Inf Retrieval (2013) 16:121–137
DOI 10.1007/s10791-011-9181-9

CROWD SOURCING

Increasing cheat robustness of crowdsourcing tasks

Carsten Eickhoff · Arjen P. de Vries

Received: 6 May 2011 / Accepted: 1 December 2011 / Published online: 14 February 2012

Abstract Crowdsourcing successfully strives to become a widely used means of collecting large-scale scientific corpora. Many research fields, including Information Retrieval, rely on this novel way of data acquisition. However, it seems to be undermined by a significant share of workers that are primarily interested in producing quick generic answers rather than correct ones in order to optimise their time-efficiency and, in turn, earn more money. Recently, we have seen numerous sophisticated schemes of identifying such workers. Those, however, often require additional resources or introduce artificial limitations to the task. In this work, we take a different approach by investigating means of a priori making crowdsourced tasks more resistant against cheaters.

Keywords Crowdsourcing · User experiments · Stability · Human factors

1 Introduction

Many scientific fields including information retrieval, artificial intelligence, machine translation or natural language processing rely heavily on large-scale corpora for system building, training and evaluation. The traditional approach to acquiring these data collections is employing human experts to annotate or create the relevant material. A well-known example from the area of information retrieval is the series of extensive corpora created in the context of the NIST's *Text REtrieval Conference* (TREC) (Harman 1993). Since the manual creation of such resources typically requires substantial amounts of time and money there have long been advances into using automatically generated or extracted resources (Riloff 1996; Lesher and Sanelli 2000; Soboroff et al. 2001; Amitay et al. 2004). While there are several promising directions and

C. Eickhoff (✉)
Delft University of Technology, Delft, The Netherlands
e-mail: c.eickhoff@tudelft.nl

A. P. de Vries
CWI, Amsterdam, The Netherlands
e-mail: arjen@acm.org

methods that are reported to correlate well with human judgements, for many applications that require high precision, human judgements are still necessary (Marcus et al. 1993).

Especially in novel or niche areas of research for which there are little or no existing resources that could be re-used, the demand for an alternative way of data acquisition becomes apparent. With the advent of commercial crowdsourcing, a new means of satisfying this need for large-scale human annotations emerged. Starting in 2005, Amazon Mechanical Turk (MTurk 2011) and others provide platforms 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., a few US cents per quickly solvable HIT. This process is now widely accepted and represents the basis for data collection, resource annotation or validation in many recent research publications (Ambati et al. 2010; Kittur et al. 2008).

Over the years crowdsourcing became substantially more popular among both requesters and workers. As a consequence, the relatively small initial crowd of workers that was mainly attracted by the prospect of completing odd or entertaining tasks as a diversion, changed. Nowadays, the number of users who are mainly attracted by the monetary reward represents a significant share of the crowd's workforce (Baio 2008; Kaufmann et al. 2011; Ross et al. 2010). At the same time we observe a significant share of cheaters entering the market. Those workers try to maximise their financial gains by submitting quick generic, non-reflected answers that, in turn, serve for weak or altogether compromised corpus quality. In response to this trend, research work based on crowdsourcing nowadays has to pay careful attention to monitoring result quality. The accepted way of addressing cheat submissions in crowdsourcing is the use of high quality gold standard data or inter-annotator agreement ratios to check on and if necessary reject deceivers. In this work we present an alternative approach by designing HITs that are less attractive for cheaters. Based on the experience gained from several previous crowdsourcing tasks and a number of dedicated experiments, this work aims to quantify the share of deceivers as well as to identify criteria and methods to make tasks more robust against this new form of annotation taint.

The remainder of this work is structured as follows: Sect. 2 gives an overview of related work in the domain of crowdsourcing. In Sect. 3, we analyse commonly observed cheating 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 remedial effect of several design criteria. Finally, Sect. 5 concludes with a summary of our findings and an outlook on future directions in countering cheaters and thus preserving result quality.

2 Related work

Despite the fact that crowdsourcing is being widely used to create and aggregate data collections for scientific and industrial use, the current amount of research work dedicated to methodological evaluations of crowdsourcing is relatively limited. One of the early studies by Sorokin and Forsyth in (2008) investigated the possibility of using crowdsourcing for image annotation.

They found an interesting non-monotonic dependency between the assigned monetary reward per HIT and the observed result quality. While very low pay resulted in sloppy work, gradually increasing the reward improved annotation quality up to a point where further increases even deteriorated performance due to attracting more cheaters. In the same year, Kittur et al. (2008) published their influential overview on 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 time as faithfully completing it. The authors additionally underline the importance of clearly verifiable questions in order to reject deceivers.

In the course of the following year, several independent research groups studied the performance of experts versus non-experts for various natural language processing tasks such as paraphrasation, translation or sentiment analysis (Snow et al. 2008; Hsueh et al. 2009). The unanimous finding, also confirmed by Alonso and Mizzaro (2009), was, that a single expert is typically more reliable than a single non-expert. However, aggregating the results of several cheap non-experts, the performance of an expensive professional can be equalled at significantly lower cost. In the same year, Little et al. (2009) released TurkIt, a framework for iterative programming of crowdsourcing tasks. In their evaluation, the authors mention relatively low numbers of cheaters. 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 type of workers attracted to it. In this work we will carefully investigate this dependency through a series of experiments.

There is a line of work dedicated to studying HIT design in order to facilitate task understanding and worker efficiency. Examples are Khanna et al. (2010)'s investigation of the influence of HIT interface design on Indian workers' ability to successfully finish a HIT or Grady and Lease's (2010) study of human factors in HIT design. The interface-related study in Sect. 4.3 inspects a very different angle by using interface design as a means of making cheating less efficient and therefore less tempting.

We can conclude that there are numerous good publications that detail tailor-made schemes to identify and reject cheaters in various crowdsourcing scenarios. Snow et al. (2008) do not treat cheaters explicitly, but propose modelling systematic worker bias and subsequently correcting for it. For their sentiment analysis of political blog posts, Hsueh et al. (2009) rely on a combination of gold standard labels and majority voting to ensure result quality. Soleymani and Larson 2010) use a two-stage process. In a first round, the authors offer a pilot HIT as recruitment and manually invite well-performing workers for the actual task. Hirth et al. (2010) describe a sophisticated workflow in which one (or even potentially several) subsequent crowdsourcing step is used in order to check on the quality of crowdsourced results. In her recent work, Gabriella Kazai discusses how the HIT setup influences result quality, for example through pay rate, worker qualification or worker type (Kazai 2011; Kazai et al. 2011).

In this article, we take a different approach from the state of the art by (1) Aiming at discouraging cheaters rather than detecting them. While there is extensive work on the posterior identification and rejection of cheaters, we deem these methods sub-optimal as they still bind resources such as time or money. Instead, we try to find out what makes a HIT look appealing to cheaters and subsequently aim to remedy these aspects. (2) While there are many publications *also* detailing the authors' cheater detection schemes, we are not aware of comprehensive works on cheat robustness that are applicable to a wide range of HIT types. By giving a broad overview of frequently encountered adversarial strategies as well as established countermeasures, we hope to close this gap.

3 How to cheat

Before proceeding to our experimentally supported inspection of various cheat countering strategies, we will spend some thought on the nature of cheating on large-scale crowdsourcing platforms. The insights presented in this section are derived from related work, discussions with peers, as well as our own experience as HIT providers. They present, what we believe is an overview of the most frequently encountered adversarial strategies in the commercial crowdsourcing field. While one could theorize about many more potential exploits, especially motivated by the information security domain (e.g., Pfleeger and Pfleeger 2007; Moore et al. 2001), we try to concentrate on giving an account of the main strategies HIT designers have to face regularly.

As we will show in more detail, the cheaters' methods are typically straightforward to spot for humans, but, given the massive HIT volume, such a careful manual inspection is not always feasible. Cheating as a holistic activity can be assumed to follow a breadth-first strategy in that the group of cheating workers will explore a wide range of naive cheats and move on to a different HIT if those prove to be futile. When dealing with cheating in crowdsourcing, it is important to take into consideration the workers' different underlying motivations for taking up HITs in the first place (Ipeirotis 2010b). We believe that there are two major types of workers with fundamentally different motivations for offering their work force. Entertainment-driven workers primarily seek diversion by taking up interesting, possibly challenging HITs. For this group the financial reward plays a minor role. The second group are money-driven workers. These workers are mainly attracted by monetary incentives. We expect the latter group to contain more cheating individuals as an optimization of time efficiency and, subsequently, an increased financial reward, is clearly appealing given their motivation. In this work, we also regard any form of automated HIT submission, i.e., bots, scripts, etc. to originate from money-driven workers. We could get an interesting insight into the organization of the money-driven crowdsourcing subculture when running a HIT that involved filling a survey with personal information (Eickhoff et al. 2011) (See Fig. 4 in the Appendix). For this HIT we received multiple submissions by unique workers that contained largely contradictory statements. We suspect these workers to be organised in large-scale offices from where multiple individuals connect to the platform under the same worker id. While this rather anecdotal observation is not central to our work and demands further evidence in order to be quantifiable, we consider it an interesting one, worth sharing with the research community.

Our following overview of cheating approaches will be organised according to the types of HITs and quality control mechanisms they are aimed at.

3.1 Closed class questions

Closed class questions are probably the most frequently used HIT elements. They require the worker to choose from a limited, pre-defined list of options. Common examples of this category include radio buttons, multiple choice question, check boxes and sliders. There are two widely-encountered cheating strategies targeting closed-class tasks: (1) Arbitrarily picked answers can often easily be rejected by using good gold standard data or by inspecting agreement with redundant submissions by multiple workers, either in terms of majority votes or more sophisticated combination schemes (Dawid and Skene 1979). (2) Some clever cheaters may learn from previous HITs and come up with educated guesses based on the answer distribution underlying the HIT. An example could be the typically sparsely distributed relevance in web search scenarios for which a clever cheater might learn that arbitrarily selecting only a very small percentage of documents closely resembles meaningful judgements. This is often addressed by introducing a number of very easy to answer gold standard awareness questions. A user that fails to answer those questions can be immediately rejected as he is clearly not trying to produce sensible results.

3.2 Open class questions

Open questions allow workers to provide less restricted answers or perform creative tasks. The most common example of this class are text fields but it potentially includes draw boxes, file uploads or similar. Focussing on the widely used text fields, there are three different forms of cheats: (1) Leaving the field blank can be disabled during HIT design. (2) Entering generic text blocks is easily detectable if the same text is used repeatedly. (3) Providing unique (sometimes even domain-specific) portions of natural language text copied from the Web is very hard to detect automatically.

3.3 Internal quality control

Most current large-scale crowdsourcing platforms collect internal statistics of the workers' reliability in order to fend off cheaters. Reliability is, to the best of our knowledge, measured by all major platforms in terms of the worker's share of previously accepted submissions. There are two major drawbacks of this approach: (1) Previous acceptance rates fail to account for the high share of submissions that are uniformly accepted by the HIT provider and are post-processed and filtered, steps, to which the platform's reputation system has no access. (2) Previous acceptance rates are prone to gaming strategies such as rank boosting (Ipeirotis 2010a) in which the worker simultaneously acts as a HIT provider. He can then artificially boost his reliability by requesting and submitting small HITs. This gaming scheme is very cheaply implementable as the cycle only loses the service fee deducted by the crowdsourcing platform.

In addition to these theoretical considerations concerning the shortcomings of current quality control mechanisms, Sect. 4.4 will show an empirical evaluation backing the assumption that we need better built-in quality measures than prior acceptance rates.

3.4 External quality control

Some very interactive HIT types may require more sophisticated technical means than offered by most crowdsourcing platforms. During one of our early experiments, we dealt with this situation by redirecting workers to an external, less restricted 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 deceiver 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 our discussion of adversarial approaches and common remedies, this section will give a quantitative experimental overview of various cheating robustness criteria of crowdsourcing tasks. The starting point of our evaluation are two very different HITs that we originally requested throughout 2010 and that showed substantially different cheat rates. The first task is a straightforward binary relevance assessment between pairs of web pages and queries. The second task asked the workers to judge web pages according to

Table 1 Submission distribution for all HITs

Channel	Absolute	Relative share (%)
Amazon mechanical turk	4285	85
Samasource	454	9
Gambit	303	6
GiveWork	0	0

their suitability for children of different age groups and to fill a brief survey on their experience in guiding children's web search (Eickhoff et al. 2011). Examples of both tasks can be found in the Appendix.

All experiments were run through CrowdFlower[1] in 2010 and 2011. The platform incorporates the notion of "channels" to forward HITS to third party platforms. To achieve broad coverage and results representative of the crowdsourcing market, we chose all available channels, which at that time were Amazon Mechanical Turk[2] (AMT), Gambit[3], SamaSource[4] as well as the GiveWork smartphone application jointly run by Samasource and CrowdFlower. Table 1 shows the overall distribution of received submissions according to the channels from which they originated. The figures are reported across all HITs collected for this study, as there were no significant differences in distribution between HIT types. The vast majority of submissions came from AMT. We are not are not aware of the reason why we did not receive any submissions from the GiveWork app. HITs were offered in units of 10 at a time with initial batch sizes of 50 HITS. Each HIT was issued to at least 5 independent workers. Unless stated differently, all HITs were offered to unrestricted audiences with the sole qualification of having achieved 95% prior HIT acceptance, the default setting on most platforms. The monetary reward per HIT was set to 2 US cents per relevance assessment and 5 US cents per filled web page suitability survey. We did not change the reward throughout this work. Previous work, e.g., by Harris (2011), has shown the influence of different financial incentive models on result quality. Statistical significance of results was determined using a Wilcoxon Signed Rank test with $\alpha < 0.05$.

A key aspect of our evaluation is identifying cheaters. There is a wide range of indicators for this purpose, including: (1) Agreement with trusted gold standard data can be used to measure the general quality of an answer. (2) Agreement with other workers enables us to identify hard tasks on which even honest workers occasionally fail. (3) HIT completion times (either compared per HIT or HIT type) give an estimate of how much effort the worker put into the task. (4) Simple task awareness questions that are correctly and unambiguously answerable for every worker can be introduced to identify distracted or cheating individuals. Mistakes on this kind of question are typically penalized heavily in cheater detection schemes.[5] The concrete scheme chosen in this work will be formally detailed in the following section.

Our analysis of methods to increase cheating robustness was conducted along four research questions: (1) How does the concrete task type influence the number of observed cheaters? (2) Does interface design affect the share of cheaters? (3) Can we reduce

[1] http://www.crowdflower.com

[2] http://www.mturk.com

[3] http://getgambit.com/

[4] http://samasource.org/

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

fraudulent tendencies by explicitly filtering the crowd? (4) Is there a connection between the size of HIT batches and observed cheater rates?

4.1 What qualifies a cheater?

Before beginning our inspection of different strategies to fend off cheaters in crowdsourcing scenarios, let us dedicate some further consideration to the definition of cheaters. Following our previous HIT experience we can extend the worker classification scheme by Kittur et al. (2008) to identify several dysfunctional worker types. *Incapable workers* do not fulfil all essential requirements needed to create high quality results. *Malicious workers* try to invalidate experiment results by submitting wrong answers on purpose. *Distracted workers* do not pay full attention to the HIT which often results in poor quality. The source of this distraction will vary across workers and may be of external or intrinsic nature. The exclusively money-driven cheater introduced in Sect. 3 falls into the third category, as he would be capable of producing correct results but is distracted by the need to achieve the highest possible time-efficiency. As a consequence, we postulate the following formal definition of cheaters for all subsequent experiments in this work:

Definition 1 A cheater is a worker who fails to correctly answer simple task awareness questions or who is unable to achieve better than random performance given a HIT of reasonable difficulty.

In the concrete case of our experiments we measure agreement as a simple majority vote across a population of at least 5 workers per HIT. Disagreeing with this majority decision for at least 50% of the questions asked will flag a worker as a cheater. Additionally, we inject task awareness questions that require the worker to indicate whether the resource in question is written in a non-English language (each set of 10 judgements that a worker would complete at a time would contain one known non-English page). Awareness in this context represents that the worker actually visited the web page that he is asked to judge. Failing to answer this very simple question will also result in being considered a cheater. The cheater status is computed on task level (i.e., across a set of 10 judgements in our setting) in order to result in comparable reliability. Computing cheater status on batch level or even globally would serve for very strict labels as a single missed awareness question would brand someone a cheater even if the remainder of his work in the batch or the entire collection were valuable. Our approach can be considered lenient as cheaters are “pardoned” at the end of each task. Our decision is additionally motivated by the fact that the aim of this study is to gauge the proportion of cheaters attracted by a given HIT design rather than achieving high confidence at identifying individuals to be rejected in further iterations. At the same time, we are confident that a decision based on 10 binary awareness questions and the averaged agreement across 10 relevance judgements produces reliable results that are hard to bypass for actual cheaters.

This approach appears reasonable as also it focuses on workers that are distracted to the point of dysfunctionality. In order to not be flagged as a cheater, a worker has to produce at least mediocre judgements and not fail on any of the awareness questions. Given the relative simplicity of our experiments we do not expect incapability to be a major hindrance for well-meaning workers in our setting. Truly malicious workers, finally, can be seen as a rather theoretical class given the large scale of popular crowdsourcing platforms on the web. This worker type is expected to be more predominant in small-scale environments where their activity has higher detrimental impact. We believe that our definition is applicable in a wide range of crowdsourcing scenarios due to its generality and

Table 2 Task-dependent share of cheaters before and after using gold standard data

Task	Before gold (%)	After gold (%)
Suitability	2.2	1.6
Relevance	37.3	28.4

flexibility. The concrete threshold value of agreement to be reached as well as an appropriate type of awareness question should be selected depending on the task at hand.

4.2 Task-dependent evaluation

As a first step into understanding the dynamics of cheating on crowdsourcing platforms, we compare the baseline cheater rate for the two previously introduced HIT types. The main differences between the two tasks are task novelty and complexity. Plain relevance judgements are frequently encountered on crowdsourcing platforms and can be assumed to be well-known to a great number of workers. Our suitability survey, on the other hand, is a novel task that requires significantly more consideration. Directly comparing absolute result quality across tasks would not be meaningful due to the very different task-inherent difficulty. Table 2 shows the observed share of cheaters for both tasks before and after using gold standard data.

We can find a substantially higher cheater rate for the straightforward relevance assessment. The use of gold standard data reduced cheater rates for both tasks by a comparable degree (27.3% relative reduction for the suitability HIT and 24% for the relevance assessments). With respect to our first research question, we note that more complex tasks that require creativity and abstract thinking attract a significantly smaller percentage of cheaters. We assume this observation to be explained by the interplay of two processes: (1) Money-driven workers prefer simple tasks that can be easily automated over creative ones. (2) Entertainment-seekers can be assumed to be more attracted towards novel, enjoyable and challenging tasks. For all further experiments in this work we will exclusively inspect the relevance assessment task as it has a higher overall cheater rate that is assumed to more clearly illustrate the impact of the various evaluated factors.

4.3 Interface-dependent evaluation

We have shown how innovative tasks draw a higher share of faithful workers that are assumed to be primarily interested in diversion. However, in the daily crowdsourcing routine, many tasks are of rather straightforward nature. In this section, we will evaluate how interface design can influence the observed cheater rate even for well-known tasks such as image tagging or relevance assessments. Traditional interface design commonly tries to not distract the user from the task at hand (Shneiderman 1997). As a consequence, the number of context changes is kept as low as possible to allow focused and efficient working. While this approach is widely accepted in environments with trusted users, crowdsourcing may require a different treatment. A smooth interaction with a low number of context changes makes a HIT prone to automation, either directly by a money-driven worker or by scripts and bots. We investigate this connection at the example of our relevance assessment task.

Table 3 shows the results of this comparison. In the first step, we present the workers with batches of 10 web page/query pairs using gold standard data. In order to keep the

Table 3 Interface-dependent percentage of cheaters for variable queries, variable documents and fully variable pairs

Interface type	Observed cheater rate (%)
Variable queries	28.4
Variable documents	21.9
Both variable	18.5

number of context changes to a minimum we asked the workers to visit a single web page[6] and create relevance judgements for that page given 10 different queries. The resulting share of cheaters turns out to be substantial (28.4%). Now we increase the amount of interaction in the HIT by requiring the worker to create 10 judgements for query/document pairs in which we keep the query constant and require visiting 10 unique web pages. Under this new setting the worker is required to make significantly more context changes between different web pages. While in a controlled environment with trusted annotators this step would be counterproductive, we see a significant relative decline of 23% to a proportion of 21.9% cheaters. In a final step, we issue batches of 10 randomly drawn query/document pairs. As a result, the proportion of cheaters decreases by another 15 to 18.5%. The general HIT interface remains unchanged from the original example shown in Fig. 3, only the combinations of query/document pairs vary. With respect to our second research question, we find that greater variability and more context changes discourage deceivers as the task appears less susceptible to automation or cheating, and therefore less profitable.

4.4 Crowd filtering

In this section, we will address our third research question by inspecting a number of commonly used filtering strategies to narrow down the pool of eligible workers that can take up a HIT. In order to make for a fair comparison, we will regard two settings as the basis of our juxtaposition: (1) The initial cheat-prone relevance assessment setup with 10 queries and 1 document, using gold standard verification questions. (2) The previous best performance that was achieved using gold standard verification sets as well as randomly drawn query/document pairs as described in Sect. 4.3 Please note that the crowd filtering experiment were exclusively run on AMT as not all previously used platforms offer the same filtering functionalities. The HITs created in this section are not shown in Table 1, as they would artificially boost AMT's prominence even further.

4.4.1 By prior performance

In the course of this document we argued that the widely used prior acceptance rates are not an optimal means of assessing worker reliability. In order to evaluate the viability of our hypothesis we increase the required threshold accuracy to 99% (The default setting is

[6] The pages used in this study originate from the ClueWeb'09 collection (http://lemurproject.org/clueweb09.php/) and the queries and gold standard judgements for topics 51-57 from NIST's TREC 2010 Web track adhoc task (Clarke 2009).

95%). Given a robust reliability measure, this, at least seemingly very strict standard should result in highest result quality.

4.4.2 By origin

Offering the HIT exclusively to inhabitants of certain countries or regions is a further commonly-encountered strategy for fending off cheaters and sloppy workers. Following our model of money-driven and entertainment-driven workers we assume that offering our HITs to developed countries should result in lower cheater rates. In order to evaluate this assumption we repeat the identical HIT that was previously offered unrestrictedly, on a pure US crowd.

4.4.3 By recruitment

Recruitment (sometimes also called qualification) HITs are a further means of a priori narrowing down the group of workers. In a multi-step process, workers are presented with preparatory HITs. Workers that achieve a given level of result quality are eligible to take up the actual HIT. In our case we presented workers with the identical type of HIT that was evaluated later and accepted every worker that did not qualify as a cheater (according to Definition 1) for the final experiment.

4.4.4 Results

The first two columns of Table 4 shows an overview of the three evaluated filtering dimensions. Raising the threshold of prior acceptance from the 95% default to 99% only gradually lowered the observed cheater rate. Filtering depending on worker origin was able to cut cheater rates down to less than a third of the originally observed 28.4%. However, this substantial reduction comes at a cost. The run time of the whole batch increased from 5 hours to slightly under one week as we limit the crowd size. Providers of time-sensitive or very large HIT batches may have to consider this trade-off carefully. The introduction of a recruitment step prior to the actual HIT was able to reduce the cheater rate, however, the cheat reduction vs. increase in completion time is worse than for origin-based filtering. To further confirm and understand these trends, columns 3 and 4 of the same table display the same statistics for the varied HIT setting in which we assigned random query/document pairs. In general, the effect of filtering turned out to be largely independent of the previously applied interface changes. The relative merit of the applied methods was found to be comparable for both the initial and the high-variance interface.

Table 4 Effect of crowd filtering on cheater rate and batch processing time

Filtering method	Cheaters (initial) (%)	Time (initial) (h)	Cheaters (varied) (%)	Time (varied) (h)
Baseline	28.4	3.2	18.5	5.2
99% prior acc.	26.2	3.8	17.7	7.6
US only	8.4	140	5.4	160
Recruitment	19	145	12.2	144

The conclusion towards our third research question is twofold: (1) We have seen how prior crowd filtering can greatly reduce the proportion of cheaters. 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 robust stand-alone predictor of his reliability.

4.5 The influence of batch sizes

Crowdsourced HITs are typically issued on large scale in order to collect significant amounts of data. Currently, HIT batch sizes are typically adjusted according to practical or organizational needs but with little heed to result quality. Wang et al. (2011) gave a first intuition of an instrumental use of batch sizes by showing that small batches typically have longer per-HIT completion times than large ones. We assume that this tendency is explained by large HIT batches being more attractive for workers interested time-efficiency. A batch of only 2 HITs has a relatively large overhead of reading and understanding the task instructions before completing actual work. For large batches, workers have a significantly higher reuse potential. The same holds true for cheating. Investing time into finding a way to game a 5-HIT batch is far less attractive than doing the same for a batch of 100 HITs. As a consequence, we expect large HIT batches to attract relatively higher cheater rates than small batches. Previously, all HITs were offered in batches of 50. In order to evaluate our hypothesis we issued several batches of relevance assessment HITs (see Fig. 3 in the Appendix) and compared the observed cheater rates depending on the batch size. For each setting, we collected judgements for 100 query/document pairs. Except for the batch size, all experiment parameters were kept at the settings described in Sect. 4. Batches were requested one at a time. Only after a batch's completion would we publish the following one. In this way we aim to avoid giving the impression that there was a large amount of similar HITs available to be preyed on by cheaters. As a consequence, we do not expect major external effects caused

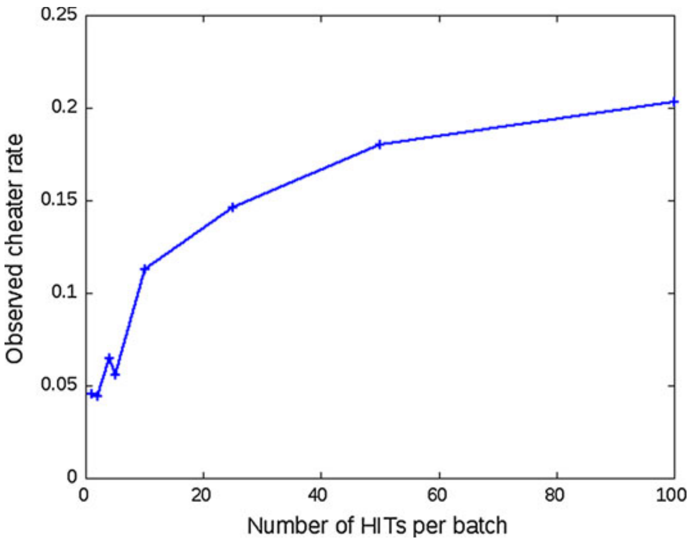


Fig. 1 Observed percentage of cheaters for HIT batches of variable size

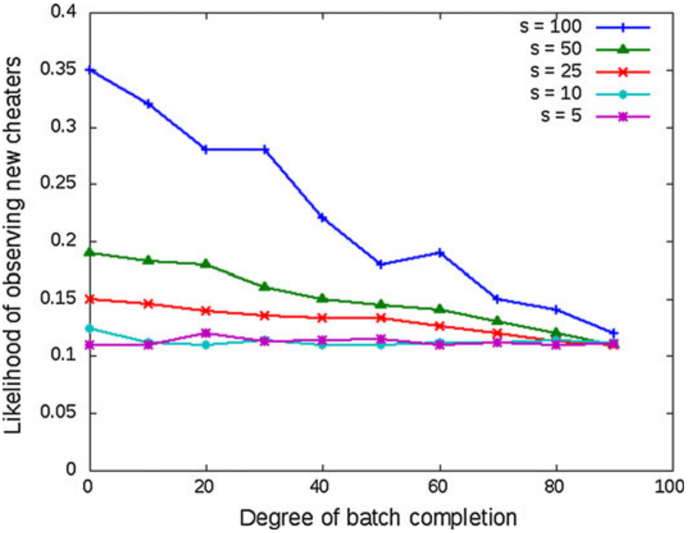


Fig. 2 Likelihood of observing new cheaters as batches approach completion

by the resulting higher number of batches offered as they are never available at the same time. Figure 1 shows the result of this comparison. The figure represents the mean observed cheater rate for each batch size s across a population of $n = \frac{100}{s}$ batches. We can note a statistically significant (using Wilcoxon signed Rank test with $\alpha < 0.05$) increase in cheating activity for batch sizes of at least 10 HITs. As a consequence of determining cheater status at task level, we do not expect any influence of the batch size on the confidence of our cheater detection since the number of HITs per task remained unchanged across batch size settings.

As a further piece of evidence, let us investigate how the cheater rate develops within a batch as HIT submissions arrive. The previously made observations would imply that, as the batch approaches completion, the arrival of new cheaters should become less frequent as the batch of available HITs shrinks in size. To pursue this intuition, workers were ordered according to their time of first submission to the batch. Subsequently, we determined the maximum likelihood estimate of encountering a new cheater, $p(c)$ given an original batch size and a degree of completion as:

$$p(c|s, \omega) = \frac{|C_{s,\omega}|}{|W_{s,\omega}|}$$

where $|C_{s,\omega}|$ is the number of new cheaters observed for size s at degree of completion ω, and $|W_{s,\omega}|$ is the overall number of new workers arriving at that time. Figure 2 shows the resulting distributions. For significantly large s, we can clearly see our intuition confirmed. As the number of remaining HITs declines, new cheaters are observed less and less frequently. For settings of $s < 25$ the distributions are near uniform and we could not determine significant changes over time.

With respect to our fourth research question, we conclude that larger batches indeed attract more cheaters as they offer greater potential of automation or repetition. This finding holds interesting implications for HIT designers, who may consider splitting up large batches into multiple smaller ones.

5 Discussion and conclusion

In this work we investigated various ways of making crowdsourcing HITs more robust against cheat submissions. Many state of the art approaches to deal with cheating rely on posterior result filtering. We choose a different focus by trying to design and formulate HITs in such a way that they are less attractive for cheaters. The factors evaluated in this article are: (1) The HIT type. (2) The HIT interface. (3) The composition of the worker crowd. (4) The size of HIT batches.

Based on a range of experiments, we conclude that cheaters are less frequently encountered in novel tasks that involve creativity and abstract thinking. Even for straightforward tasks we could achieve significant reductions in cheater rates by phrasing the HIT in a non-repetitive way that discourages automation. Crowd filtering could be shown to have significant impact on the observed cheater rates, while filtering by origin or by means of a recruitment step were able to greatly reduce the amount of cheating, the batch processing times multiplied. We are convinced that implicit crowd filtering through task design is a superior means to cheat control than excluding more than 80% of the available workers from accessing the HIT. An investigation of batch sizes further supported the hypothesis of fundamentally different worker motivations, as the observed cheater rates for large batches that offer a high reuse potential, were significantly higher than those for small batches.

Finally, our experiments confirmed that prior acceptance rates, although widely used, cannot be seen as a robust measure of worker reliability. Recently, we have seen a change in paradigms in this respect. In June 2011, Amazon introduced a novel service on AMT that allows to issue HITs to an selected crowd of trusted workers, so-called Masters (at higher fees). Master status is granted per task type and has to be maintained over time. For example, as a worker reliably completes a high number of image tagging HITs, he will be granted Master status for this particular task type. Currently, the available Master categories are “Photo Moderation” and “Categorization”. Due to the recency of this development, we were not able to set up a dedicated study of the performance-cost trade-off of Master crowds versus regular ones.

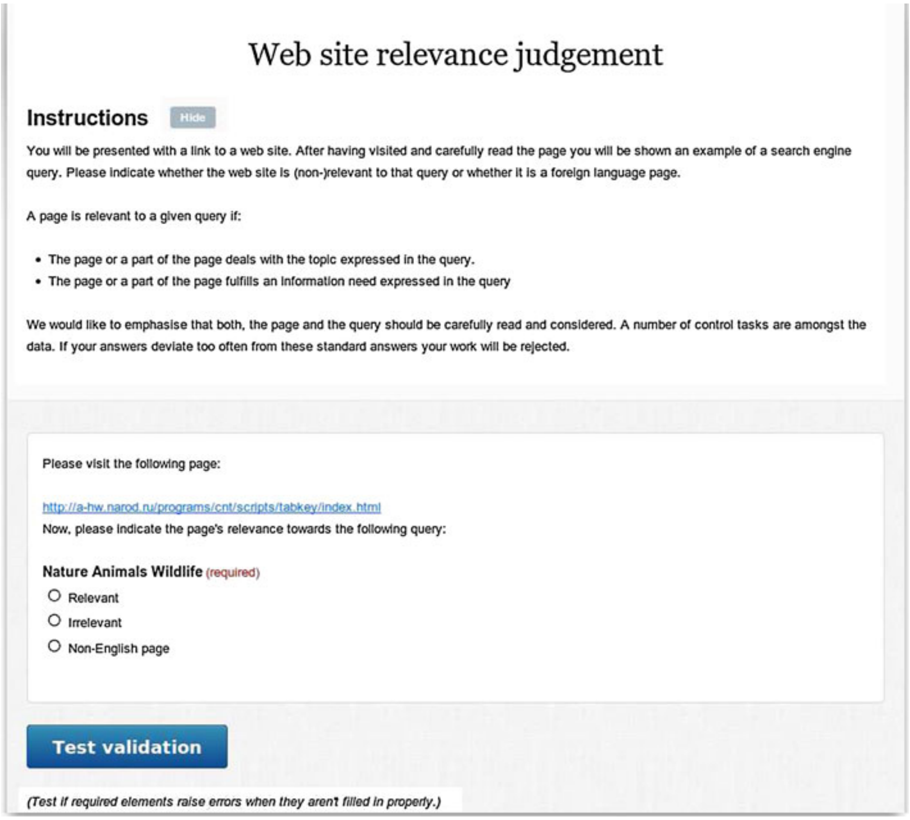
Novel features like this raise an important general question to be addressed by future work: Temporal instability is a major source of uncertainty in current crowdsourcing research results. The crowdsourcing market is developing and changing at a high pace and is connected to the economical situation outside the cloud. Therefore, it is not obvious whether this year’s findings about worker behaviour, the general composition of the crowd or HIT design would still hold two years from now. Besides empirical studies, we see a clear need for explicit models of the crowd. If we could build a formal representation of the global (or, depending on the application, local) crowd, including incentives and external influences, we would have a reliable predictor of result quality, process costs and required time at our fingertips, where currently the process is trial-and-error-based.

One particularly interesting aspect of such a model of crowdsourcing lies in a better understanding of worker motivation, gained for example through activity log analyses and usage history, can solicit more sophisticated worker reliability models. To this end, we are currently investigating the use of games in commercial crowdsourcing tasks. There are different fundamental motivations for offering one’s labour on a crowdsourcing platform. Following previous experience, we hypothesise that workers who are mainly entertainment-driven and for whom the financial reward only plays a subordinate role, are less likely to cheat during the task. Using games, such workers can be rewarded more appropriately by representing HITs in engaging and entertaining ways. Ultimately, this should lead to better results and greater cost-efficiency of crowdsourcing.

Acknowledgments This research is part of the PuppyIR project (PuppyIR (2011). It is funded by the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement no. 231507.

Appendix

See Figs. 3 and 4.



Web site relevance judgement

Instructions

You will be presented with a link to a web site. After having visited and carefully read the page you will be shown an example of a search engine query. Please indicate whether the web site is (non-)relevant to that query or whether it is a foreign language page.

A page is relevant to a given query if:

- The page or a part of the page deals with the topic expressed in the query.
- The page or a part of the page fulfills an information need expressed in the query

We would like to emphasise that both, the page and the query should be carefully read and considered. A number of control tasks are amongst the data. If your answers deviate too often from these standard answers your work will be rejected.

Please visit the following page:

http://a-hw.narod.ru/programs/cnt/scripts/tabkey/index.html

Now, please indicate the page's relevance towards the following query:

Nature Animals Wildlife (required)

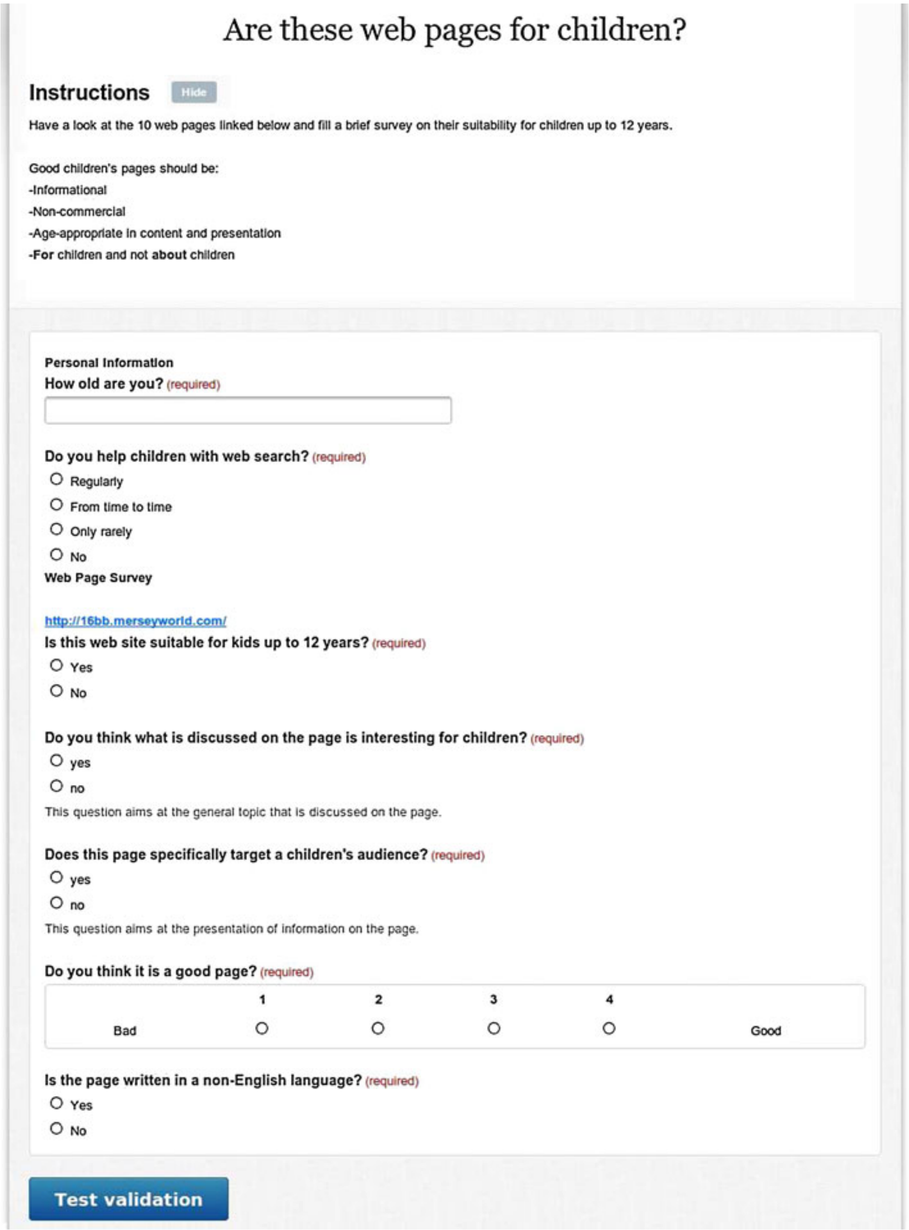
○ Relevant

○ Irrelevant

○ Non-English page

(Test if required elements raise errors when they aren't filled in properly.)

Fig. 3 Relevance judgement HIT



Are these web pages for children?

Instructions

Have a look at the 10 web pages linked below and fill a brief survey on their suitability for children up to 12 years.

Good children's pages should be:
-Informational
-Non-commercial
-Age-appropriate in content and presentation
-For children and not about children

Personal Information

How old are you? (required)

Do you help children with web search? (required)

○ Regularly
○ From time to time
○ Only rarely
○ No

Web Page Survey

http://16bb.merseyworld.com/

Is this web site suitable for kids up to 12 years? (required)

○ Yes
○ No

Do you think what is discussed on the page is interesting for children? (required)

○ yes
○ no

This question aims at the general topic that is discussed on the page.

Does this page specifically target a children's audience? (required)

○ yes
○ no

This question aims at the presentation of information on the page.

Do you think it is a good page? (required)

	1	2	3	4	
Bad	○	○	○	○	Good

Is the page written in a non-English language? (required)

○ Yes
○ No

Fig. 4 Web page suitability HIT

References

Alonso, O., & Mizzaro, S. (2009). 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* (pp. 15–16). Citeseer.

Ambati, V., Vogel, S., & Carbonell, J. (2010). Active learning and crowd-sourcing for machine translation. *Language Resources and Evaluation (LREC) , 7*, 2169–2174.

Amitay, E., Carmel, D., Lempel, R., & Soffer, A. (2004). Scaling IR-system evaluation using term relevance sets. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 10–17), ACM.

Baio, A. (2008). The faces of mechanical turk. http://waxy.org/2008/11/the_faces_of_mechanical_turk.

Clarke, C. (2009). *Overview of the trec 2009 web track*. Technical report, Waterloo University.

Dawid, A., & Skene, A. (1979). Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society Series C (Applied Statistics), 28*(1), 20–28.

Eickhoff, C., Serdyukov, P., & de Vries, A. (2011). A combined topical/non-topical approach to identifying web sites for children. In *Proceedings of the 4th ACM international conference on Web search and data mining* (pp. 505–514). ACM.

Grady, C., & Lease, M. (2010). Crowdsourcing document relevance assessment with mechanical turk. In *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's mechanical turk* (pp. 172–179). Association for Computational Linguistics.

Harman, D. (1993). Overview of the first trec conference. In *Proceedings of the 16th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 36–47). ACM.

Harris, C. (2011). Youre hired! An examination of crowdsourcing incentive models in human resource tasks. *Crowdsourcing for search and data mining* (*CSDM 2011*) (p. 15).

Hirth, M., Hoßfeld, T., & Tran-Gia, P. (2010). *Cheat-detection mechanisms for crowdsourcing*. Technical report, University of Würzburg.

Hsueh, P., Melville, P., & Sindhwani, V. (2009). Data quality from crowdsourcing: A study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing* (pp. 27–35). Association for Computational Linguistics.

Ipeirotis, P. (2010a). 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.

Ipeirotis, P. (2010b). *Demographics of mechanical turk*. Center for Digital Economy Research, NYU Stern School of Business, Working paper.

Kaufmann, N., Schulze, T., & Veit, D. (2011). More than fun and money. Worker motivation in crowdsourcing—A study on mechanical turk. In *Proceedings of the 17th Americas Conference on Information Systems*.

Kazai, G. (2011). In search of quality in crowdsourcing for search engine evaluation. *Advances in Information Retrieval, 6611*, 165–176.

Kazai, G., Kamps, J., & Milic-Frayling, N. (2011). Worker types and personality traits in crowdsourcing relevance labels. In *Proceedings of 20th International Conference on Information and Knowledge Management (CIKM)*. ACM.

Khanna, S., Ratan, A., Davis, J., & Thies, W. (2010). Evaluating and improving the usability of mechanical turk for low-income workers in india. In *Proceedings of the first ACM symposium on computing for development* (p. 12). ACM.

Kittur, A., Chi, E., & Suh, B. (2008). Crowdsourcing user studies with mechanical turk. In *Proceeding of the 26th annual SIGCHI conference on human factors in computing systems* (pp. 453–456). ACM.

Lesher, G., & Sanelli, C. (2000). A web-based system for autonomous text corpus generation. In *Proceedings of ISSAAC*.

Little, G., Chilton, L., Goldman, M., & Miller, R. (2009). Turkit: Tools for iterative tasks on mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation* (pp. 29–30). ACM.

Marcus, M., Marcinkiewicz, M., & Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics 19*(2), 313–330.

Moore, A., Ellison, R., & Linger, R. (2001). *Attack modeling for information security and survivability*. Technical report, Carnegie-Mellon University, Software Engineering Institute.

MTurk. (2011). *Amazon mechanical turk—Artificial artificial intelligence*. https://www.mturk.com/.

Pfleeger, C., & Pfleeger, S. (2007). *Security in computing* (Vol. 604). New Yok: Prentice Hall.

PuppyIR. (2011). PuppyIR: An open source environment to construct information services for children. http://www.puppyir.eu.

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence* (pp. 1044–1049).

Ross, J., Irani, L., Silberman, M., Zaldivar, A., & Tomlinson, B. (2010). Who are the crowdworkers? Shifting demographics in mechanical turk. In *Proceedings of the 28th of the international conference extended abstracts on human factors in computing systems* (pp. 2863–2872). ACM.

Shneiderman, B. (1997). *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Snow, R., O'Connor, B., Jurafsky, D., & Ng, A. (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.

Soboroff, I., Nicholas, C., & Cahan, P. (2001). Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 66–73). ACM.

Soleymani, M., & Larson, M. (2010). *Crowdsourcing for affective annotation of video: Development of a viewer-reported boredom corpus*. In *Proceedings of the ACM SIGIR 2010 workshop on crowdsourcing for search evaluation* (*CSE 2010*) (pp. 4–8).

Sorokin, A., & Forsyth, D. (2008). Utility data annotation with amazon mechanical turk. In *Computer vision and pattern recognition workshops, 2008. CVPRW'08. IEEE computer society conference on IEEE* (pp. 1–8).

Wang, J., Faridani, S., Ipeirotis, P. (2011). Estimating the completion time of crowdsourced tasks using survival analysis models. *Crowdsourcing for search and data mining* (*CSDM 2011*) (p. 31).