questions needed for each worker based on their performances [Oleson *et al.*, 2011; El Maarry and Balke, 2018].

In our work we define and experimentally analyse an attack scheme to gold questions in crowdsourcing. To this end, we make the following assumption:

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

In this work, we show that this inherent limit on the size of the gold set can be exploited by a group of colluding workers to perform an attack on the crowdsourcing quality control mechanism: it is possible to build an inferential system able to detect which questions are most likely to be gold questions.

We will also make the following simplifying assumption.

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 (to avoid workers recognizing them).*

We will relax this assumption in Section 4. We will show that the proposed attack approach is also: *anonymous*, in the sense that the worker does not need to be identified to perform the attack; and *secure*, meaning that the information contained in an annotation task does not need to be circulated to other workers.

[*] This paper is an abridged version, with some additional content, of a paper titled "All That Glitters Is Gold—An Attack Scheme on Gold Questions in Crowdsourcing" published at the HCOMP2018.

[†] Contact Author

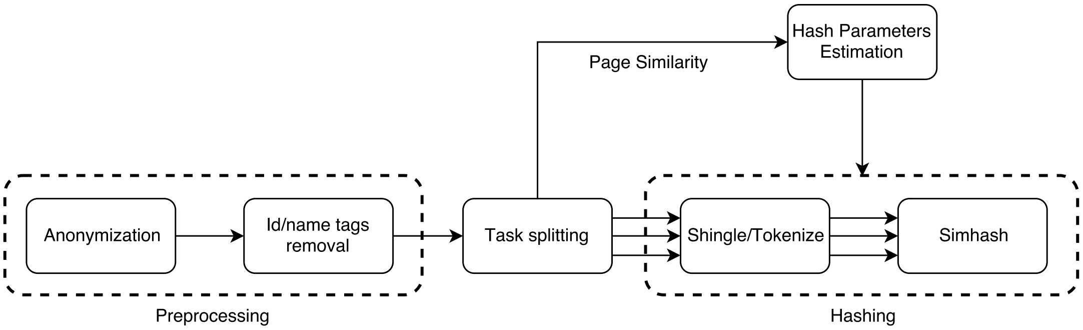

Figure 1: Workflow of the hashing mechanism (from [Checco *et al.*, 2018]).

2 System Architecture

Let us consider a batch (also known as job) of crowdsourcing tasks, that is, a sequence of data labelling tasks sharing the same template where only the data item differs (e.g., a batch of image annotation tasks where, for each image, we require workers to count how many dogs are present in the image). We assume a subset of crowd workers collude to defy the gold question mechanism. Workers access the crowdsourcing job via a web browser, that will show one or more tasks per page.

Each colluding crowd worker runs the task on a browser, where a local browser plugin is able to parse the HTML of the crowdsourcing page, to create a fingerprint of the tasks displayed in the browser by means of a hash function over HTML code. The computed set of hashes, together with the ID of the specific batch, is sent to the external server, that in turn will employ an inference technique to update the global understanding of that batch and signal back to the worker an estimation of the likelihood of each task being a gold question.

Figure 1 shows the usage workflow of the browser plugin, run on the worker side. Whenever the Document Object Model (DOM) of the crowdsourcing page change significantly, the plugin performs the following operations: (i) anonmymisation, to strip out the worker ID and other identifying information; (ii) ID/name tags removal, to remove unique or dynamically created tags, while preserving important HTML tags like "src" that will help the fingerprinting; (iii) task splitting, using an heuristic to separate different tasks in a web page; (iv) shingle/tokenization and simhashing to generate a fingerprint [Sadowski and Levin, 2007] of each crowdsourcing task in the page. The use of simhashes guarantees a secure, fast and scalable transmission of the fingerprints.

The clustering process works as follows. The server keeps a repository of triples (Job ID; simhash; multiplicity), where multiplicity is the count of each simhash appearing in the collected data. The Manhattan distance between the bit representation of the simhases is used to generate a clustering, to group together simhashes that are likely to belong to the same task. Each cluster will now represent a task, and its multiplicity is equal to the times that fingerprint cluster has been reported by the set of colluding workers.

Because of Assumption 1, we expect that the cluter multiplicity will follow a bimodal distribution: gold questions will have a higher multiplicity than non-gold questions. A Gaussian mixture model with two modes is now able to establish the likelihood of each task being a gold question. After a transient phase in which task frequency data is being collected, the inferential system will be able to classify correctly into gold or not-gold the majority of the tasks.

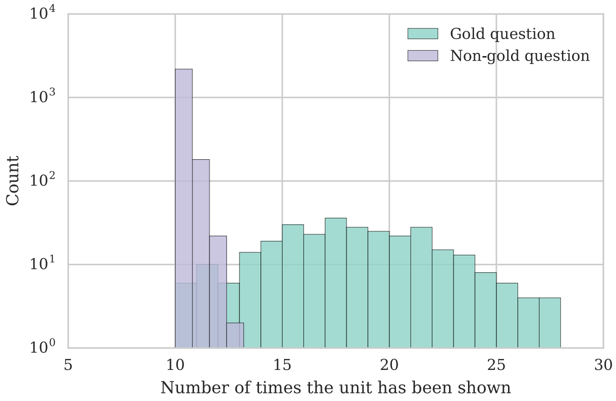

Figure 2: Distribution of the multiplicity for gold and non-gold questions in the CSTA task when using 12.4% of gold questions. Assumptions 1 and 2 are satisfied (from [Checco *et al.*, 2018]).

3 Experimental Results

We simulate the proposed attack scheme over the CSTA datasets and task logs described in [Benoit *et al.*, 2016][1], that contains 29,594 judgements from 336 workers, including timestamps and worker answers. As shown in Figure 2, the gold and non-gold questions form a bimodal distribution. However, the two distributions overlap, making this dataset a challenging candidate for our attack. The ratio of gold questions available in this dataset if 12.4%: we will subsample uniformly at random from this set when we want to simulate smaller gold sets. The baseline result in terms of crowd worker accuracy and efficiency is obtained with the answers provided by the workers. The results obtained by our technique assumes the following worker behaviour: the worker only spends time in answering questions that are signalled as

[1] We used the job in the repository with ID f269506, available from https://github.com/kbenoit/CSTA-APSR.

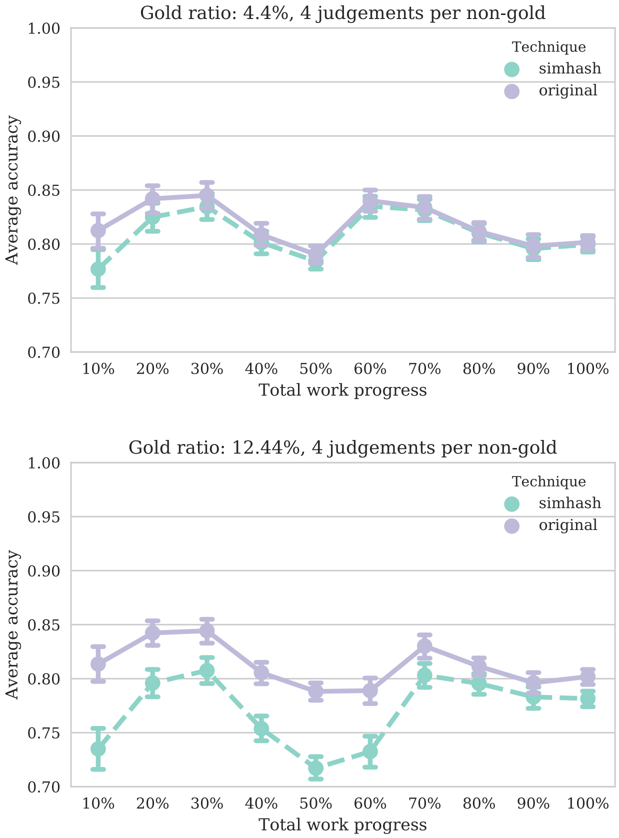

Figure 3: Average worker accuracy for the original worker labels and for the proposed method, with 4 judgments for each non-gold question. On the top row we used a number of gold questions equal to 4.4%, on the bottom row 12.4% (from [Checco *et al.*, 2018]).

potential gold, while they will answer randomly to any questions that is not signalled by the plugin. This is an extreme behaviour, where the worker completely trusts the plugin and attempts to perform a very aggressive attack, to extract the maximum value from the job.

We compute the worker accuracy and time spent per page, for a realistic value of 4.4% of gold questions and for the rather extreme case of 12.4% gold questions in a batch. As shown in Figure 3, more time is required for the inferential system to gain high detection accuracy when the the gold ratio is high, and after a transient phase the accuracy of the worker is very close to the baseline approach where workers carefully answer all questions.

More importantly, the time saved by a worker performing the attack, as shown in Figure 4, is consistently high, allowing the worker to gain a hourly wage that is up to five times higher than the baseline non-attack approach: the worker will be able to ignore most of the questions, because the plugin will provide a signal for the ones that are used to track the worker accuracy, disrupting completely the quality assurance mechanism based on gold questions.

3.1 Number of Colluding Workers

We now relax our assumption that all N workers are colluding, by considering the case in which only $M < N$ workers are using the proposed attack scheme, and assuming that workers have equal retention and enter the job at times distributed uniformly at random.

When not all workers are colluding, there will be a set of gold questions that are under-reported in the inferential system as they have been presented to non-colluding workers. However, if the gold questions are sampled randomly from the gold set (Assumption 2), this scenario is equivalent to having the under-reported gold questions because those questions have not been shown yet to any worker. For this reason M colluding workers can at best expect to reach, at the end of the batch, an accuracy that is equivalent to the one obtained when all workers N are colluding but only $\frac{M}{N}$ of the batch has been completed. Importantly, the presence of non-colluding workers is not affecting in any other way the inference mechanism.

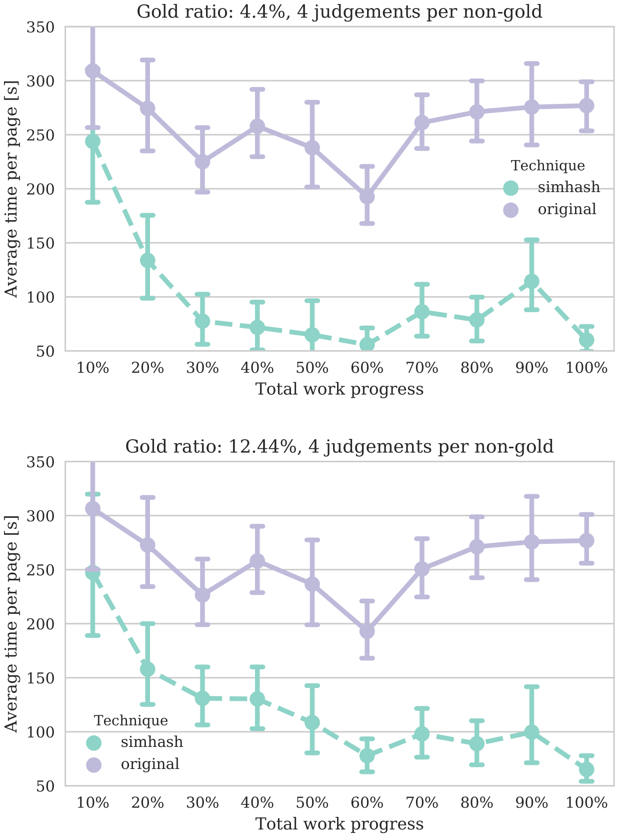

Figure 4: Average time spent per page for original and proposed method, with 4 judgements per non-gold question. On the top row we used a number of gold questions equal to 4.4%, on the bottom row 12.4% (from [Checco *et al.*, 2018]).

4 Countermeasures

There are many countermeasures that crowdsourcing requesters could employ to mitigate the effects of the attack scheme proposed in our work. In this section we analyse here the most promising ones.

Gold set size. Increasing the gold size set is the most obvious countermeasure. However, this choice will significantly increase the cost of the crowdsourcing project. In [Clough *et al.*, 2013] it has been estimated that the cost of generating gold questions for the relevance assessment problem is above four times the UK minimum wage. In our experiment, this would mean that moving from a gold set size of 4.4% to 12.4% (as shown in Figure 4) would require an additional 54% of the crowdsourcing cost already undertaken to label the dataset.

Number of judgements. Increasing the number of judgements required per non-gold question seems to have low/-moderate effect (as shown in more detail in [Checco *et al.*, 2018]). In this case as well, it is necessary to consider the additional cost required for such countermeasure.

Worker retention. As shown in [Checco *et al.*, 2018], having a relatively small number of prolific workers can significantly reduce the strength of this attack, as the total number of gold questions and the number of repeated ones will be lower because each worker should not see repeated gold questions. This solution is promising because increasing crowd worker retention on a batch can at the same time improve the task quality thanks to learning effects [Difallah *et al.*, 2014], but could also reduce the independence of the judgments and increase the labelled dataset bias given by the dominating presence of certain annotators.

Non uniform selection from the gold set. We can relax Assumption 2 by envisioning the case in which the crowdsourcing platform uses a smarter approach when serving gold questions, taking into account the possible presence of this attack scheme. We repeated the experiment described before by serving, at each step, the least seen question from the whole pool of gold questions, keeping the constraint of not showing the same gold question twice to a worker. Surprisingly, the difference in accuracy between this approach and the one under Assumption 2 is less than 2.5 %, and the difference in time spent is of the order of seconds: these differences are not statistically significant. The reason for the failure of this countermeasure could be explained by the relatively small size of gold sets. Thus, a uniform serving is statistically indistinguishable from a lexicographic serving. An alternative approach is to exploit the inner mechanics of the Gaussian Mixture model by serving some gold questions a high number of times to throw off the calculated threshold, but this could lead to manual detection by workers directly.

Programmatic gold questions. Carefully modifying the way in which questions are rendered and using always different gold questions that are programmatically generated [Oleson *et al.*, 2011] could result in tasks with sufficiently distant simhases and lead to gold question being undetected by the attack scheme. This would however increase the setup and design cost.

Additional quality controls. Punishing workers that are too fast, and similar additional quality controls would make this attack harder, at the cost of risking to increase the number of the so-called *gold preys*: legitimate workers that are faster than average and thus unfairly punished [Gadiraju *et al.*, 2015].

Constant Number of Gold Questions. A promising alternative to the classic paradigm of quality assessment is using deep Bayesian trust techniques [Goel and Faltings, 2019] to infer the quality of workers based on their answer similarity. This approach is especially suitable for large scale tasks.

5 Conclusions

In this paper we have presented an attack scheme to quality control mechanisms commonly used in paid micro-task crowdsourcing platforms like, e.g., Amazon MTurk. We focus on the use of gold questions, that is, crowdsourcing tasks for which the correct answer is known which are used to detect low quality workers by controlling the accuracy of the labels they provide over gold questions. The proposed attack scheme relies on a group of colluding crowd workers who make use of an inferential system based on a browser plugin and an external server. The attack method exploits the limited size of the gold question set and the need to serve the same gold questions to multiple workers in the crowd. By sharing information about which questions have been observed by the colluding workers, it is possible to infer which parts of a crowdsourcing job are more likely to be gold questions.

In our experimental results we observed how the proposed method is robust to randomisation and automatic generation of gold questions[2]. Our results also show that crowd workers participating to this attack can obtain high accuracy on gold questions (thus not being identified as low-quality workers) and, at the same time, complete the task much more efficiently thus consistently increasing their hourly wage on the crowdsourcing platform.

We additionally discussed potential countermeasures that crowdsourcing requesters may use to deal with such an attack scheme. These include the increase of the gold question set size which may be, however, infeasible due to the high cost of generating gold questions. A better alternative is the increase of crowd worker retention on the batch by making them complete more tasks which may, on the other hand, introduce bias in the labelled dataset. Possible future extensions of the proposed attack scheme include sharing additional information like, for example, the answer to gold questions which would increase the efficiency benefit for colluding crowd workers.

Acknowledgments

This project is supported by the EU's H2020 research and innovation programme under grant agreement No. 732328 and by the Australian Research Council (ARC) Discovery Project scheme (DP190102141).

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

References

[Benoit *et al.*, 2016] Kenneth Benoit, Drew Conway, Benjamin E Lauderdale, Michael Laver, and Slava Mikhaylov. Crowd-sourced text analysis: Reproducible and agile production of political data. *American Political Science Review*, 110(2):278–295, 2016.

[Checco *et al.*, 2018] Alessandro Checco, Jo Bates, and Gianluca Demartini. All That Glitters Is Gold – An Attack Scheme on Gold Questions in Crowdsourcing. In *Sixth AAAI Conference on Human Computation and Crowdsourcing*, 2018.

[Clough *et al.*, 2013] Paul Clough, Mark Sanderson, Jiayu Tang, Tim Gollins, and Amy Warner. Examining the limits of crowdsourcing for relevance assessment. *IEEE Internet Computing*, 17(4):32–38, 2013.

[Daniel *et al.*, 2018] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Comput. Surv.*, 51(1):7:1–7:40, January 2018.

[Difallah *et al.*, 2014] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, and Philippe Cudré-Mauroux. Scaling-up the crowd: Micro-task pricing schemes for worker retention and latency improvement. In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

[Dow *et al.*, 2011] Steven Dow, Anand Kulkarni, Brie Bunge, Truc Nguyen, Scott Klemmer, and Björn Hartmann. Shepherding the crowd: managing and providing feedback to crowd workers. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 1669–1674. ACM, 2011.

[El Maarry and Balke, 2018] Kinda El Maarry and Wolf-Tilo Balke. 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, pages 185–194, New York, NY, USA, 2018. ACM.

[Gadiraju *et al.*, 2015] Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. 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, pages 1631–1640, New York, NY, USA, 2015. ACM.

[Goel and Faltings, 2019] Naman Goel and Boi Faltings. Deep Bayesian Trust: A Dominant and Fair Incentive Mechanism for Crowd. 2019.

[Huang and Fu, 2013] Shih-Wen Huang and Wai-Tat Fu. Enhancing reliability using peer consistency evaluation in human computation. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 639–648. ACM, 2013.

[Ipeirotis *et al.*, 2010] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010.

[Le *et al.*, 2010] John Le, Andy Edmonds, Vaughn Hester, and Lukas Biewald. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *SIGIR 2010 workshop on crowdsourcing for search evaluation*, pages 21–26, 2010.

[Oleson *et al.*, 2011] David Oleson, Alexander Sorokin, Greg P Laughlin, Vaughn Hester, John Le, and Lukas Biewald. Programmatic Gold: Targeted and Scalable Quality Assurance in Crowdsourcing. *Human computation*, 11(11), 2011.

[Sadowski and Levin, 2007] Caitlin Sadowski and Greg Levin. Simhash: Hash-based similarity detection, 2007.

[Snow *et al.*, 2008] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 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*, pages 254–263. Association for Computational Linguistics, 2008.