How Crowdsourcable is Your Task?

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

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

ABSTRACT

Lately, crowdsourcing has become an accepted means of creating resources for tasks that require human intelligence. Information Retrieval and related fields frequently exploit it for system building and evaluation purposes. However, malicious workers often try to maximise their financial gains by producing generic answers rather than actually working on the task. Identifying these individuals is a challenging process into which both crowdsourcing providers and requesters invest significant amounts of time. In this work we aim to identify measures that we can take to make a crowdsourced task more resistant to fraudulent attempts.

1. INTRODUCTION

In the history of Information Retrieval and related areas such as artificial intelligence, machine translation, document summarization, etc. researchers and engineers have always relied on human notions of correctness for system building and evaluation. The field of Information Retrieval depends on large scale data collections to best simulate system behaviour on the massive amounts of data on the Internet. An example is the series of extensive corpora created by the well-known *Text Retrieval Conference* (TREC) [8]. The manual creation of these resources typically requires great amounts of time and money. Recently, we have seen some advances into automatically creating or extracting resources for scientific corpora [16, 13]. However, for many applications that require high precision, human judgements are still necessary [15]. Especially in newly explored research directions, without existing evaluation data, the demand for new resources becomes apparent. A novel way of satisfying the need for large collections of human-annotated data was presented in late 2005. Amazon Mechanical Turk [2] offers a platform on which task requesters can reach a large number of freelance employees to solve *human intelligence tasks* (HITs). The payment is typically done on micro level, e.g., $ 0.01 per quickly solvable HIT. This process, known as crowdsourcing, is now widely accepted and represents the basis for validation in many recent research publications [5, 12]. With growing popularity of crowdsourcing platforms, the group of workers has become more diverse. In the beginning many workers fulfilled tasks out of interest or boredom, with the payment being only a minor attraction. Nowadays, the number of users who are exclusively attracted by the monetary reward represents a significant share of the crowd's workforce [6]. As a consequence of this development one can observe a high number of malicious users who try to finish HITs as quickly as possible in order to maximize their profit. This results in large proportions of crowd judgements being generic arbitrary answers. Consequently, research work based on crowdsourcing nowadays has to pay careful attention to the resulting data quality.

There are two main approaches in this respect: (1) The use of high quality gold standard data or inter-annotator agreement ratios to check on and if necessary reject malicious workers. (2) The task can often be designed in such a way that it becomes less attractive for scammers. Based on the experience gained from several previous crowdsourcing tasks and a number of dedicated experiments, this work aims to quantify the share of malicious workers as well as to identify criteria and methods to make tasks more robust against this new form of judgement taint.

The remainder of this work is structured as follows: Section 2 gives an overview of related work in the domain of crowdsourcing. In Section 3, we analyse commonly observed scam strategies in crowdsourcing environments. Section 4 describes a number of experiments that were conducted in order to measure the current extent of crowdsourcing scam as well as the effectiveness of various counter measures. Finally, Section 5 concludes with a summary of our findings and an outlook on future directions of mitigating the effect of malicious workers.

2. RELATED WORK

Although crowdsourcing has become a frequently used means of creating scientific resources the research community only lately began dedicating research work to crowdsourcing performance evaluation and methodology. In 2008, Sorokin et al. [19] conducted a feasibility study of the applicability of crowdsourcing for image annotation. Their task was to identify people in images. Over a series of experiments they varied the reward per task and studied the quality of the results. They identified a strong dependency between the amount of reward and resulting quality. While extremely low rewards led to slow task uptake and generally fewer interested workers, very high rewards were found to attract more inefficient and malicious workers. In the same year Kittur et al. [12] published their findings about the importance of task formulation to obtaining good results. Their main conclusion was that a task should be given in such a way, that cheating takes approximately the same

WSDM 2011 Workshop on Crowdsourcing for Search and Data Mining (CSDM 2011), Hong Kong, China, Feb. 9, 2011

time as faithfully completing it. The authors additionally underline the importance of clearly verifiable questions in order to reject malicious users.
Throughout the following year various groups of researchers investigated the reliability of non-expert judgements for natural language applications such as paraphrasation, translation or sentiment analysis [18, 9]. They find that a single expert in the majority of cases is more reliable than a non-expert. However, using an aggregate of several cheap non-expert judgements approximates the performance of expensive expertise. The same tendency was observed by Alonso et al. [4] for TREC-like relevance judgement tasks. Also in 2009, Little et al. [14] released TurkIt, a framework for iterative programming of crowdsourcing tasks. In their evaluation, the authors mention relatively low numbers of malicious users. This finding is somewhat conflicting with most publications in the field, that report higher figures. We suspect that there is a strong connection between the type of task at hand and the share of malicious users attracted to it. In this work we will carefully investigate this dependency.

3. HOW TO CHEAT

In this section we will give an overview of scam methods described in related work, discussed on-line (e.g., in blogs), and encountered in our own research. Keeping these in mind, we will later on evaluate various strategies of countering their efforts to make crowdsourced research tasks more robust.
The general assumption that underlies most scamming attempts is the perceived anonymity of single workers within the massive workforce of the crowd. As we will see, most methods are rather straightforward and easy to detect when inspected by a human assessor. Within a large-scale batch of HITs, however, identifying cheaters becomes more challenging. To better understand worker motivation, one should realise the circumstances under which they connect to the crowdsourcing platform. Besides the shrinking share of exclusively recreational workers who are driven by actual interest in novel HITs, there is a growing number of workers who depend on the financial reward [11]. The latter group is responsible for a significant share of crowdsourcing scam. We noticed an interesting tendency when running a HIT that involved filling a survey with personal information. For this HIT we received multiple submission by some workers that contained largely contradictory details about age, marital status, or origin. We suspect these workers are organised in large offices from where multiple individuals connect to the crowdsourcing platform under the same worker id.
In the following we will group the various adversarial methods by the type of task that they are targeted towards.

Closed Class Questions

A frequently used class of HITs require the worker to make one or more choices from a range of possible answers. They are typically represented as radio buttons, check boxes or sliders. For HITs of this type we can commonly observe two classes of cheating strategies: (1) Giving arbitrary answers, either in a uniform (check all / check none) or truly random fashion, is one of the most frequent methods. They can often be rejected using high quality gold standard data or annotator agreement over redundant HITs. (2) Workers who actually spend time to think about the task and subsequently issue a number of educated guess answers are very hard to detect as they will largely agree with the gold standard as well as the majority of the crowd. An example of this approach can be found in the well-known task of relevance judgements between documents and queries. A worker who judges everything as irrelevant will be right in most cases. This method is traditionally countered by issuing a number of very easy gold standard tasks that are unambiguous for users who actually answer the task. Failing to complete any of these tasks results in an immediate rejection of the user.

Open Class Questions

Often HIT designers include open questions in the form of free text fields into which the worker types a more detailed reasoning of his decision than the closed class options would allow for. Malicious workers tend to either leave these fields blank (if they are not marked as mandatory) or to copy and paste a generic string of words. The latter approach is automatically detectable if the same string occurs repeatedly. For truly arbitrary free text answers, e.g., copied from a large chunk of unrelated natural language text, this becomes hard to identify.

Internal Quality Control

State of the art crowdsourcing platforms feature internal quality control options in the form of worker reputations. These consist typically of the worker's accuracy on previously submitted HITs. This widely used approach has two potential problems. Firstly the accuracy is exclusively computed through the acceptance rate of HITs. HIT designers often accept all answers and only filter out noise afterwards. The mischievous user, however, already received his increase in accuracy and sets out to complete further HITs.
The second method, so called rank boosting, was presented by Panos Ipeirotis on his weblog [10]. Following this strategy the worker creates a HIT designer account, issues a large number of cheap HITs and immediately completes them with his worker account. While the worker's rank is artificially boosted, this method hardly costs him any money as he loses only the small share that the crowdsourcing platform deducts per HIT.

External Quality Control

During one of our early experiments, we directed the workers to an external web page on which they would complete the actual task and receive a confirmation code to be entered on the original crowdsourcing platform. Despite this openly announced completion check workers tried to issue made-up confirmation codes, to resubmit previously generated codes multiple times or to submit several empty tasks and claim that they did not get a code after task completion. While such attempts are easily fended off, they offer a good display of malicious worker strategies. They will commonly try out a series of naive exploits and move on to the next task if they do not succeed.

4. EXPERIMENTS

After having discussed common adversarial strategies, we dedicate a range of experiments to understanding the extent of scam on crowdsourcing platforms as well as typical criteria of robust tasks. Our experiments are based on two very different HIT types. The first one is a straightforward binary relevance assessment between web pages and queries. The

second task asked the workers to judge web pages according to their suitability for children of different age groups and to fill a brief survey on their experience in guiding children's web search [7]. The experiments were run through Amazon Mechanical Turk [2] and CrowdFlower [3] in the course of the year 2010 and each HIT was issued to 5 independent workers. We inspected a sample of 200 HITs for each of the tasks resulting in a grand total of 2000 HITs.
For both tasks we manually determined whether the worker attempted to properly answer the HIT or whether he cheated. This decision was made based on the following four indicators: (1) The agreement with the gold standard was used to measure the general quality of the answer. (2) Agreement with other workers enabled us to identify hard tasks on which even honest workers occasionally fail. (3) The HIT completion time gave us an estimate of how much effort the worker put into the task. (4) A trick question asked whether the website was written in a non-English language. Mistakes on this question almost invariably identified cheaters, as it is very easy to answer unless the worker did not look at the actual page.[1] Our detailed analysis of worker performance was conducted along three research questions: 1. How does the concrete task type influence the number of malicious workers? 2. Does interface design effect the share of malicious workers? 3. Can we reduce fraudulent tendencies by a priori filtering the crowd?

4.1 Task-dependent Evaluation

The fundamental differences between our two experimental tasks are complexity and novelty. Relevance judgements are relatively straightforward to create and are one of the best-known applications of crowdsourcing as many IR projects depend on them. The web page suitability survey, on the other hand, was a novel task that requires more creativity and consideration. In this section we investigate the degree to which the task type influences the share of attracted cheaters. Please note that comparing absolute worker performance in this case is not meaningful due to the different task-inherent difficulty and ambiguity. Table 1 shows the share of malicious workers for both tasks with and without using gold standard data. The only qualification that was required in this example was an acceptance rate of 95% of the worker's previous HITs (the default setting). We could note a significantly higher proportion of malicious users for the well-known relevance assessment task. Introducing a gold standard set decreased the number of malicious users by a comparable amount (23.7% and 25%). With respect to our first research question, we conclude that higher task complexity drastically discourages malicious workers from attempting to cheat. The more creativity and consideration a task requires the less attractive it seems to be for workers who simply want to exploit it. For the further experiments we will concentrate exclusively on the relevance assessment task as it features a significantly higher share of malicious workers so that the effect of our measures is assumed to be visible more clearly.

[1]The suggestion of this trick resulted from personal communication between the authors, Mark Sanderson and William Webber.

Table 1: Task-dependent share of malicious workers with and without using gold standard data.

Task	before gold	after gold
Suitability	2.0%	1.5%
Relevance	38.0%	29.0%

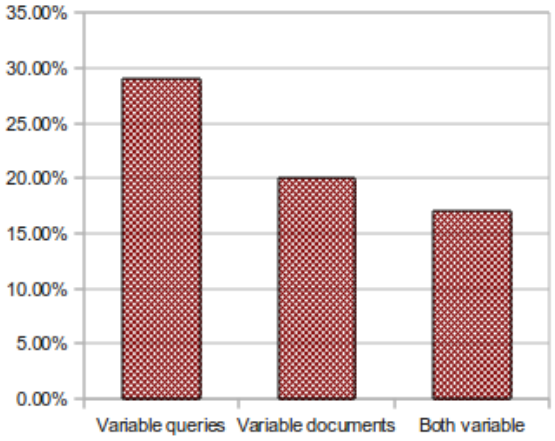

Figure 1: Interface-dependent share of malicious workers for variable queries, variable documents and fully variable pairs.

4.2 Interface-dependent Evaluation

In classical interface design a well-known practice is to reduce context changes for users in order to keep them focused and enable them to work efficiently [17]. Efficient task completion, however, is an explicit aim of financially driven workers. We quantify this notion at the example of our relevance assessment task. Figure 1 shows the results of this comparison. In the first step, we present the workers batches of 10 web page/query pairs using gold standard data. In order to keep the number of context changes to a minimum we asked the workers to visit a single web page and afterwards create relevance judgements for that page given 10 different queries. The resulting share of malicious workers turns out to be very high (29%). In a second step, we kept the query constant and asked the workers for relevance judgements of 10 different web sites. While in a controlled environment with trusted annotators this step would be counter productive, we see a significant decline of 31% in the number of scammers as the task requires opening 10 distinct web pages which makes it less easily repeatable. Finally, we issued batches with randomly drawn query/document pairs. As a result the number of malicious workers decreased by another 15%. With respect to our second research question, we find that greater variability and more context changes discourage malicious workers as the task appears less susceptible to automation or cheating, in other words less profitable.

4.3 Audience-dependent Evaluation

The final dimension of our evaluation is the composition of the underlying crowd of workers. We previously assumed that primarily money-driven workers tend to be malicious more often than those who mainly seek distraction. The

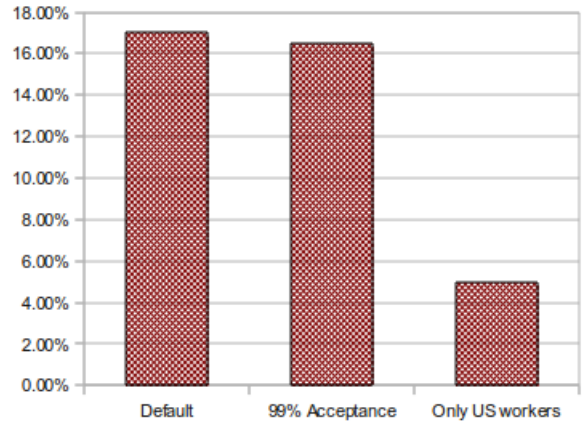

Figure 2: Crowd-dependent share of malicious workers filtered by previous acceptance rate and origin.

baseline of this comparison is the performance with variable query/document pairs and gold standard data. Figure 2 shows how the share of malicious workers shrinks by another 71% when exclusively admitting workers from developed countries as for example the USA. This gain however comes at a cost. The completion time for the full batch increased from several hours to almost one week, since many US workers were not interested in the rather straightforward task. In an additional experiment we raised the threshold acceptance rate for workers from 95% to 99%. Figure 2 shows that this requirement hardly influences the rate of malicious workers.

The conclusion for our third research question is twofold: (1) We have seen how prior crowd filtering can greatly reduce the number of malicious workers. This narrowing down of the workforce may however result in longer completion times. (2) Additionally, we could confirm the assumption that a worker's previous task acceptance rate can not be seen as a stand-alone predictor of his reliability.

5. CONCLUSION

In this work we inspected the commonly observed methods of malicious crowdsourcing workers and attracting/ discouraging factors of HITs. Based on a range of experiments, we conclude that malicious workers are less frequently encountered in novel tasks that involve a degree of creativity and abstraction. While there are various means of identifying forged submissions, setting tasks up in a non-repetitive way and requiring creative input can greatly increase the share of faithful workers.

Crowd filtering by worker origin has been shown to have significant impact on the share of malicious users. However, we are convinced that implicit crowd filtering based on task design is a more promising method. If we can discourage malicious workers of any origin from becoming interested in our task that is clearly preferable to a priori excluding more than 80% of the world's population from accessing the HIT. Future directions for preserving the quality of crowdsourcing for research purposes should include the development of a more sophisticated worker grading system than just prior acceptance rate. Aspects such as the types of tasks that the worker submitted previously might be of great value with regard to this. A potential measure could be the frequency distribution of certain input types (e.g., check boxes vs. free text fields). Workers who never complete tasks that require free writing or even more complex operations may for example have a higher likelihood of bearing malicious intent as they are particularly efficiency-driven. In our opinion, understanding worker behaviour better will serve for improved reliability metrics.

Acknowledgements

This research is part of the PuppyIR project [1]. It is funded by the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement no. 231507.

6. REFERENCES

[1] PuppyIR: An Open Source Environment to Construct Information Services for Children. http://www.puppyir.eu.
[2] Amazon Mechanical Turk - Artificial Artificial Intelligence. https://www.mturk.com/, 2010.
[3] CrowdFlower - Harness the advantages of crowdsourcing. http://www.crowdflower.com, 2010.
[4] O. Alonso and S. Mizzaro. Can we get rid of TREC assessors? Using Mechanical Turk for relevance assessment. In *Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation*, pages 15–16. Citeseer, 2009.
[5] V. Ambati, S. Vogel, and J. Carbonell. Active learning and crowd-sourcing for machine translation. In *LREC 2010*.
[6] A. Baio. The Faces of Mechanical Turk. http://waxy.org /2008/11/the_faces_of_mechanical_turk/, 2008.
[7] C. Eickhoff, P. Serdyukov, and A.P. de Vries. Web Page Classification on Child Suitability. In *CIKM 2010*.
[8] DK Harman. *First text retrieval conference (TREC-1): proceedings*. DIANE Publishing, 1993.
[9] P.Y. Hsueh, P. Melville, and V. Sindhwani. Data quality from crowdsourcing: a study of annotation selection criteria. In *NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*.
[10] P. Ipeirotis. Be a Top Mechanical Turk Worker: You Need $5 and 5 Minutes. http://behind-the-enemy-lines.blogspot .com/2010/10/be-top-mechanical-turk-worker-you -need.html, 2010.
[11] P.G. Ipeirotis. Demographics of mechanical turk.
[12] A. Kittur, E.H. Chi, and B. Suh. Crowdsourcing user studies with Mechanical Turk. In *SIGCHI 2008*.
[13] G.W. Lesher and C. Sanelli. A web-based system for autonomous text corpus generation. *ISSAAC 2000*.
[14] G. Little, L.B. Chilton, M. Goldman, and R.C. Miller. Turkit: Tools for iterative tasks on mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 29–30. ACM, 2009.
[15] M.P. Marcus, M.A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):330, 1993.
[16] E. Riloff. Automatically generating extraction patterns from untagged text. In *NCAI 1996*.
[17] B. Shneiderman. *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1997.
[18] R. Snow, B. O'Connor, D. Jurafsky, and A.Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP 2008*.
[19] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. In *CVPRW'08*.