wsdm
2011 Hong Kong
Fourth ACM International Conference
on Web Search and Data Mining
9-12 February 2011
Hong Kong

Crowdsourcing for Search and Data Mining (CSDM 2011)

Hong Kong, China, Feb. 9, 2011
http://ir.ischool.utexas.edu/csdm2011

Workshop of the Fourth ACM International Conference on Web Search and Data Mining (WSDM 2011)

Organized by

Vitor Carvalho, Intelius
Matthew Lease, University of Texas at Austin
Emine Yilmaz, Microsoft Research

Program Committee

Omar Alonso, Microsoft Bing
Adam Bradley, Amazon
Ben Carterette, University of Delaware
Charlie Clarke,University of Waterloo
Deepak Ganesan, University of Massachusetts Amherst
Panagiotis G. Ipeirotis, New York University
Gareth Jones, Dublin City University
Jaap Kamps, University of Amsterdam
Gabriella Kazai, Microsoft Research
Mounia Lalmas, University of Glasgow
Martha Larson, Delft University of Technology
Winter Mason, Yahoo! Research
Don Metzler, University of Southern California
Stefano Mizzaro, University of Udine
Gheorghe Muresan, Microsoft Bing
Iadh Ounis, University of Glasgow
Mark Sanderson, RMIT University
Mark Smucker, University of Waterloo
Siddharth Suri, Yahoo! Research
Fang Xu, Saarland University

wsdm
2011 Hong Kong
Fourth ACM International Conference
on Web Search and Data Mining
9-12 February 2011
Hong Kong

Crowdsourcing for Search and Data Mining (CSDM 2011)

Workshop Proceedings

(alphabetized by first author's last name)

Invited Talks

The Smarter Crowd: Active Learning, Knowledge Corroboration, and Collective IQs.
Thore Graepel. 3

Crowdsourcing using Mechanical Turk: Quality Management and Scalability.
Panagiotis G. Ipeirotis. 5

Individual vs. Group Success in Social Networks.
Winter Mason. 6

Accepted Papers

Perspectives on Infrastructure for Crowdsourcing.
Omar Alonso 7

How Crowdsourcable is Your Task?
Carsten Eickhoff and Arjen de Vries. ………………………..............……………………….. 11

You're Hired! An Examination of Crowdsourcing Incentive Models in Human Resource Tasks.
Christopher Harris. ……………………………………………………………………………. 15

Modeling Annotator Accuracies for Supervised Learning.
Abhimanu Kumar and Matthew Lease. ………………………………………………………... 19

Crowdsourcing Blog Track Top News Judgments at TREC.
Richard McCreadie, Craig Macdonald and Iadh Ounis. ……………………………………….. 23

Investigating Factors Influencing Crowdsourcing Tasks with High Imaginative Load.
Raynor Vliegendhart, Martha Larson, Christoph Kofler, Carsten Eickhoff and Johan Pouwelse .. 27

Estimating Completion Time for Crowdsourced Tasks Using Survival Analysis Models.
Jing Wang, Siamak Faridani and Panagiotis Ipeirotis. …………………………………………… 31

Crowdsourcing Interactions - A proposal for capturing user interactions through crowdsourcing.
Guido Zuccon, Teerapong Leelanupab, Stewart Whiting, Joemon Jose and Leif Azzopardi. ……. 35

The Smarter Crowd: Active Learning, Knowledge Corroboration, and Collective IQs

Thore Graepel
Microsoft Research
Cambridge, UK
thoreg@microsoft.com

Abstract

Crowdsourcing mechanisms such as Amazon Mechanical Turk (AMT) or the ESP game are now routinely being used for labelling data for machine learning and other computational intelligence applications. I will discuss three important aspects of crowdsourcing which can help us tap into this powerful new resource in a more efficient way.

When obtaining training data from a crowdsourcing system for the purpose of machine learning we can either collect all the training data in one batch or proceed sequentially and decide which labels to obtain based on the model learnt from the data labelled so far, a method often referred to as active learning. I will discuss which criteria can be used for selecting new examples to be labelled and demonstrate how this approach has been used in the FUSE/MSRC news recommender system projectemporia.com to categorise news stories in a cost-efficient way.

Data obtained from crowdsourcing systems is typically plentiful and cheap, but noisy. The redundancy in the data can be used to improve the quality of the inferred labels based on models that take into account the reliability and expertise of the workers as well as the nature and difficulty of the tasks. I will present an algorithm for such a corroboration process based on graphical models, and show its application on the example of verifying the truth values of facts in the entity-relationship knowledge base Yago.

Finally, I will talk about some very recent results on the effects of parameters of crowdsourcing marketplaces (such as price and required track record for participation) on the quality of results. This work is based on methods from psychometrics, effectively measuring the IQ of the Mechanical Turk when viewed as a form of collective intelligence.

This is joint work with Ralf Herbrich, Ulrich Paquet, David Stern, Jurgen Van Gael, Gjergji Kasneci, and Michal Kosinksi.

Biography

Thore Graepel is a senior researcher at Microsoft Research Cambridge (MSRC), UK, and heads the Online Services and Advertising (OSA) research group. The OSA group conducts research in the area of applied machine learning with applications to online advertising,

WSDM 2011 Workshop on Crowdsourcing for Search and Data Mining (CSDM'11). Hong Kong, China, Feb. 9, 2011.

online gaming, and web search. A common theme of Thores research is the idea of using the tremendous wealth of data Microsoft is generating through its services to improve those services based on data-driven computational intelligence. Thores research has impacted Microsoft products and services, e.g. through the TrueSkill ranking and matchmaking system in Xbox Live and Halo 3, as well as through his work with adCenter on the prediction of user behaviour. Before starting the OSA group together with Ralf Herbrich, Thore co-founded the Applied Games group at MSRC whose research is aimed at bringing adaptable artificial intelligence to computer games. Thore has a strong academic track record with over 50 peer-reviewed publications in machine learning and probabilistic modelling. His basic research focuses on inference in large scale probabilistic models and knowledge bases. He is also interested in classification, recommender systems, and crowdsourcing for machine learning.

Crowdsourcing using Mechanical Turk: Quality Management and Scalability

Panos Ipeirotis
Stern School of Business, New York University
New York, NY, USA
panos@stern.nyu.edu

Abstract

I will discuss the repeated acquisition of "labels" for data items when the labeling is imperfect. Labels are values provided by humans for specified variables on data items, such as "PG-13" for "Adult Content Rating on this Web Page." With the increasing popularity of micro-outsourcing systems, such as Amazon's Mechanical Turk, it often is possible to obtain less-than-expert labeling at low cost. We examine the improvement (or lack thereof) in data quality via repeated labeling, and focus especially on the improvement of training labels for supervised induction. We present repeated-labeling strategies of increasing complexity, and show several main results: (i) Repeated-labeling can improve label quality and model quality (per unit data-acquisition cost), but not always. (ii) Simple strategies can give considerable advantage, and carefully selecting a chosen set of points for labeling does even better (we present and evaluate several techniques). (iii) Labeler (worker) quality can be estimated on the fly (e.g., to determine compensation, control quality or eliminate Mechanical Turk spammers) and systematic biases can be corrected. I illustrate the results with a real-life application from on-line advertising: using Mechanical Turk to help classify web pages as being objectionable to advertisers. Time permitting, I will also discuss our latest results showing that mice and Mechanical Turk workers are not that different after all.

Biography

Panos Ipeirotis is an Associate Professor at the Department of Information, Operations, and Management Sciences at Leonard N. Stern School of Business of New York University. His recent research interests focus on crowdsourcing and on mining user-generated content on the Internet. He received his Ph.D. degree in Computer Science from Columbia University in 2004, with distinction. He has received two "Best Paper" awards (IEEE ICDE 2005, ACM SIGMOD 2006), two "Best Paper Runner Up" awards (JCDL 2002, ACM KDD 2008), and is also a recipient of a CAREER award from the National Science Foundation. He also blogs about crowdsourcing and Mechanical Turk from time to time, an activity that seems to generate more interest and recognition than any of the above.

WSDM 2011 Workshop on Crowdsourcing for Search and Data Mining (CSDM'11). Hong Kong, China, Feb. 9, 2011.

Individual vs. Group Success in Social Networks

Winter Mason
Yahoo! Research
NY, USA
winteram@yahoo-inc.com

Abstract

There are many times when individuals work independently to find a solution to a problem but share information about their progress along the way, such as companies innovating on the latest product, scientists searching for a cure to a disease, or inventors competing for an "X-prize". In these situations, when individuals are collectively searching for a solution to a problem, the flow of information between them can affect both how quickly the group finds the solution and who benefits the most.

To study this situation, we had participants recruited from Amazon's Mechanical Turk play a game in which they were searching a hidden payoff function resembling a rugged landscape with a single global optimum, and were paid based on the points they earned. They could also see where three other players had explored and how much they earned, and were able to exactly copy these players' choices and rewards. Unknown to the participants, they were connected to each other in one of several pre-defined networks. For this type of problem we find decentralized networks are most beneficial for the group, although the best-performing position in a hierarchy does roughly as well as the average member in a decentralized network. Additionally, we find a social dilemma emerges, where it is individually advantageous to exploit previously found solutions, but better for the group if individuals explore.

Biography

Winter Mason received a B.S. in Psychology from University of PIttsburgh in 1999 and a Ph.D. in Cognitive Science and Social Psychology from Indiana University in 2007. For the past 3 years he has worked as a post-doctoral fellow at Yahoo! Research in the Human Social Dynamics group.

WSDM 2011 Workshop on Crowdsourcing for Search and Data Mining (CSDM'11). Hong Kong, China, Feb. 9, 2011.

Perspectives on Infrastructure for Crowdsourcing

Omar Alonso
Microsoft Corp.
1065 La Avenida, Mountain View, CA
omar.alonso@microsoft.com

ABSTRACT

Human computation has gained a lot of interest lately as a paradigm for solving problems that computers can't do yet. Crowdsourcing marketplaces are showing that is possible to get tasks completed in time with a modest budget in a wide range of applications. However, the infrastructure that is currently available for supporting crowdsourcing tasks is very limited. There is non-trivial extra work that the experimenter has to do to get successful results. Consequently, the developer tends to spend more time dealing with the complexity and operational aspects of the task than on its design.

Given the mix of different subareas of computer science that are potentially involved, defining and designing such platform presents a number of interesting problems. In this paper, I outline the rise of human computation, summarize the limitations of a popular platform, and present a perspective on the topic that can be viewed as a starting point to debate requirements for this new research area.

Keywords

Human computation, crowdsourcing, infrastructure, experiment design

1. INTRODUCTION

Human computation is a new research area that studies the process of channeling the vast Internet population to perform tasks or provide data towards solving problems that no known efficient computer algorithms can yet solve. There are two main variations of human processing: games with a purpose and crowdsourcing marketplaces.

Games with a purpose (e.g., the ESP Game) specifically target online gamers who, in the process of playing an enjoyable game, generate useful data (e.g., image tags). There are quite a bit of game prototypes for a wide range of tasks that follow the same structure: you play a nice game while providing ("labeling") good data. A crowdsourcing marketplace is a human computation application that coordinates workers to perform tasks in exchange for rewards (usually money).

In the last couple of years, there has been a growing research interest on leveraging human computation for a broad range of tasks such as relevance evaluation, machine translation, natural language processing, etc. Besides workshops in academic venues like data mining (KDD) and information retrieval (SIGIR), this area is getting industrial traction as well with the first crowdsourcing conference (CrowdConf).

Current research on this emerging area focuses mainly on two fronts: design of games with a purpose and improving the quality of work in certain domains. Little work has been done regarding infrastructure for supporting this type of research.

Borrowing terminology from cloud computing, we can think of human computation as an *elastic human workforce*. Similar to a cloud computing platform that supports utility CPU computing, it should be possible to support utility HPU (human processing unit) computing.

In this paper, I am interested in tasks that a developer would like to run *continuously* (in a production environment) over extended periods of time in an enterprise environment rather than proof of concepts. Hence, one should expect more functionality from such infrastructure.

2. USING MTURK IN PRODUCTION

Amazon Mechanical Turk (MTurk) is probably the biggest commercial crowdsourcing platform available today, where people perform thousands of tasks on a given day. One reason on the adoption of human computation is due to the popularity of MTurk that makes it easier and affordable to conduct experimentation. Traditionally, the process of recruiting users for conducting studies in computer science is expensive and extremely time consuming, making the experiment setup very costly.

Competition in this area is growing rapidly. There are a number of startups trying to provide similar functionality (CloudCrowd) or implement wrappers around MTurk (CrowdFlower) and similar services. Metaweb is another example of a human computation platform with a pool of known workers [10] in contrast to anonymous MTurk workers.

One can perform an analysis of previous experiments with crowdsourcing and identify a number of commonalities among published research. There are four main beneficial properties of using crowdsourcing: speed, cost, quality, and diversity:

1. Experiments go very *fast*, usually getting results in less than 24hrs.
2. Running experiments is usually *cheap*. You can pay a few cents per task and end up spending $25 for the whole experiment.
3. The output is usually of good *quality*. This doesn't mean that there is no need to deal with spammers but

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

with some quality control mechanisms in place, one can yield good results.

4. *Diversity* of workers is good.

Besides the current adoption, MTurk has several drawbacks and has received numerous criticisms about spam handling and the lack of mechanisms to review both workers and requesters. Ipeirotis has pointed out the many things that need to get fixed in MTurk [9] and there is no intention to describe them again here.

Apart from the well-known limitations, the service works and has an important user base. It also has a key feature that is very important in the case of incentives: *payments*. Transferring money around in the world is a tricky business and requires process and legislation in place. A worker wants to get paid on time and the requester wants a pay a minimum with the option to include other monetary incentives.

MTurk, from a HPU view, works reasonably well like a matching site for requesters and workers. Work gets done and workers get paid with some exceptions (see [13] for a report from the field). Other features like analytics are non-existing so all data analysis has to be performed off-line using other tools.

In summary, the requester/developer ends up building a number of tools around the platform to make sure the experiments are successful. That is fine for a first generation of such platforms but more needs to be done to make crowdsourcing more successful and available to wide adoption.

3. AN INFRASTUCTURE PERSPECTIVE

Let's explore the idea of an alternative platform for crowdsourcing. At first, it would make sense to look at this platform from the following landscape: the human view (the workers), the experimenter view (the task designer), and the engine view (the common software features). For each view, I present a scenario and describe a list of desirable features.

3.1 The human view

The end user view (the worker, the human, the end user) is pretty straightforward to understand: easy to use interfaces that allow any human with *minimum* instructions to perform a well-defined task. This can be accomplished by using a browser or cell phone in combination with a task that looks *doable* with the right incentive. Like in every crowdsourcing task, we are interested in grabbing attention for a few minutes so that a number of tasks are completed successfully.

A naïve lecture would suggest that, after all, one only needs a form-based user interface to collect data. A closer look reveals that the designer must follow well know usability techniques for presenting tasks. The elastic human workforce is the planet so cultural and multilingual characteristic should be part of the design.

According to behavioral economics, it is important for people to see the value in the work they perform. An important incentive is obviously money but it should be possible to use other equivalents like points or reputation as currency that workers can use to see that their work is meaningful.

A major complaint from workers is the uncertainty with payments due to in part to fraudulent requesters [13]. At the same time, a major complain from requesters is that workers perform sloppy work or try to game the system to maximize profit.

Features:

- Routing/recommendation of similar tasks based on past behavior and/or content.
- Requester rating based on payment performance, rejected work, and overall task difficulty. A worker should be able to rate the quality of work and also the quality of the requester.
- Ability to comment on a task
- Work categorization. Similarly to a job search site, all work that is available should be classified.

3.2 The experimenter view

The experimenter has two main goals to fulfill: design the right task that produces the data that he/she is looking for and make it appealing to workers. The task has to grab *attention*. Without interesting content people won't complete the task and human computation would not be usable.

A key factor for the success of any task is *attrition*. The requester depends on HPU to complete tasks. So it must provide good trust mechanisms so there is always human workforce available.

The experimenter has to know how to ask the right questions so that minimize the number of instructions. Workers are not experts so it is not possible to assume the same understanding in terms of terminology and expertise.

Having quality control statistics and feedback from workers are key factors for improving the task. It is widely known that inter-coder reliability is a critical component of content analysis. When it is not established properly, the data and interpretations of the data should not be considered valid.

How to get qualified workers for a particular task and how to detect workers that are not doing a good job is an important part of quality control. It is possible to pre-qualify workers with a test or include honey pots in data sets to improve overall quality.

It is important to differentiate workers that are not *suitable* for a particular task from the *spammers* in the system.

Features:

- Ability to manage workers in different levels of expertise including spammers and potential cases.
- Abstract the task as much as possible from the quality control statistics. The developer should provide thresholds for good output.
- Ability to mix different pools of workers based on different profile and expertise levels.
- Honey-pot management and incremental qualification tests based on expertise and past performance.

3.3 The engine view

Cloud computing seems to be a natural fit for this type of system. One thing that is needed is to extend the idea of utility computing and combine CPU + HPU.

MapReduce with human computation looks like an interesting avenue to explore. Both share the following similarities:

- Large task divided into smaller sub-problems
- Work distributed among nodes (workers)
- Collect all answers and combine them

There are some variations as well, notably:

- Human response time varies
- Some tasks are not suitable for human computation
- Consensus among the results

The engine view should have the ability to integrate human-computation into a language. For example, if a developer wants to incorporate human computation to a program, he/she should be able to execute a given task using a particular data set to say a number of workers and expect an inter-agreement level that meets a particular threshold. Parameswaran and Polyzotis describe a declarative language involving human computation functions [19] and discuss the advantages versus a procedural approach.

There are number of statistics available for computing inter agreement like Cohen's kappa, Fleiss' kappa, and Krippendorff's alpha to name a few. In some cases, they may not be appropriate for a given experiment but it would be nice if the system provides such inter-rater statistics by default and the ability to plug-in your own reliability metric.

Features:

- Performance and high availability
- Spam detection built in the system
- Payments (including international markets)
- Inter-agreement statistics library and ability to plug-in a user-defined one
- Uncertainty management
- High-level language for designing tasks
- Analytics

3.4 Integration point

The point of the views discussed before is to show that is possible to think of a crowdsourcing platform on those three actors.

I believe that the power of human computation relies on designing tasks that need to be completed with CPU and HPU. The challenge is then to identify where HPU should be added in such a way that adds value to the task that needs to get done. An example of such approach is Soylent [2] that integrates crowdsourcing into a traditional software program like MS Word.

Remote procedure call is a well-know technique to transfer control and data over a communication network in the context of CPU. A similar mechanism, human procedure call, can be achieved if the platform provides consensus, reliability, and validity as part of the result aggregation for a given task.

Independently of the current offers on the market, the success of a new platform would rely on the ease of use, integration mechanism, and features that make the creation of work and analysis of results accessible to developers.

Features:

- Language model that allows easy integration with other enterprise systems.
- ETL tools

4. RELATED WORK

Related work in human computation and crowdsourcing touches several fields. I mention some of the current research that is relevant to the ideas presented so far.

The notion of human computation as a distributed system is presented in different ways in the literature. Davis et al. outline a comparison with CPU and shows examples of HPU vs. CPU [6]. Heymann and Garcia-Molina present a human programming environment and its model based on MTurk [8]. Quinn and Bederson introduce a taxonomy of distributed human computation in [12].

There is previous work on using crowdsourcing for information retrieval and natural language processing. Alonso and Mizzaro compared a single topic to TREC and found that workers were as good as the original assessors [1]. Tang and Sanderson used crowdsourcing to evaluate user preferences on spatial diversity [16]. Grady and Lease focused their work in human factors for crowdsourcing assessments [7]. Other kind of experiments in IR can be found in [4].

The NLP community has been using MTurk for different tasks. The research work by Snow et al. [14] shows the quality of workers in the context of four different NLP tasks such as affect recognition, word similarity, textual entailment, and event temporal ordering. Callison-Burch shows how Mechanical Turk can be used for evaluating machine translation [3].

A number of tools have been developed for solving some of the limitations of MTurk with TurKit being one of the most popular ones [11]. TurKontrol, a planner for controlling crowdsourced workflows, is presented in [5].

In terms of generic infrastructure, scientific workflows and databases have gotten a lot of attention in recent years that should be possible to adapt to human computation [15].

5. SUMMARY AND CONCLUSIONS

The panel on crowds, clouds, and algorithms covers similar topics to the ones presented in this paper [19]. The development of systems that include crowdsourcing and cloud computing systems will be a major driver of information technology innovation going forward.

Applications that leverage "big data" are likely to benefit from a mix of CPU and HCU. Techniques from a number of computer science areas can used to design such platforms.

From an infrastructure perspective, the main questions are:

- What are the basic software components for HPU?

- Should the underlying infrastructure be a game engine? People are very familiar with game consoles so there is no training needed. On the other hand, games only task may impose a bound on the kind of experiments one would interested to run.

- Should be part of a social networking infrastructure? After all, social networking can provide elastic workforce at any time if there is the right level of engagement.

- Is crowdsourcing for external consumption or can it be adopted on the enterprise as well? There is previous research that shows that crowdsourcing works on the enterprise [17]. Would be possible to combine workforce inside and outside of the enterprise?

- What is the database model for crowdsourcing/HPU?

Human computation is here to stay. Researchers on different areas of information technology are finding HPU an alternative way to collect data to solve problems efficiently.

There are a number of problems with current commercial platforms: they are very rudimentary, there are no tools for data analysis, no data integration with existing systems, and lack of feedback loop between workers and requesters. This creates a number of adoption problems as developers have to spend upfront considerable effort to make experiments viable. For ad-hoc experimentation this may be fine but then an organization needs to run human computation tasks on a continuum, a better service is needed.

The main research questions for this emerging area are:

- What are the tasks suitable for human computation?

- What is the best way to perform human computation?

- What is the best way to combine CPU with HPU for solving problems?

- What are the desirable integration points for a computation that involves CPU and HPU?

I presented a perspective on infrastructure for human computation and outlined challenges and opportunities having an enterprise setting on mind. The observations are made after running lots of experiments in different domains and building missing features that would make life easier.

The list is not exhaustive nor pretends to be a requirements document. The goal is mainly to open a debate on what kind of features and services a crowdsourcing platform should provide in the future. I believe that the more we crowdsource tasks over time, the more we learn about potential features and useful scenarios.

Finally, this area is attracting a lot of interest from industry and academia so I would expect a considerable amount of work in the coming years dedicated to build the next generation of crowdsourcing/human computation infrastructure.

6. REFERENCES

[1] O. Alonso and S. Mizzaro. “Can we get rid of TREC Assessors? Using Mechanical Turk for Relevance Assessment”. *ACM SIGIR Workshop on the Future of IR Evaluation* (2009)

[2] M. Bernstein et al. “Soylent: A Word Processor with a Crowd Inside”, *UIST* (2010)

[3] C. Callison-Burch. “Fast, Cheap, and Creative: Evaluating Translation Quality Using Amazon’s Mechanical Turk”, *EMNLP* (2009)

[4] V. Carvalho, M. Lease and E. Yilmaz (Eds). *Proc. of the SIGIR Workshop on Crowdsourcing for Search Evaluation* (2010)

[5] P. Dai, Mausam, and D. Weld. “Decision-Theoretic of Crowd-Sourced Workflows”, *AAAI* (2010)

[6] J. Davis et al. “The HPU”, *IEEE ACVHL* (2010)

[7] C. Grady and M. Lease. “Crowdsourcing Document Relevance Assessment with Mechanical Turk”. *NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk* (2010)

[8] P. Heymann and H. Garcia-Molina. “Human Processing”, Technical Report, Stanford Info Lab (2010)

[9] P. Ipeirotis http://behind-the-enemy-lines.blogspot.com/2010/10/plea-to-amazon-fix-mechanical-turk.html

[10] S. Kochhar, S. Mazzocchi, P. Paritosh. “The Anatomy of a Large-Scale Human Computation Engine”, *KDD HCOMP* (2010)

[11] G. Little et al. “TurKit: Tools for Iterative Tasks on Mechanical Turk”, *KDD HCOMP* (2009)

[12] A. Quinn and B. Bederson. “A Taxonomy of Distributed Human Computation”, Technical Report HCIL-2009-23, UMD, Human-Computer Interaction Lab (2009)

[13] M. Silberman et al. “Seller’s problems in human computation markets”, *KDD HCOMP* (2010)

[14] R. Snow et al. “Cheap and Fast But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks”. *EMNLP* (2008)

[15] M. Stonebraker et al. “Requirements for Science Data Bases and SciDB”, *CIDR Perspectives* (2009)

[16] J. Tang and M. Sanderson. “Evaluation and User Preference Study on Spatial Diversity”, *ECIR* (2010)

[17] M. Vukovic, J. Laredo, and S. Rajagopal “Challenges and Experiences in Deploying Enterprise Crowdsourcing Service”, *ICWE* (2010)

[18] Panel on “Crowds, Clouds, and Algorithms: Exploring the Human Side of ‘Big Data’ Applications, *SIGMOD* (2010)

[19] A. Parameswaran and N. Polyzotis. “Answering Queries using Humans, Algorithms and Databases”, *CIDR* (2011)

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.

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%

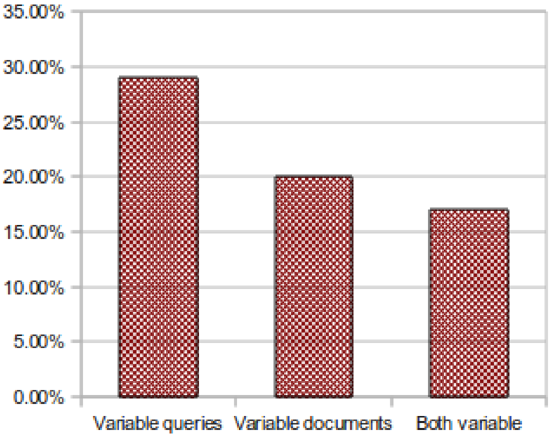


Figure 1: Interface-dependent share of malicious workers for variable queries, variable documents and fully variable pairs.

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

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

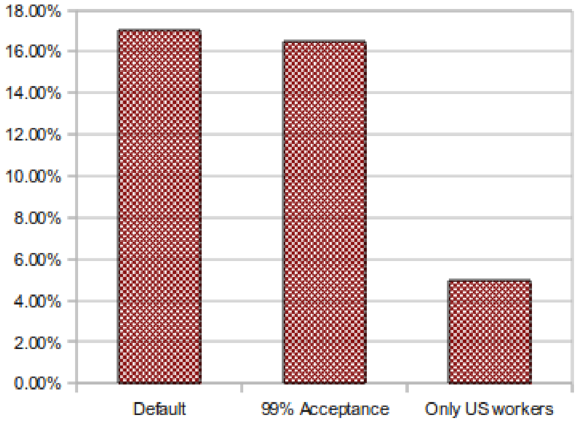


Figure 2: Crowd-dependent share of malicious workers filtered by previous acceptance rate and origin.

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.*

You’re Hired! An Examination of Crowdsourcing Incentive Models in Human Resource Tasks

Christopher G. Harris
Informatics Program
The University of Iowa
Iowa City, IA 52242

christopher-harris@uiowa.edu

ABSTRACT

Many human resource tasks, such as screening a large number of job candidates, are labor-intensive and rely on subjective evaluation, making them excellent candidates for crowdsourcing. We conduct several experiments using the Amazon Mechanical Turk platform to conduct resume reviews. We then apply several incentive-based models and examine their effects. Next, we assess the accuracy measures of our incentive models against a gold standard and ascertain which incentives provide the best results. We find that some incentives actually encourage quality if the task is designed appropriately.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software - *Performance Evaluation*

General Terms

Measurement, Design, Experimentation. Human Factors

Keywords

Relevance judgment, crowdsourcing, incentive models.

1. INTRODUCTION

One challenge companies constantly face is increasing worker productivity without substantially increasing costs. Recent technology has aided many of these productivity gains; however, the most elusive gains are those related to tasks that are repetitive, subjective, and not easy to define algorithmically.

A case in point is a company’s human resources (HR) department, responsible for the new employee recruitment. The typical HR recruiter looks through an average of 200 resumes to fill a single mid-level position; for highly-desirable positions, they can receive ten times this number to review [11]. Technology can help with the search process to discover thousands of online resumes, but is yet unable to make the subjective assessment of which are adequate resumes for a job versus an inadequate one.

Frequently the task of hiring mid-level employees and above is outsourced to executive search firms. These outside recruiters typically charge around one third of the annual base salary of a newly-hired employee. Therefore an inexpensive method of examining resumes can benefit employers or outside search firms cut costs substantially if this activity can be done effectively.

Crowdsourcing tools such as Amazon’s Mechanical Turk[1] (AMT) show considerable promise in having simple yet tedious tasks executed rapidly. These platforms provide a legion of available Internet workers to complete HITs (Human Intelligence Tasks) in exchange for micro-payments – precisely the type of activity that can help HR recruiters narrow a pile of resumes to only those of interest. By dividing a tedious task among a large number of participants, a company can quickly and inexpensively execute tasks in a short timeframe, often within 24 hours.

Our objective is to examine how platforms such as AMT can do well with the types of subjective evaluations computers cannot perform well. Additionally, we wish to examine the role incentives play in aligning the worker’s needs with those of the requester in this anonymous environment.

The remainder of this paper is organized as follows: In the next section, we briefly discuss the background of this emerging area. Next, we describe our experiments and the incentive-based variants of each model. In Section 4, we present our results. Finally, we summarize our findings and indicate our anticipated future work in this area.

2. BACKGROUND

The use of technology in the job search process is certainly not new. Companies such as Monster.com[2] and Jobing.com[3] make use of technology to aid in the indexing, searching and dissemination of resumes. More recently, online recruitment firms have made use of more advanced techniques such as semantic search. Some have even forayed into aspects of crowdsourcing. Previously, job search website TalentSpring[4] had job seekers rank 12 pairs of resumes in a specific professional niche, selecting which candidate is preferable [6]. This technique introduces potential bias – can job seekers be expected to fairly rate their anonymous competitors when a potential job is at risk?

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

[1] `www.mturk.com`

[2] `www.monster.com`

[3] `www.jobing.com`

[4] `www.talentspring.com`

Additionally, even if care was exercised to ensure they are not competitors for a specific position, what would encourage these job seekers to make an accurate assessment of another candidate?

The use of incentive models in behavioral economics has been well-studied, but none of this research has covered incentive models applied to anonymous workers that incorporate worker quality measures. With this in mind, we recognize that this resume selection task is a relevance judgment task and may be ideal for crowdsourcing. Recent research has demonstrated the many benefits of this approach to tasks such as annotating images [10], relevance judgments [2], tracking sentiment [1], and for translation tasks [9]. Likewise, the corporate world has embraced it for soliciting feedback and creative purposes [4], such as designing advertising campaigns, user studies, and or designing a corporate logo [1].

In contrast, there are also several well-discussed drawbacks, such those discussed in [1] [3] and [8] regarding poor or indifferent worker quality and potentially malicious worker intent. Moreover, when unqualified workers perform a judgment task, care must be taken to prevent noisy data as discussed in [5].

This tradeoff raises some important questions: First, can workers with little or no training be used to rate the resumes of job candidates effectively? Also, do some judgment models work better than others? Finally, is there a way to motivate workers through positive or negative incentives? We address these considerations through specifically-designed experiments in the next section.

3. EXPERIMENTAL DESIGN

Our objective is to examine one of the most laborious steps of the hiring process – the resume review – and examine its fit to crowdsourcing. We begin with three actual management-level job descriptions, one for a Human Resources Manager in a financial services company, one for a National Sales Manager in a manufacturing company, and one for a Project Manager in a chemical company. These descriptions were provided by an executive search firm along with 16 applicant-submitted resumes for each of these positions.

After removing all contact information for each of the 48 candidates and anonymizing both the job descriptions and the resumes to alter any potentially-identifiable information, we then replaced all acronyms in the documents with the corresponding terms. We concentrated on management-level positions for three reasons: first, this data was available to us; second, there is more work experience and educational history provided by the candidates for evaluation, and third, this level of candidate represents the largest portion of a recruiter's workload and this is ripe for potential cost-savings through crowdsourcing.

For each of the following eight bundled HITs, we required a brief qualification process to ensure English ability and an AMT approval rating of at least 95%. Each HIT began with 100 participants who passed this initial qualification step. Participants were unable to participate in more than one HIT and had to complete all 48 ratings to have their answers considered for this study.

To ensure participants were not "gaming the ratings", or providing answers without careful consideration simply for compensation, as described in [3] and [8], we included some additional straightforward free-form questions about the job descriptions to ensure attention to detail. Our participants were prompted for basic information after the fifth and tenth rating for each of the three job descriptions, and the participants were not considered if the answers to these six questions indicated a participant had not read the job description carefully – a subjective assessment made by us based on their responses. We used the AMT (Amazon Mechanical Turk) platform for all of our experiments.

3.1 Resume Relevance HIT Design

Participants were asked to evaluate the fit of each resume to the job description on a five-point scale, from a score of 1 (non-relevant) to 5 (highly-relevant). The same anonymized information was provided to a HR Hiring Director with 14 years of experience in management-level executive search, who evaluated these resumes on the same five-point scale. These ratings were used as our gold standard.

3.1.1 Baseline Resume Relevance

Participants in this HIT were provided with 48 resumes to evaluate and were compensated $0.06 per question. No incentives were offered to participants based on their ratings.

3.1.2 Resume Relevance with Positive Incentive

Compensation in this HIT was set as $0.06 per rating; however, each participant was initially told that each resume had already been rated by an expert and if the participant's rating matched the expert's, they would receive a post-task *bonus* payment of $0.06, providing for the possibility of earning $0.12 per rating.

3.1.3 Resume Relevance with Negative Incentive

Compensation in this HIT was set as $0.06 per question; however, each participant was also told a previous expert rating had been made. If the participant's rating differed from the expert's, their compensation would be *reduced* to $0.03 for that rating.

3.1.4 Resume Relevance with Combined Incentives

Compensation for this HIT was set at $0.06 per rating. Participants were told a previous expert rating had been made. They were paid a *bonus* of $0.06 if it matched; however, if their rating differed from the expert's in more than half of the 48 resumes rated, compensation was *reduced* to $0.03 per rating for those which differed; therefore compensation could range from $1.44 (having all ratings differ) to $5.76 (having all ratings match our gold standard).

3.2 Resume Screening HIT Design

In this HIT, we wanted to examine the ability for crowdsourced workers to perform an initial screening of resumes. We included each of the three job descriptions and one resume for each of the 16 candidates for that position; the participant had to mark each resume in one of two ways: either as relevant or non-relevant. For our gold standard, we took the 17 resumes with ratings of 4 or 5 from our HR director as 'relevant'. Participants were unaware of the number of resumes that were determined relevant.

3.2.1 Baseline Resume Screening

Participants completing the HIT successfully were paid $0.06 per rating. No incentives were offered to participants.

3.2.2 Resume Screening with Positive Incentive

Compensation was set as $0.06 per rating. As with the Resume Relevance HIT, participants were notified about the potential of earning a *bonus* payment of $0.03 per rating if their rating matched the one made by our expert.

3.2.3 Resume Screening with Negative Incentive

Compensation in this HIT was set as $0.06 per question. Each participant was also told that if their rating differed from our expert's, compensation would be *reduced* to $0.03 for that rating.

3.2.4 Resume Screening with Combined Incentive

In this HIT, participants were paid $0.06 per rating, and told they could earn a *bonus* of $0.06 for each expert rating they matched; however, if their rating differed from the expert's in more than half of the 48 resumes rated, their compensation was *reduced* to $0.03 per rating for all ratings which differed.

Although in four of the HITs we clearly indicated to participants in advance that we would reduce their compensation if they failed to match the expert ratings, in actuality no participant compensation was reduced.

4. DATA ANALYSIS

Approximately 87% of all task participants passed our qualification exercise (i.e., they supplied coherent answers to our six free-form questions). This passing percentage was fairly consistent across all eight examined HITs. As expected, the average time taken for each Resume Relevance rating was significantly higher than the Resume Screening rating.

4.1 Resume Relevance

The distribution of ratings in all four Resume Relevance HITs was roughly normal as shown in Figure 1. Like our gold standard, the positive incentive model showed a positive bias (skewed right). The negative incentive model was much tighter around the mean (smaller variance). This may indicate that participants with positive incentives may rate job candidates more highly, whereas those with negative incentives take a far more conservative approach. The combined incentive model showed a mix of these effects (positive bias but with a smaller variance).

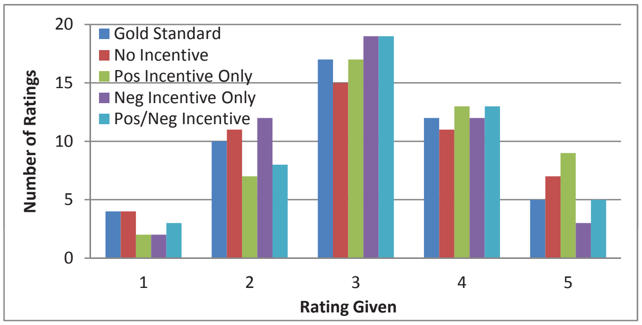


Figure 1. Rating distribution of for the Resume Review HIT

A more important issue was the degree to which the participant's ratings matched our gold standard. As observed in Figure 2, the best matches to our gold standard were the positive model and combined incentive models. Since the granularity of a five-point scale may be too fine, we divide the judgments into two resume judgment groups: scores of 4 or 5 to be 'accepts' and 3 or less to be 'rejections' and compare this with our gold standard. We can then calculate the recall, precision and F-scores for each model (provided in Table 1). Again we find all three incentive models are an improvement over the baseline, with the positive and combined incentive models performing best.

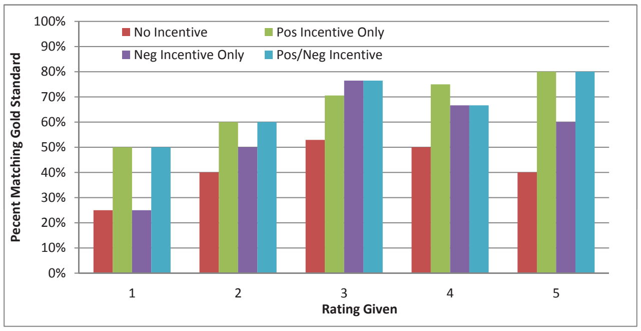


Figure 2. Distribution of Resume Review Ratings Matching the Gold Standard

Since the granularity of a five-point scale may be too fine, we divide the judgments into two resume judgment groups: scores of 4 or 5 to be 'accepts' and 3 or less to be 'rejections' and compare this with our gold standard. We can then calculate the recall, precision and F-scores for each model (provided in Table 1). Again we find all three incentive models improve upon our baseline and the best performers were the positive and combined incentive models.

Table 1. Accuracy Measures for the Resume Review HIT

Incentive Model	Recall	Precision	F-Score
None	0.32	0.47	0.38
Pos	0.54	0.76	0.63
Neg	0.48	0.65	0.55
Pos/Neg	0.55	0.71	0.62

In all HITs, the 48 resumes to be ranked were roughly the same length. By examining the time taken to rank them, we can ascertain a rough metric on each model's encouragement for attention to detail. We were surprised to see the difference in magnitude our incentive models had on each participant's time to complete each rating. Figure 3 illustrates this difference, showing the HIT response time (y-axis) varies as the participant moves through a group of 16 resumes matching a single job description (x-axis).

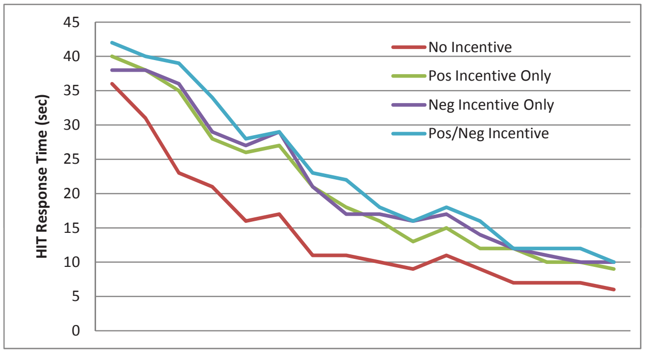


Figure 3. Time taken per rating in the Resume Review HIT.

The three incentive models show a markedly higher response time compared with the baseline model. We believe that the higher rating accuracy for the incentive models and greater response times is likely due to participants with incentives intentionally making more careful decisions.

4.2 Resume Screening

Our Resume Screening HIT was a simple binary judgment and therefore our interest was to investigate which of our models best matched the gold standard. As observed with the Resume Review HIT, the combined and positive incentive models perform best, followed by the negative incentive model. Table 2 illustrates the summary recall, precision and F-score for each model.

Table 2. Accuracy Measures for the Resume Screening HIT

Incentive Model	Recall	Precision	F-Score
None	0.33	0.47	0.39
Pos	0.67	0.82	0.74
Neg	0.54	0.68	0.60
Pos/Neg	0.78	0.82	0.80

All three incentive models performed significantly better than our baseline, non-incentive model, and are similar to those obtained in our Resume Review HIT. We note that the recall measure – arguably more important than precision for our relevance judgment task – is significantly higher for both the positive and the combined incentive models. This further demonstrates the strength of incentives, even when used for simple binary judgments.

The time to complete the Resume Screening HIT showed a similar gap between the three incentive models and the non-incentive model, although the gap was not as pronounced. As with the Resume Review HIT, this likely indicates a higher attention to detail relative to the non-incentive model.

5. CONCLUSION

This preliminary study examined the use of crowdsourcing in resume review and examined the effects of incentives on participant’s accuracy in rating resumes. We observe that these platforms, when the correct incentives are offered, can provide a method of classifying resumes. We also discover that incentives encourage participants to make more accurate judgments.

Although none of the examined incentive models perfectly matched the gold standard in our resume rating assessments, we observe that incentives in general have promise in crowdsourcing activities. Positive and combined incentives are best to encourage more careful consideration of tasks compared with no incentives. These observations applied equally to the five-point ratings in our Resume Review and our binary Resume Screening task.

6. FUTURE WORK

We plan to explore other relevance judgment methods, such as pair-wise preference, respond to incentive models. Additionally, we plan to examine if the size and frequency of the incentive offered has an impact on our results. We also plan to extend the size of our study to incorporate additional raters, examine some of the demographic aspects of our participants, investigate how the clarity of instructions affect participant performance, and examine what is the appropriate task length to achieve the best results. We also plan to examine how crowdsourcing can compare with many machine learning methods. Finally, we plan to examine methods to limit the amount of noisy data in our results.

7. REFERENCES

[1] Alonso, O. 2009. Guidelines for designing crowdsourcing-based relevance evaluation. In *ACM SIGIR*, July 2009.

[2] Alonso, O., Rose, D. E., and Stewart, B. 2008. Crowdsourcing for relevance evaluation. *SIGIR Forum*, 42(2):9-15, 2008.

[3] Donmez P, Carbonell J.G., and Schneider J. 2010. A probabilistic framework to learn from multiple annotators with time-varying accuracy. In: *Proceedings of the SIAM Conference on Data Mining (SDM 2010)*, 826–837

[4] Howe, J. The rise of crowdsourcing. *Wired Magazine 14*, 6 (2006).

[5] Hsueh, P., Melville, P., and 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* (HLT '09). Association for Computational Linguistics, Morristown, NJ, USA, 27-35.

[6] John Cook’s Venture Blog. http://blog.seattlepi.com/venture/archives/115549.asp. Retrieved on Nov 16, 2010.

[7] Kelly, P.G. 2010. Conducting usable privacy & security studies with amazon’s mechanical turk. In *SOUPS ’10: Proceeding on Symposium on User Privacy and Security*. Redmond, WA. July 14-16, 2010.

[8] Kittur, A, Chi, E. H. and Suh, B. 2008. Crowdsourcing user studies with mechanical turk. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, 453-456.

[9] Negri, M. and Mehdad, Y. 2010. Creating a bi-lingual entailment corpus through translations with Mechanical Turk: $100 for a 10-day rush. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk* (CSLDAMT '10). Association of Computational Linguistics, Morristown, NJ, USA, 212-216.

[10] Rashtchian, C., Young, P., Hodosh, M., and Hockenmaier, J. 2010. Collecting image annotations using Amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk* (CSLDAMT '10). ACL, Morristown, NJ, USA. 139-147.

[11] Recruiting Blogs. http://www.recruitingblogs.com/profiles/blogs/talentspring-secures-16. Retrieved on Nov 16, 2010.

[12] Snow, R., O'Connor, B., Jurafsky, D., and Ng, A.Y. 2008. Cheap and fast---but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference for Empirical Methods in Natural Language Processing* Conference (EMNLP '08). Association of Computational Linguistics, Morristown, NJ, USA, 254-263.

[13] Yang, J., Adamic, L.A., and Ackerman, M.S. 2008. Crowdsourcing and knowledge sharing: strategic user behavior on taskcn. In *Proceedings of the 9th ACM conference on Electronic commerce* (EC '08). ACM, New York, NY, USA, 246-255.

Modeling Annotator Accuracies for Supervised Learning

Abhimanu Kumar
Department of Computer Science
University of Texas at Austin
abhimanu@cs.utexas.edu

Matthew Lease
School of Information
University of Texas at Austin
ml@ischool.utexas.edu

ABSTRACT

Crowdsourcing [5] methods are quickly changing the landscape for the quantity, quality, and type of labeled data available to supervised learning. While such data can now be obtained more quickly and cheaply than ever before, the generated labels also tend to be far noisier due to limitations of current quality control mechanisms and processes. Given such noisy labels and a supervised learner, an important question to consider, therefore, is how labeling effort can be optimally utilized in order to maximize learner accuracy? For example, should we (a) label additional unlabeled examples, or (b) generate additional labels for labeled examples in order to reduce potential label noise [12]? In comparison to prior work, we show faster learning can be achieved for case (b) by incorporating knowledge of worker accuracies into consensus labeling [13]. Evaluation on four binary classification tasks with simulated annotators shows the empirical importance of modeling annotator accuracies.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

crowdsourcing, human computation, active learning

1. INTRODUCTION

Historically, supervised learning methods often outperformed their unsupervised counter-parts since providing a learner with more information can enable it to more quickly and effectively learn a desired pattern. Recent years saw this trend reverse, however, due to the massive growth of the Web having provided unsupervised methods with free and seemingly limitless training data [9]. Now the advent of crowdsourcing (e.g. via Amazon's Mechanical Turk[1]), has introduced another potentially disruptive shift: labeled data can suddenly also be obtained far cheaper, easier, and faster than ever before. A significant obstacle remains, though: crowdsourcing methodologies tend to suffer from poor quality control. Consequently, crowdsourced labels are typically quite noisy and exhibit high variance. An important research question, then, is how to most efficiently utilize a crowdsourced-based method for obtaining new labels in order to maximize learning rate with respect to annotation time and cost?

In this paper, we expand on Sheng et al.'s investigation [12] of how labeling effort may be best utilized in order to maximize learner accuracy. Should we (a) label additional unlabeled examples, or (b) generate additional labels for labeled examples in order to reduce potential label noise? In comparison to Sheng et al., the key difference of our work is incorporating knowledge of annotator accuracies into the model, which has large impact on resultant accuracies achieved by the learner. In another line of work by Snow et al. [13], a (slightly less) simple Naive Bayes approach is used to construct a weighted ensemble for consensus labeling in which labels are weighted proportionally to the accuracy of the annotator they come from. Snow et al. assume a fixed number of labels are obtained per example and do not investigate learning rates from consensus labeling. We integrate these two lines of prior work by using knowledge of annotator accuracy to more effectively aggregate labels and thereby improve the learning rate of our supervised model. Results on four binary classification tasks using C4.5 [10] show the empirical effectiveness of our approach, as well as suggesting potential benefit for other tasks and learning models.

2. RELATED WORK

Recent years have seen significant growth in label aggregation research. For example, Raykar et al. model label expertise via the EM algorithm to predict underlying labels [11], building on earlier work by Dawid and Skine [3]. Ipeirotis et al. differentiate error and bias in labeling mistakes with the idea that the bias can still be helpful for learning [6]. Dekel and Shamir give a unique approach to solve noisy label problem by pruning out experts who produce the most noise [4]. Whitehill et al. follow a different approach in that the labeler accuracies are not known a priori to them [15]. Yan et al. provide a predictive algorithm that reduces number of overlapping labels required for label prediction by determining which labels obtained need verification [16].

Alonso et al. use crowd workers to assess relevance [1]. Yang et al. predict the number of overlapping expert labels needed when there is substantial disagreement among the experts [17]. Both Alonso et al. and Yang et al. perform aggregation by simple majority vote. Mason et al. investigate the effect of

[1] https://www.mturk.com

WSDM 2011 Workshop on Crowdsourcing for Search and Data Mining (CSDM'11). Hong Kong, China, Feb. 9, 2011.

compensation on worker accuracies [8]. Little et al. provide a comparative evaluation of collaborative and independent labeling approaches for labeling accuracy and cost [7].

3. TASK

Our task formulation largely mirrors that of Sheng et al. [12]: training a supervised learner for binary classification. We are given a large pool of unlabeled examples from which to draw examples for labeling. Our goal is to maximize classifier accuracy relative to labeling effort (unlike Sheng et al., we assume unlabeled examples are freely obtained). We assume each label requires a fixed cost to produce, regardless of the specific example or the annotator involved (we ignore issues of varying example difficulty or annotator expertise with regard to labeling cost). In addition to the pool of unlabeled examples, we assume a small set of seed examples already assigned a single label. Seed data provides a minimal training set for the classifier to which additional labeled examples may be added. All labels are potentially noisy.

We expect classifier accuracy to improve with more accurate and/or plentiful training data, suggesting a tradeoff for using labeling effort. At each labeling opportunity, should we (a) label an additional, previously unlabeled example or (b) generate a new label for a previously labeled example ("multi-labeling"). While individual labels may be noisy, effective aggregation of multiple labels can potentially yield more accurate consensus labels for training. As in Sheng et al., examples to be labeled are chosen as follows: for (a), uniformly at random from the pool of unlabeled examples, and for (b), using a fixed round-robin schedule which visits each (previously labeled) example once before repeating. We do not consider selection of examples to maximally benefit the learner, to maximally reduce uncertainty of existing labels, or based on example difficulty or the annotator expertise. We also assume the system's choice of (a) or (b) is fixed *a priori*; a more difficult task would require the system to repeatedly choose between (a) and (b) at run-time.

4. METHODS

We compare performance of several methods which differ in two dimensions: how labeling effort is utilized ((a) or (b) above), and for (b), how label aggregation is achieved.

Single Labeling (SL) [12]. Always label a previously unlabeled example; examples are never multi-labeled.

Multi-Labeling with Majority Voting (MV) [12]. Always generate an additional label for a previously labeled example. Labels are aggregated via simple majority vote.

Multi-Labeling with Naive Bayes (NB) [13]. As with MV, always re-label a previously labeled example. Labels are aggregated via Naive Bayes. Given labels $Y^j_{1:w}$ generated by w workers for example j, NB predicts label $X^j = \hat{x}$ via:

$$\begin{aligned} \widehat{x} &= \operatorname*{argmax}_x P(X^j = x | Y^j_{1:w}) \\ &\propto P(Y^j_{1:w} | X^j) P(X^j) \\ &= \prod_{i=1}^{w} P(Y^j_i | X^j) P(X^j) \end{aligned}$$

where we assume each annotator's labels are conditionally independent. Rather than model the full conditional distribution $P(Y|X)$,we instead model annotator i's accuracy by a single accuracy parameter $p_i = P(Y = X)$.

5. EVALUATION

Simulation. As in Sheng et al. [12], we assume each label is generated by a unique annotator with accuracy independent of the particular example. Given example j with true label $X^j = x$, annotator i generates a label $Y^j_i = x$ with probability p_i. Unlike Sheng et al., we assume these accuracies are known to the system, e.g. established from past work (this assumption will be further discussed later in the paper). Annotator accuracies are drawn from a uniform distribution whose interval is varied to simulate different annotator behaviors. As in Sheng et al., we assume each annotator generates exactly one label. Whenever a new label is needed, the simulator first samples a new annotator accuracy from this uniform distribution, then samples a correct or incorrect binary label based on this accuracy and the example's true label (known to the simulator but not to the system).

Data. We report on four benchmarks also used by Sheng et al.: `Mushroom`, `Spambase`, `Tic-Tac-Toe` and `Chess:King-Rook vs. King-Pawn`[2]. Since inspection of the datasets revealed minimal class imbalance (empirical proportion of examples with $X = 1$ is 48.2%, 39.4%, 65.3%, and 52%, respectively), for the NB method we assume a simplifying uniform prior for $P(X)$ which can be ignored. Results on the last two datasets exhibited similar trends as on the first two, so we omit these latter results due to space constraints.

Learning. We adopt the same C4.5 decision tree classifier [10] implemented by J48 in WEKA[3] as used by Sheng et al. We also follow the same experimental setup of a 70/30 train/test partition, and report results of averaging across 10 trials with different random partitions. We fix the number the of seed examples at 64, and as in Sheng et al., we generate labels in pairs to avoid tie breaking for MV. As an example, assume 64 labels are generated beyond the seed set. This would yield a total of 128 single-labeled training examples for SL, while for MV and NB, we would have 32 examples with 3 labels and 32 examples with one label.

Results. Figures 1-5 compare results of SL, MV, and NB methods across five experimental conditions which vary the range of annotator accuracies simulated. Results are summarized in each Figure's caption. Note the x-axis in these figures denotes the number of *additional* labels beyond the seed data (when the methods begin to be applied). While for multi-labeling methods it would have been interesting to directly measure consensus label accuracy achieved on training data, we focus our analysis instead on the effect of these consensus labels on classifier accuracy. Since datasets used exhibit minimal class imbalance, we report simple accuracy rather than measuring precision and recall.

Overall, NB tends to perform as well or better than the other two methods. When single label accuracies are already high (Figure 1), multi-labeling has little benefit and we should

[2] `http://archive.ics.uci.edu/ml/datasets.html`
[3] `http://www.cs.waikato.ac.nz/ml/weka`

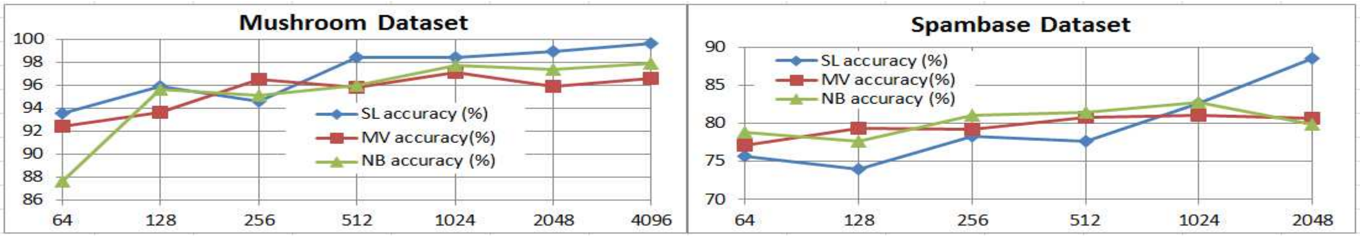


Figure 1: $p_{1:w}$ ~U$(0.6, 1.0)$. **With very accurate annotators, generating multiple labels (to improve consensus label accuracy) provides little benefit. Instead, labeling effort is better spent single labeling more examples.**

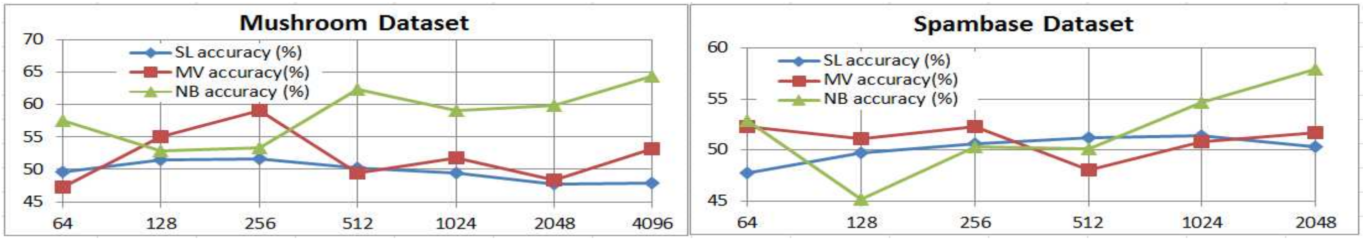


Figure 2: $p_{1:w}$ ~U$(0.4, 0.6)$. **With very noisy annotators, single labeling yields such poor training data that there is no benefit from labeling more examples (i.e. a flat learning rate). MV just aggregates this noise to produce more noise. In contrast, by modeling worker accuracies and weighting their labels appropriately, NB can improve consensus labeling accuracy (and thereby classifier accuracy).**

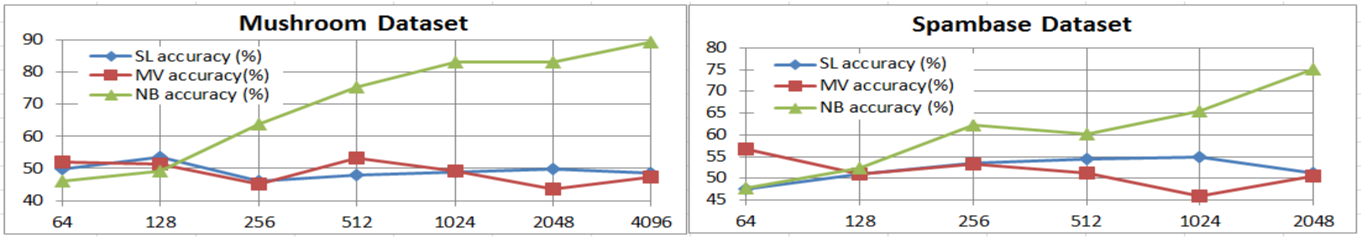


Figure 3: $p_{1:w}$ ~U$(0.3, 0.7)$. **With greater variance in accuracies vs. Figure 2, NB further improves.**

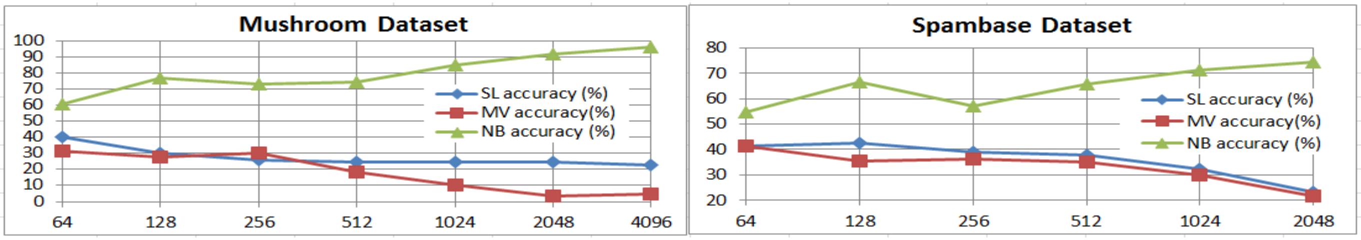


Figure 4: ($p_{1:w}$ ~U$(0.1, 0.7)$). **When average annotator accuracy is below 50%, SL and MV perform exceedingly poorly. However, variance in worker accuracies known to NB allows it to concentrate weight on workers with accuracy over 50% in order to achieve accurate consensus labeling (and thereby classifier accuracy).**

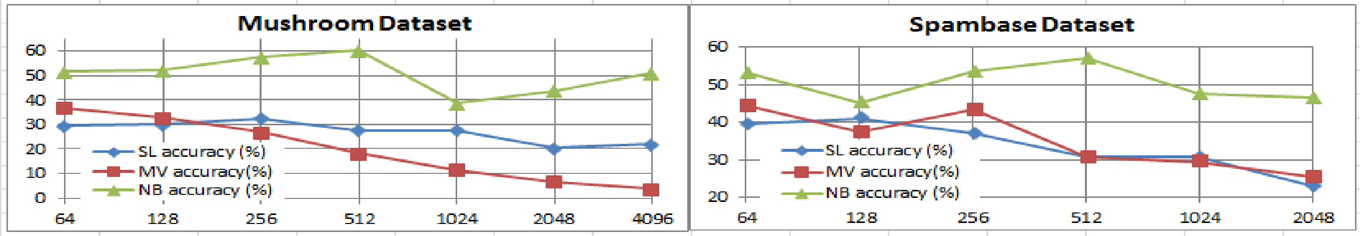


Figure 5: $p_{1:w}$ ~U$(0.2, 0.6)$. **When nearly all annotators typically produce bad labels, failing to "flip" labels from poor annotators dooms all methods to low accuracy.**

simply label more examples. As average annotation becomes noisier, however, we see both SL and MV having diminishing accuracy while NB continues to be able to effectively exploit the most accurate annotators to improve classifier accuracy. Of particular note is the *adversarial* case of annotators achieving below 50% accuracy (average accuracy is below this in both Figures 4 and 5). In the former case, NB achieves accuracies above 90% despite this adversarial average accuracy, whereas the SL and MV schemes analyzed by Sheng et al. [12] perform very poorly. Even in the latter case, NB well-outperforms the other methods.

There is an important caveat to these results to mention. While we have assumed the system has knowledge of annotator accuracies, in the experiments above, only NB is exploiting this information, putting SL and MV methods at an unfair disadvantage. To remedy this, we tried making a trivial change to SL and MV methods to always "flip" labels produced by adversarial annotators, and we repeated all of our experiments. While not shown here, results are strikingly different: all methods generally perform comparably across conditions (excepting only the case of the most accurate annotators, in which case SL continues to dominate). As such, the key lesson appears to be the importance of modeling worker accuracies at all, rather than the specific method for how these accuracies are used. In practice, annotator accuracies must be estimated from prior observations, and so system knowledge of them will be noisy. This and other assumptions of our setup here will be important to test with actual crowd annotated data in future work.

6. CONCLUSION

This paper expanded upon Sheng et al.'s investigation [12] of how labeling effort can be optimally utilized in order to maximize learner accuracy, assuming a crowdsourced environment in which labels obtain may be very noisy and exhibit high variance. Results with simulated annotators showed that incorporating knowledge of worker accuracies into the model can have a very large impact on classifier accuracy, particularly in adversarial settings. Future work will investigate these issues and findings on real crowd-annotated data.

Other follow-on work includes analysis of crowd data in order to characterize general properties of crowd labor for modeling, e.g. expected number of workers and distribution of worker accuracies as a function of task nature and difficulty, etc. While we assumed all labeling effort was used either for labeling new examples or for re-labeling, a clear generalization will be to decide at each labeling opportunity during run-time which strategy is likely to be most effective. Similarly, we would like to couple this work with traditional active learning methods in which we must decide which example to label next in order to maximally benefit the learner or reduce variance of existing labels, etc. Annotators can be better modeled via: (a) estimating their accuracies from trap-questions or inter-annotator agreeement, (b) tracking and updating dynamic worker accuracies which change over time, and (c) modeling directional errors or biases of annotators rather than modeling accuracy via a single parameter.

"Wisdom of crowds" generally suggests a group of laymen can outperform a smaller number of experts assuming certain conditions are met (e.g. independence of judgment between crowd members) [14]. Similar effects have been observed with automated systems in which combining uncertain predictions from multiple independent learners via ensemble techniques tends to outperform the best individual systems [2]. This suggests an an interesting synergy to investigate between effective ensemble methods for leveraging the crowd and automated systems in tandem. A related trend will see hybrid systems increasingly integrate human effort with automation to "close the loop" and achieve greater functionalities than either can achieve on its own [16].

7. ACKNOWLEDGMENTS

We would like to thank our anonymous reviewers for their valuable suggestions and feedback.

8. REFERENCES

[1] O. Alonso, D. Rose, and B. Stewart. Crowdsourcing for relevance evaluation. *ACM SIGIR Forum*, 42(2):9–15, 2008.

[2] R. Bell and Y. Koren. Lessons from the Netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[3] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. In *Applied Statistics, Vol. 28, No. 1.*, 1979.

[4] O. Dekel and O. Shamir. Vox populi: Collecting high-quality labels from a crowd. In *COLT*, 2009.

[5] J. Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[6] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *KDD-HCOMP*, 2010.

[7] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Exploring iterative and parallel human computation processes. In *KDD-HCOMP*, 2010.

[8] W. Mason and D. J. Watts. Financial incentives and the "performance of crowds". In *SIGKDD*, 2009.

[9] P. Norvig. Statistical learning as the ultimate agile development tool. In *CIKM*, 2008.

[10] J. Quinlan. *C4. 5: programs for machine learning.* Morgan Kaufmann, 1993.

[11] V. C. Raykar, S. Yu, L. H. Zhao, and G. H. Valadez. Learning from crowds. In *Journal of Machine Learning Research 11 (2010) 1297-1322*, MIT Press, 2010.

[12] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD*, 2008.

[13] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast—but is it good? In *EMNLP*, 2008.

[14] J. Surowiecki. The Wisdom of Crowds, 2004.

[15] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, Vancouver, 2009.

[16] T. Yan, V. Kumar, and D. Ganesan. CrowdSearch: exploiting crowds for accurate real-time image search on mobile phones. In *MobiSys*, pages 77–90, 2010.

[17] H. Yang, A. Mityagin, K. M. Svore, and S. Markov. Collecting high quality overlapping labels at low cost. In *SIGIR 2010*, Geneva, 2010.

Crowdsourcing Blog Track Top News Judgments at TREC

Richard McCreadie
School of Computing Science
University of Glasgow
Glasgow, G12 8QQ
richardm@dcs.gla.ac.uk

Craig Macdonald
School of Computing Science
University of Glasgow
Glasgow, G12 8QQ
craigm@dcs.gla.ac.uk

Iadh Ounis
School of Computing Science
University of Glasgow
Glasgow, G12 8QQ
ounis@dcs.gla.ac.uk

ABSTRACT

Since its inception, the venerable TREC retrieval conference has relied upon specialist assessors or participating groups to create relevance judgments for the tracks that it runs. However, recently crowdsourcing has been proposed as a possible alternative to traditional TREC-like assessments, supporting fast accumulation of judgments at a low cost. 2010 was the first year that TREC experimented with crowdsourcing. In this paper, we report our successful experience in creating relevance assessments for the TREC Blog track 2010 top news stories task. We conclude that crowdsourcing is an effective alternative to using specialist assessors or participating groups for this task.

1. INTRODUCTION

Relevance assessments is a crucial component when developing and evaluating information retrieval (IR) systems like search engines. Since its inception in 1992, the *T*ext *RE*trieval *C*onference (TREC) has played an important role in the IR community, creating reusable test collections and relevance assessments for a series of IR tasks. This has been underpinned by robust relevance assessments by specialist TREC assessors, or by the participating groups themselves.

However, this style of assessments also holds some profound limitations. Most notably, judgement by TREC assessors is expensive in terms of time and resources, while not being greatly scalable [2]. Furthermore, while engaging the participants for judging is free, the volume of judgments that can be produced is limited by the number of participants to the task in question.

On the other hand, *crowdsourcing* [9] has been championed as a viable method for creating relevance assessments, and indeed, as an alternative to traditional TREC assessments [2]. The reputed advantages of crowdsourcing are four-fold: judging can be performed quickly, cheaply, at a larger scale and with redundancy to achieve sufficient quality [3]. However, crowdsourcing has also been the subject of much controversy as to its effectiveness, in particular with regard to the lower quality of work produced [5], the lack of motivation for workers due to below-market wages [6] and susceptibility to malicious workers [7].

In TREC 2010, the Blog track examined real-time news story ranking within the blogosphere. In particular, participants were asked to rank news stories for a day of interest by their relative importance on that day, based upon evidence from the blogosphere [10]. Notably, 'importance' in this case is relative to the other stories published upon the same day. In this paper, we describe our successful experience when crowdsourcing relevance judgments for the Blog track top news stories task. Our contributions are three-fold: 1) we summarise the first successful instance of crowdsourcing at TREC, 2) we quantitatively assess both the crowdsourcing job itself, as well as the judgments produced and 3) we propose best practices based upon experience gained.

The structure of this paper is as follows. Section 2 describes the task that we crowdsourced, in addition to the interface and experimental setup employed. In Section 3, we detail the research questions that we investigate with regard to our crowdsourcing of relevance judgments for TREC 2010 and describe our experimental results. We provide concluding remarks in addition to some best practices in Section 4.

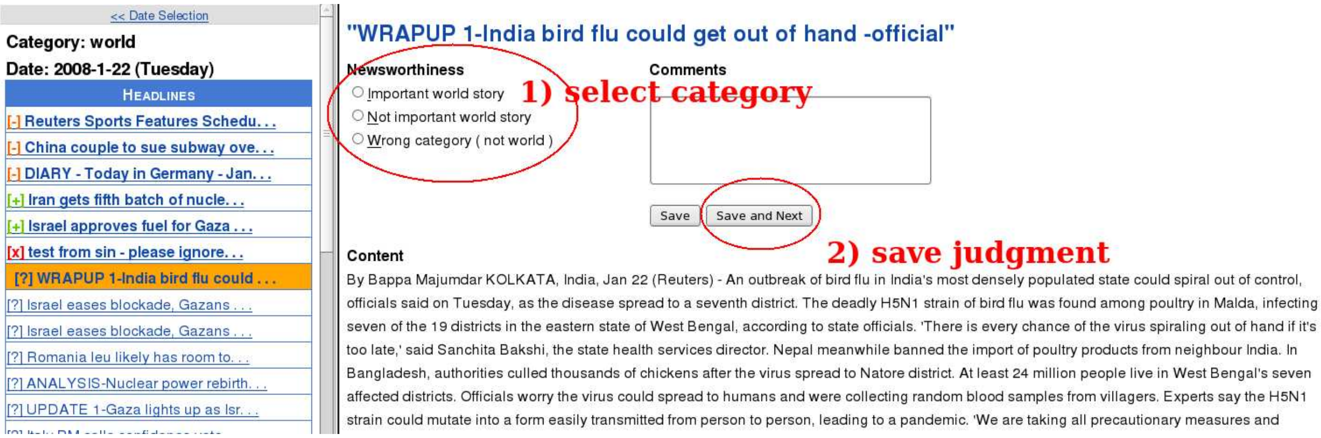
2. JUDGING NEWS STORY IMPORTANCE

The task that we address in this paper is the crowdsourcing of relevance assessments (qrels) for the Blog track top news stories task at TREC 2010. In particular, for each day of interest (query day), the participating systems returned a ranking of 50 news stories that they deemed important on that day for each of 5 news categories, namely: U.S., World, Sport, Business/Financial and Science/Technology news. The rankings from the participants were sampled using statMAP sampling [4], to a depth of 32 stories per day and category, resulting in 160 stories per day to be judged, with 8,000 stories in total [10]. The relevance assessment task is to label each of these sampled stories as important or not from an editorial perspective, such that a system's ranking based upon the blogosphere can be compared to that produced by a newspaper editor. In the following subsections we detail our crowdsourcing methodology as well as the interface that was used.

2.1 Crowdsourcing Task

We used Amazon's online marketplace Mechanical Turk (MTurk) to perform our judging. In particular, each MTurk Human Intelligence Task (HIT) covers the 32 top stories sampled for a single day and news category. For these stories, we ask workers to judge each as either: 1) Important and of the correct category, 2) Not important but of the correct category or 3) of the wrong category. To inform this judgement, the worker was presented with both the head-

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



Category: world

Date: 2008-1-22 (Tuesday)

[-] Reuters Sports Features Schedu. . .
[-] China couple to sue subway ove. . .
[-] DIARY - Today in Germany - Jan. . .
[+] Iran gets fifth batch of nucle. . .
[+] Israel approves fuel for Gaza . . .
[x] test from sin - please ignore. . .
[?] WRAPUP 1-India bird flu could . . .
[?] Israel eases blockade, Gazans . . .
[?] Israel eases blockade, Gazans . . .
[?] Romania leu likely has room to. . .
[?] ANALYSIS-Nuclear power rebirth. . .
[?] UPDATE 1-Gaza lights up as Isr. . .

"WRAPUP 1-India bird flu could get out of hand -official"

Newsworthiness

Comments

1) select category

2) save judgment

Content

By Bappa Majumdar KOLKATA, India, Jan 22 (Reuters) - An outbreak of bird flu in India's most densely populated state could spiral out of control, officials said on Tuesday, as the disease spread to a seventh district. The deadly H5N1 strain of bird flu was found among poultry in Malda, infecting seven of the 19 districts in the eastern state of West Bengal, according to state officials. 'There is every chance of the virus spiraling out of hand if it's too late,' said Sanchita Bakshi, the state health services director. Nepal meanwhile banned the import of poultry products from neighbour India. In Bangladesh, authorities culled thousands of chickens after the virus spread to Natore district. At least 24 million people live in West Bengal's seven affected districts. Officials worry the virus could spread to humans and were collecting random blood samples from villagers. Experts say the H5N1 strain could mutate into a form easily transmitted from person to person, leading to a pandemic. 'We are taking all precautionary measures and

Figure 1: A screenshot of the external judging interface shown to workers within the instructions.

line and article content of the news story. According to best practices in crowdsourcing, we had three individual workers perform each HIT [12]. From these three judgments we take the majority vote for each story to create the label. The entire task totals 24,000 story judgments spread over 750 HIT instances. We paid our workers $0.50 (US dollars) per HIT (32 judgments), totalling $412.50 (including Amazon's 10% fees).

Notably, each HIT requires 32 judgments to be made, much larger than typical MTurk HITs. The reasoning behind this decision is two-fold. Firstly, the relative nature of importance in this context requires that the worker hold some background knowledge of the other news stories of the day when judging. To this end, we asked that workers make two passes over the stories. During the first and longer pass, the worker would judge each story based on the headline and content of that story and the previous stories judged, while upon the second pass, the worker can change their judgement for any story now that they have knowledge of more news stories from that day. The second reason is one of best practice. In particular, when submitting large jobs with thousands of required judgments, it has been shown that it is advantageous to retain workers over many judgments to maintain consistency in judging [11]. By increasing the HIT size, we have each worker perform at least 32 judgments.

2.2 Judging Interface

Another notable aspect of our crowdsourcing strategy was the use of an externally hosted interface. Figure 1 shows an instance of the external interface for a single HIT. Again, the reasoning was two-fold. Firstly, our previous experiences with crowdsourcing indicates that there were bots exploiting common HIT components, e.g. single entry radio buttons/text boxes, to attempt jobs on MTurk [11]. The degree of user interaction that our external interface requires makes this unlikely to be an issue. Secondly, this interface was central to our validation strategy for the work produced. Indeed, instead of using a typical validation based upon a gold-standard judgments [12], we used colour-coded summaries of the stories and the judgments that each worker made to manually validate whether they were doing an acceptable job. In particular, we qualitatively assessed each of the 750 HIT instances based on 3 criteria, namely: *1)* are all 32 stories judged, *2)* are the judgments similar across the 3 redundant judgments and *3)* are the stories marked important sensible. Although this validation strategy appears to involve a considerable volume of work, we estimate that it took no longer than 5 hours for one person to validate all 750 HIT instances, which is comparable to the time required to create a recommended gold-standard set of 5% of the full workload size. This speed is due to the fact that colour coding of the judgments factitilate assessment of criteria 1) and 2) at 'a glance', while only a small proportion of judgments need be examined under 3). Moreover, this approach is advantagious, both because one does not have to waste judgments on validation, and by manualy assessing we can have greater confidence that the workers are judging correctly. Indeed, overall the assessed work was of good quality, with less than 5% of HITs rejected.

Lastly, following an iterative design methodology [3], we submitted our HITs in 6 distinct batches, allowing for feedback to be accumulated and HIT improvements to be made. Indeed, between each batch we made minor modifications to the judging interface and updated the instructions based upon feedback from the workers.

3. EVALUATING CROWDSOURCED RELEVANCE JUDGMENTS

In this section, we analyse our crowdsourcing job and the relevance assessments produced. We aim to determine how successful crowdsourcing was and areas where improvements can be made. In particular, in each of the following four subsections, we investigate a research question. These are:

1. Is crowdsourcing actually fast and cheap? (Section 3.1)
2. Are the resulting relevance assessments of sufficient quality for crowdsourcing to be an alternative to traditional TREC assessments? (Section 3.2)
3. Is having three redundant workers judge each story necessary? (Section 3.3)
4. If we use worker agreement to introduce multiple levels of story importance, would this affect the final ranking of systems at TREC? (Section 3.4)

3.1 Crowdsourcing Analysis

Before analysing the actual relevance assessments produced, it is useful to examine the salient features of the crowdsourcing job. In particular, it has been suggested that the crowdsourcing of relevance assessments can be completed at little cost, and often very quickly [3]. We investigate whether this was indeed the case for our TREC task.

Batch	# Query Days	# HITs	# judgments	Hourly-rate ($)
Batch 1	1	15	480	3.284
Batch 2	9	135	4320	3.574
Batch 3	10	150	4800	3.528
Batch 4	10	150	4800	5.184
Batch 5	10	150	4800	4.89
Batch 6	10	150	4800	6.056

Table 1: Average amount paid per hour to workers and work composition for each batch of HITs.

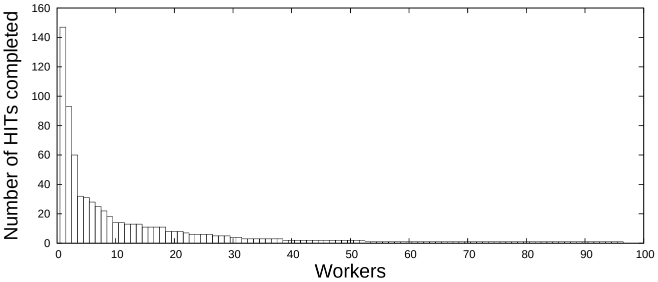


Figure 2: The number of HITs completed by each of our workers.

Prior to launching our job, we estimated that to judge the 32 stories (one HIT) it would take approximately 15 minutes, accounting for the one-off time to read the instructions and the time taken to read each story. Based upon an estimated hourly-rate (amount paid per hour of work completed) of $2, we paid a fixed rate of $0.50 per HIT. Table 1 reports the per-hourly rate paid to workers during each of the six batches. There are two points of interest. Firstly, our hourly-rate is higher than expected ($3.28 to $6.06), indicating that workers took less time than estimated to complete each HIT. Secondly, we observe an upward trend in the hourly-rate in later batches. This shows that in general, HITS in these batches took equal to or less time to complete (although there are exceptions). We believe that there are two reasons for this: firstly, between each batch we iteratively improved the instructions, hence making the task easier, and secondly, we observed a high degree of worker retainment between batches and, as such, the workers had the opportunity to become familiar with the task. Indeed, as can be seen from Figure 2, which reports the number of HITs completed by each of the 96 workers, the majority of the HITs were completed by only three workers.

In terms of the time taken by our batches, we observed a quick uptake by MTurk workers. For each of the 6 batches, the first HITs were often accepted within 10 minutes of launch, whilst the time to complete all HITs in each batch never exceeded 5 hours. Overall, crowdsourcing took a total of 8 working days to accumulate the 24,000 judgments required, including time taken by worker validation and interface improvements.

In general, we conclude that crowdsourcing judgments can be both inexpensive at $0.0156 per judgement and fast to complete. However, we believe that this task may be done 38% cheaper, as we paid above average rates for the work.

3.2 Relevance Assessment Quality

To determine the quality of our judgments, we measure the agreement between our workers. Table 2 reports the percentage of judgments for each relevance label and the between-worker agreement in terms of Fleiss Kappa [8], on average, as well as for each of the five news categories. In general, we observe that agreement on average is high (69%), lending confidence to the judgment quality. However, of interest is that agreement varies markedly between news categories. In particular, the Science/Technology and Sport categories exhibit the highest agreement with 83% and 78% respectively, while the U.S. and World categories show less agreement. Based upon the class distribution for these categories, the disparity in agreement indicates that distinquishing science from non-science stories is easier than for the U.S. or World categories. This is intuitive, as the U.S. and World categories suffer from a much higher story overlap. For example, for the story "President meets world leaders regarding climate change", it is unclear whether it is a World and/or U.S. story. Hence, workers may disagree whether it should recieve the 'important' or 'wrong category' label.

Majority (Official Qrels)	statMAP	1st Meta Worker	2nd Meta Worker	3rd Meta Worker
POSTECH_KLE	0.2206	ikm100	POSTECH_KLE	ICTNET
ikm100	0.2151	POSTECH_KLE	ICTNET	POSTECH_KLE
ICTNET	0.2138	ICTNET	ikm100	ikm100
UoS	0.1285	UoS	UoS	UoS
uogTr	0.1139	uogTr	uogTr	uogTr
ULugano	0.1000	ULugano	ULugano	ULugano
τ Correlation		0.8667	0.8667	0.7333

Table 3: Group rankings (based upon the best run submitted) using majority of three judgments against single judgments. The bottom row reports the Kendall's τ correlation between the majority and single worker rankings.

Overall, we conclude that based upon the high level of agreement observed, the relevance labels produced are of sufficient quality. Indeed, our agreement is greater than that observed in many studies of TREC assessments [1]. Hence, crowdsourcing appears to be a viable alternative to traditional TREC assessments for the Blog track top stories task.

3.3 Redundant judgments

In-line with best practices in crowdsourcing, we had three individual workers judge each HIT. However, it is important to determine to what extent this is necessary, as this is an area where costs can be dramatically decreased. To investigate this, we examine the effect of using only a single judgement on the ranking of groups that participated in TREC 2010. If the group ranking changes little, then quite possibly there is no need to have many workers judge each HIT. Table 3 reports the ranking of the six TREC 2010 groups (based upon their best run) when using the majority of the three workers (the official qrels) and the group ranking using the judgments produced by the three redundant workers individually. Furthermore, similarly to [13], Table 3 also reports Kendall's τ correlation between the group rankings produced by majority and single worker judgments. Interestingly, we observe that there is no change in the relative ranking of groups for the lower ranks, while there is a marked difference for the top three groups. As such, we conclude that redundant judging is necessary for this task, as the ranking of participating groups is not sufficiently stable at the top of the ranking, where the performances (shown in column 2) are closer.

3.4 Graded judgments

One of the advantages of using redundant judgments is that one can infer judgement confidence based on worker agreement. In particular, although not used during TREC 2010, we also created an alternative assessment set, where

Category	Important	Not	Wrong	Agreement
		Important	Category	(Kappa Fleiss)
U.S. News	21%	39%	40%	63.53%
World News	24%	38%	38%	51.69%
Sport	21%	29%	49%	77.67%
Business/Finance News	24%	43%	33%	66.88%
Science/Technology	4%	10%	86%	82.97%
Average	19%	31%	49%	68.55%

Table 2: Judgement distribution and agreement on a per category basis.

	statMap binary	statMNDCG@10 binary	statMNDCG@10 graded
statMAP binary	1.0000	0.4667	0.4667
statMNDCG@10 binary	-	1.0000	0.2000
statMNDCG@10 graded	-	-	1.0000

Table 4: Kendall's τ correlation between binary and graded relevance judgments under statMAP and statMNDCG@10 measures over the cross-category mean.

a news story's importance was measured on a three level graded scale [13]. In particular, if all workers judged a story important then the story was assigned a new 'highly important' label, two out of three workers resulted in an 'important' label, while one or no workers resulted in a 'not important' label, again following worker majority. This differs from the official binary qrels that distinguish 'important' from 'not important' only. In this section, we examine how the two level (binary) judgments compare to this three-level graded alternative. We aim to determine whether using this additional agreement evidence adversely affects the ranking of the TREC 2010 participants.

Table 4 reports Kendall's τ correlation between the ranking of groups under the binary and graded relevance judgments using the statMAP and statMNDCG@10 evaluation measures [4]. A high correlation indicates that the participating groups were not affected by the addition of a 'highly relevant' category, while a low correlation indicates that some groups favoured highly relevant stories more than others. From Table 4 we observe that the rankings produced by the binary and graded relevance assessments are not particularly well correlated, especially under statMNDCG@10. This indicates that the group ranking is affected by the addition of a highly relevant category. We believe that this merits further investigation, which we leave for future work.

4. CONCLUSIONS AND BEST PRACTICES

In this paper, we have described our crowdsourcing approach for creating relevance judgments for the TREC 2010 Blog track top news stories identification task. Based upon the high levels of agreement between our workers in addition to the manual validation that we performed, we believe that crowdsourcing is a highly viable alternative to TREC judging. Furthermore, we have confirmed the importance of redundant judging for relevance assessment in a TREC setting and shown that expanding the binary relevance assessments using worker agreement can strongly affect the overall ranking of participating groups. Indeed, we believe that this is an interesting area for future work.

Based upon our successful experience in crowdsourcing, we recommend the following four best practices in addition to those documented in [11], both for organisers of future TREC tracks considering a crowdsourced alternative, but also for the wider crowdsourcing community:

1. **Don't be afraid to use larger HITs:** As long as the workers perceive that the reward is worth the work, uptake on the jobs will still be high.

2. **If you have an existing interface, integrate it with MTurk:** There is often no need to build a new evaluation for MTurk, with a few tweaks and sufficient instruction, workers can use existing software.

3. **Gold-judgments are not mandatory:** While worker validation is essential, there are viable alternatives. We successfully validated all HITs manually with the aid of colour-coded summaries.

4. **Re-cost your HITs as necessary:** As workers become familiar with the task they will become more proficient and will take less time. You may wish to revise the cost of your HITs accordingly if cost is an issue.

5. REFERENCES

[1] A. Al-Maskari, M. Sanderson and P. Clough Relevance judgments between TREC and Non-TREC assessors. In *Proceedings of SIGIR'08.*

[2] O. Alonso and S. Mizzaro. Can we get rid of TREC assessors? Using Mechanical Turk for relevance assessment. In *Proceedings of The Future of IR Evaluation*, 2009.

[3] O. Alonso, D. E. Rose, and B. Stewart. Crowdsourcing for relevance evaluation. *SIGIR Forum*, 42(2):9–15, 2008.

[4] J. A. Aslam and V. Pavlu. A practical sampling strategy for efficient retrieval evaluation. Technical report, North Eastern University, 2007.

[5] J. Atwood. Is Amazon's Mechanical Turk a failure?, 2010. `http://www.codinghorror.com/blog/2007/04/is-amazons-mechanical-turk-a-failure.html`, accessed on 02/06/2010.

[6] C. Callison-Burch. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of EMNLP'09.*

[7] J. Downs, M. Holbrook, S. Sheng, and L. Cranor. Are your participants gaming the system? Screening Mechanical Turk workers. In *Proceedings of CHI'10.*

[8] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[9] J. Howe. The rise of Crowdsourcing, 2010. `http://www.wired.com/wired/archive/14.06/crowds.html`, accessed on 02/06/2010.

[10] I. Ounis, C. Macdonald, and I. Soboroff. Overview of the TREC 2010 Blog Track. In *Proceedings of TREC'10.*

[11] R. McCreadie, C. Macdonald, and I. Ounis. Crowdsourcing a news query classification dataset. In *Proceedings of CSE'10.*

[12] 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 *Proceedings of EMNLP'08.*

[13] E. M. Voorhees. Evaluation by highly relevant documents. In *Proceedings of SIGIR'01.*

Investigating Factors Influencing Crowdsourcing Tasks with High Imaginative Load

Raynor Vliegendhart
R.Vliegendhart@tudelft.nl

Martha Larson
M.A.Larson@tudelft.nl

Christoph Kofler
C.Kofler@tudelft.nl

Carsten Eickhoff
C.Eickhoff@tudelft.nl

Johan Pouwelse
J.A.Pouwelse@tudelft.nl

Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

ABSTRACT

This paper reports useful observations made during the design and test of a crowdsourcing task with a high "imaginative load", a term we introduce to designate a task that requires workers to answer questions from a hypothetical point of view that is beyond their daily experiences. We find that workers are able to deliver high quality responses to such HITs, but that it is important that the HIT title allows workers to formulate accurate expectations of the task. Also important is the inclusion of free-text justification questions that target specific items in a pattern that is not obviously predictable. These findings were supported by a small-scale experiment run on several crowdsourcing platforms.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.1.2 [**Models and Principles**]: User/Machine Systems—*human factors*

General Terms

Design, Human Factors, Measurement

Keywords

crowdsourcing, Mechanical Turk, user study, quality control

1. INTRODUCTION

Crowdsourcing platforms increase the ease and speed with which new search functionality can be evaluated from a user perspective. In this paper, we take a closer look at issues that arise when a search-related feature is to be evaluated, but has not yet been implemented in working form into the system. The system in question is a file-sharing system. The evaluation takes place as part of the design cycle and has the purpose of allowing us to decide which of several possible realizations of the feature will be most effective for users of the system.

During the course of designing and testing the evaluation task for the crowdsourcing platform, we realized that our task was rather different in an important respect from other, more conventional, tasks carried out by workers on crowdsourcing platforms. Specifically, we needed the workers to be able to project themselves into the role of a user of the file-sharing system and to provide feedback from the perspective of that role. The projection is necessary for two reasons, first, because the system feature that we are evaluating does not yet exist, and second, because our target group of users are general, mainstream Internet users for whom the mechanics of file sharing is rather a stretch beyond their daily online activities. In an initial exploratory phase, we noticed that there was something "special" about our task. Few workers were choosing to carry out the HITs that we published to the crowdsourcing platform, and the batch completion time was longer than was acceptable given the time constraints of our design and implementation process. Our aim was to increase the number of participants in our HIT and also the rate at which new workers took up our HIT without changing the HIT in such a way that would discourage projection or attract cheaters.

In this paper, we report on this investigations that we undertook in order to design a HIT that would achieve this aim. First, we carry out an exploratory analysis of several experimental HIT designs on Amazon's Mechanical Turk (MTurk) and formulate our findings as a series of observations. Then, we build on these observations, performing a small-scale experiment on several crowdsourcing platforms. The experiment tests two aspects of HIT design (title and free-text justifications) that we found helpful for encouraging workers to undertake projection. We refer to tasks such as our evaluation task that require workers to project beyond tangible reality and beyond their daily experience as "crowdsourcing tasks with high imaginative load". We choose the designation *imaginative load* since we see certain similarities with tasks with a high cognitive load (e.g., they take relatively long, cannot be easily routinized and are difficult to carry out in highly distracting surroundings), but have concluded it is not possible to conflate such tasks with high cognitive load tasks, which would typically require using memory or at least some factual recall effort.

The contribution of this paper is a compilation of considerations that should be taken into account when using crowdsourcing for tasks with a high imaginative load, including suggestions for choices concerning HIT design and crowdsourcing platform that make it easier to design effective HITs for such tasks. Notice that we do not report the results of the evaluation itself in this paper. Rather, we concentrate on conveying to readers the information that

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

we acquired during the design of the evaluation tasks that we anticipate will be helpful in design of further tasks.

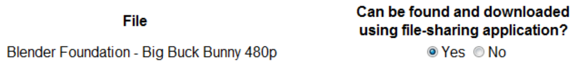
The paper is organized as follows. In the next section, we discuss related work (Section 2), then we describe the evaluation task (Section 3). In Section 4, we summarize our observations during the design and test of the task. In Section 5, we report on experiments carried out to investigate the impact of the titles and the verification on the behavior of the workers carrying out our HITs. Finally, in Section 6 we offer a summary of our conclusions.

2. RELATED WORK

In this section, we provide a brief overview of crowdsouring literature using techniques similar to ours. Often a crowdsourcing task will use a qualifying HIT to identify a set of workers who are suited to carry out the main task. In [1, 5, 4], recruitment and screening HITs were used to differentiate between serious workers and cheaters. In [3], methods to prevent workers from taking cognitive shortcuts are investigated. Many more workers completed the qualification HIT than returned to complete an actual HIT, an effect we also observe. Senstivity of workers to titles is mentioned in [5], who notice, as we do, that the selection of HIT titles influence their attraction to workers. In [2], experiments were carried out with different titles, pay rates, whether a bonus should be granted, and if so, whether this fact should be communicated to workers or not. Following the evaluation results, workers gravitate towards HITs with "attractive titles", i.e., titles which are easier to understand. In contrast to HITs that explicitly offer an additional bonus, easier-to-understand titles do not imply a high accuracy per worker. Free-text and open-ended response possibilities are often used to check whether workers had an understanding of the task, as in [5]. We make use of a similar approach, in particular asking for justifications of answers. In this respect, our work is related to that of [4], who conducted a subjective study about political opinions by asking workers to justify their given answers in free-text explanations. Giving an opinion often requires a certain degree of projection, which we equate with imaginative load. Note, however, that our task goes beyond asking mere opinions to asking workers to formulate an opinion about a feature that does not yet exist in a use context that is unfamiliar from their daily experience.

3. EVALUATION TASK

Our evaluation task involved assessing the usefulness of a time-evolving term cloud intended to make it possible for users to gain an understanding of the kinds of content that are available within a specific file-sharing system in order to facilitate browsing and search. The term cloud will offer users the possibility to find items within the system, but most importantly it is meant to allow new users unfamiliar with the system to quickly build a mental picture of what kind of content is available via the system. Users should not have to spend extensive time interacting with the system or trying out queries that are frustrating since they do not return results. In order to evaluate whether users have gained an understanding of the content available in the file-sharing system, we test their ability to distinguish five kinds of content available in the system (TV, music, books, movies and software) from five kinds not available in the system (current news, commercials, sports, how to videos and home videos). We compare this ability without the term cloud and with several different different cloud designs. Our HIT asks users to make a series of judgments on whether specific files exist in the file-sharing system. An example judgment is shown in Figure 1.

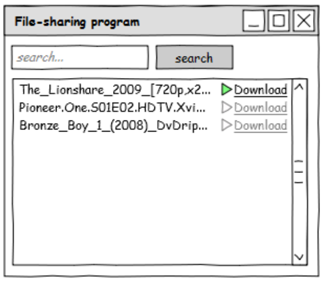


File	Can be found and downloaded using file-sharing application?
Blender Foundation - Big Buck Bunny 480p	● Yes ○ No

Figure 1: Example question from the evaluation HIT

Their answers to these questions will reflect whether or not users have generalized the information available in the term cloud into a mental picture that correctly represents the type of content in the system.

In order to prime workers to project themselves into the role of users of the file-sharing system and to discourage them from trying to use Internet search to determine which file-sharing system we are discussing and what sorts of files are present in it, we introduce the HIT with a "frame" that sets up an imaginary situation. The frame includes the following text and the diagram in Figure 2: *Jim and his large circle of friends have a huge collection of files that they are sharing with a very popular file-sharing program. The file-sharing program is a make-believe program. Please imagine that it looks something like this sketch:*



The_Lionshare_2009_[720p.x2... Download
Pioneer.One.S01E02.HDTV.Xvi... Download
Bronze_Boy_1_(2008)_DvDrip... Download

Figure 2: Mockup of a file-sharing program used to introduce (i.e., to "frame") our evaluation HIT

By naming a specific user of the file-sharing system, "Jim", we hope that users will better identify with a user of the file-sharing system, i.e., project themselves into that role.

We then ask for 10 worker judgments like the one in Figure 1. The HIT concludes with three validation questions, i.e., questions that do not ask for information necessary for the task, but rather allow us to judge the way in which the worker is approaching the task and eliminate low quality answers: (1) PrefQ, a personal preference question (multiple choice) *If you could download one of these files, which one would it be?* (2) PrefEx, a request to explain the personal preference (free-text question) *Why would you choose this particular file for download and viewing?* and (3) AnsEx, a request to justify one of the choices made while answering the 10 evaluation questions (free-text question) *Think again about the file that you chose. Why did you guess that Jim or one of his friends would have this file in their collection?* Note that there is an important difference between PrefEx, which asks workers to give a motivation for their own opinion, and AnsEx, which asks workers to give a motivation from the perspective of the role in which we would like them to project themselves, i.e., a user of the file-sharing system.

We use multiple versions of this HIT, called the "evaluation HIT", in order to collect the information necessary for our study. Most of the cases discussed here are versions of the HIT that do not contain term clouds. We are interested in gauging the user's baseline evaluation answers before exposure to the term cloud. In some cases, we also use a recruitment HIT that establishes a closed pool of qualified workers. In the next section, we discuss observations concerning our HIT made during the design and test process.

4. EXPLORATORY ANALYSIS

This section provides a qualitative discussion of the issues that we encountered during the design and testing process of our evaluation. We relate these issues to the particular nature of our task—its high imaginative load.

Recruitment and worker volume Because the evaluation needed to fit our design and implementation schedule, it was important that our evaluation HITs quickly attracted an adequate volume of workers so that the total number of assignments associated with that HIT (i.e., the batch) completed within reasonable time. We soon noticed that workers from the recruitment HIT did not continue immediately on to carry out an evaluation HIT. We started our first evaluation HIT right after manually handing out qualifications to the 81 workers that completed our recruitment HIT successfully. Since the recruitment HIT took less than 24 hours to complete, we initially assumed that the evaluation HIT would complete within roughly the same amount of time. However, only 10 out of 405 HIT-assignments offered were completed the next day. A second recruitment yielded 79 new qualified workers, but only one of them took up the main evaluation HIT within 24 hours. We conjectured that this slow uptake was due to the mismatch in expectations raised by the recruitment HIT. The recruitment HIT was titled "Like movies and music? Earn qualification with a background survey and two short questions". It contained a list of relatively easy to answer background questions, but only one question containing titles as in Figure 1. In short, it did not reflect the focus of the main HIT. Workers were possibly misled to believe the main HIT would be more related to music and movies and did not expect to receive questions like Figure 1 in the main HIT. There are two possible interfering factors affecting the volume of workers: reward level, which we did our best to optimize before publishing this HIT, and total number of assignments available to workers. During a previous crowdsourcing project, e-mails from workers suggested that HIT popularity is related to offering a large volume of assignments and keeping them in steady supply. Because our recruitment HIT asks for free-text answers that must be individually judged, it is not possible to automate the assignment of qualifications in our evaluation, and for this reason the slow worker uptake was a real concern. We decided to publish an "open" evaluation HIT, i.e., one that did not require workers to earn a qualification, and were surprised that the quality of the responses to the free-text validation questions remained very stable. Apparently, our HIT has an aspect of its design that discourages workers who are not serious and makes recruitment less necessary.

Matching strategies. Because our evaluation task is attempting to gather information about people's mental pictures and not about the external world, there are no "correct answers" to the task questions. We could enlarge our HIT with questions for which the answer is known – a popular method for quality control – but the workers' ability to answer the control questions is not guaranteed to reflect the quality of their evaluation answers. For our task, it is more important to control for the strategy the worker is using to answer the question. In particular, we need the workers to be projecting themselves into the role of the user of the file-sharing application and not applying a strategy that reflects an external source of information (such as making use of general Internet search). A particular danger in the case of the evaluation HIT is that workers will try to apply a matching strategy using the information given in the "frame" of the HIT. In other words, it is possible that workers answer the evaluation questions by literally comparing the filenames in the example in Figure 2 or the terms in the term cloud (described in Section 3, but not pictured) to the filenames in Figure 1. Reading the explanations of why the workers thought that certain files were in the file-sharing system (i.e., the answer to AnsEx), it was clear that a few of the workers would base their decision on literal matches (e.g., one answers "cloud contains DVDRIP"). However, the majority were attempting to generalize the situation and make a decision on the basis of what kind of media enjoy overall popularity (in the case which does not include the term cloud) or what general categories of content are represented in the term cloud (e.g., one answers, "With the cloud screens showing words like programming and microsoft, I think this file should be available in the collection").

5. FURTHER INVESTIGATION

We carried out a small-scale experiment run on several crowdsourcing platforms in order to further investigate the impact of title choice and of the validation questions on the quality of the workers' responses. Each version of the HIT was made available to workers with a total of 50 assignments (5 sets of 10 different filenames to be judged) paying US$0.10 each. Results are reported in Table 1 in terms of batch statistics: number of assignments that we rejected due to obvious non-serious workers (e.g., blank text boxes), total number of workers participating, effective hourly rate, run time needed to complete the batch and median time between arrivals of new workers to work on the HIT-assignments.

Table 1: Batch statistics for the five experimental conditions (varying title and validation questions) on MTurk

	Title A	Title B	Title C	Only AnsEx	No PrefEx
#Rejected assignments	0	0	2	0	0
#Workers	25	22	19	17	20
Effective hour. rate	$2.54	$2.08	$1.76	$3.13	$1.51
Run time	50h28m	13h45m	19h13m	15h55m	20h45m
Med. arrival interval	67m36s	24m05s	18m41s	17m22s	35m06s

We experimented with three titles. Title A ("Jim, his friends and a make-believe file-sharing program"), which emphasized the imaginative nature of our HIT by including reference to “make believe”. Title B (“Jim, his friends and digital stuff to download”), de-emphasized the fact that the HIT involved file sharing, terminology we thought might seem overly technical to workers. Title C (“Jim, his friends and interesting stuff to download”), which attempted to make the HIT generally attractive to a wide audience. We also experimented with omitting our validation question in order to understand which ones were important for maintaining high quality answers. We ran a version of our HIT which only asked for an explanation of the answer to the evaluation questions (“Only AnsEx”) as well as a HIT that asked for a personal preference, but did not ask for that personal preference to be justified (“No PrefEx”). For completeness, we include a list of the limitations of this experiment, necessarily imposed by its small scale and short duration: We were able to control for temporal variation by starting each version of the HIT at approximately the same time on consecutive weekdays. We did not control for differences among weekdays or for the effects of holidays (for example, Title A ran the day before the Thanksgiving holiday in the US and we are careful not to read too much into its significantly longer runtime). We did not control for workers becoming acclimated to us as a requester and thereby more inclined to do our HITs. We simply checked that the number of workers that participated in multiple conditions remained limited (2–5). In this way, we know that our results are not dominated by workers who are developing strategies on how to approach the task from one HIT version to the next.

The following generalizations emerge from our investigation. First, all HITs yielded serious results—in only two cases did we reject an assignment completed by a worker due to blatant cheating. Second, the generally attractive title (Title C) seemed to attract workers at a better rate, but needed a longer total run time than Title B. Only requiring an explanation of the answer and not of personal opinion attracted workers quickly and also improved the total running time. However, here we noticed that we attracted two types of workers: first, workers who were taking the HIT seriously, spending relatively long to complete it and giving thoughtful answers to AnsEx and, second, workers who approached AnsEx with a “quick and dirty” strategy. Either these workers realized that the same answer was more or less applicable to all 5 sets of ten filenames and copied and pasted the same answer for each HIT-assignment that they completed or they fell into trivial non-specific observations, such as “That’s what people share”. In order to understand this effect, it is important to note that the wording of AnsEx was necessarily affected by the removal of the personal preference question from the “Only AnsEx” condition. It was no longer possible to ask for an explanation concerning the file that the user had picked. Instead of the original wording, the question was changed to “Think about the files that you thought were available for download. Why did you guess Jim and his friends would have these files in their collection?” This relatively small change meant that the question no longer targeted one specific file—the generality of the question apparently was enough to encourage non-serious workers to apply cut and paste strategies. Interestingly, the workers that answered the AnsEx question seriously in the “Only AnsEx” version of the HIT gave more elaborate answers than the workers doing the version of the HIT that required them to answer multiple validation questions. Also interesting was that the “No PrefEx” condition, which omitted the question requiring workers to justify their personal interests, yielded thoughtful answers on the AnsEx question, suggesting that the PrefEx question is not necessary. We would like to note that because the number of workers was relatively small, a single worker with a particular style (e.g., tending to apply a matching strategy) could have an inordinately large influence on the outcome of the experiment. If it is not possible to completely control for worker style, it appears important to use a quite large pool of workers in order to ensure the generality of results.

We ran the same set of experiments on other available crowdsourcing platforms to make a cross-platform comparison. Gambit and Give Work did not yield any judgments at all. This finding was largely independent of the financial reward offered. We conjecture that the lack of uptake may be due to technical limitations (mobile device, etc.) or a consequence of a different culture of HITs on these platforms. Samasource seems to be a very difficult platform to use. There were several negative observations to be made with our current experiment setup: Largely independent of title or question style we notice a very high share of uncreative copy and paste answers. Additionally there seem to be issues with their worker identification system as we have multiple submissions from different worker ids, that were issued from the same IP address and contained identical copy & paste answers. The very impressive exception to this trend was one worker from Nairobi who provided extremely detailed, informed and well-written answers.

6. CONCLUSIONS

We conclude that “high imaginative load” tasks can be successfully run on MTurk. The key appears to be a combination of signaling to workers the unique nature of the task, possibly quite different than tasks they generally choose, and at the same time making each HIT-assignment require a highly individualized free-text justification response.

7. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Commission’s 7th Framework Programme (FP7) under grant agreement № 216444 (NoE PetaMedia).

8. REFERENCES

[1] J. S. Downs, M. B. Holbrook, S. Sheng, and L. F. Cranor. Are your participants gaming the system? Screening Mechanical Turk workers. CHI ’10, pages 2399–2402, 2010.

[2] C. Grady and M. Lease. Crowdsourcing document relevance assessment with Mechanical Turk. In *NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s MTurk*, pages 172–179, 2010.

[3] A. Kapelner and D. Chandler. Preventing satisficing in online surveys: A “Kapcha” to ensure higher quality data. In *CrowdConf ACM Proceedings*, 2010.

[4] S. A. Munson and P. Resnick. Presenting diverse political opinions: how and how much. CHI ’10, pages 1457–1466, 2010.

[5] K. T. Stolee and S. Elbaum. Exploring the use of crowdsourcing to support empirical studies in software engineering. ESEM ’10, pages 35:1–35:4, 2010.

Estimating the Completion Time of Crowdsourced Tasks Using Survival Analysis Models

Jing Wang
New York University
jwang5@stern.nyu.edu

Siamak Faridani
University of California, Berkeley
faridani@berkeley.edu

Panagiotis G. Ipeirotis
New York University
panos@stern.nyu.edu

ABSTRACT

In order to seamlessly integrate a human computation component (e.g., Amazon Mechanical Turk) within a larger production system, we need to have some basic understanding of how long it takes to complete a task posted for completion in a crowdsourcing platform. We present an analysis of the completion time of tasks posted on Amazon Mechanical Turk, based on a dataset containing 165,368 HIT groups, with a total of 6,701,406 HITs, from 9,436 requesters, posted over a period of 15 months. We model the completion time as a stochastic process and build a statistical method for predicting the expected time for task completion. We use a survival analysis model based on Cox proportional hazards regression. We present the preliminary results of our work, showing how time-independent variables of posted tasks (e.g., type of the task, price of the HIT, day posted, etc) affect completion time. We consider this a first step towards building a comprehensive optimization module that provides recommendations for pricing, posting time, in order to satisfy the constraints of the requester.

Keywords

crowdsourcing, mechanical turk, survival analysis

1. INTRODUCTION

Crowdsourcing has been used in a variety of different applications. Researchers have used Mechanical Turk to perform user experiments, online businesses have used it to extend the capabilities of their platforms, and in some cases it has been even used in search and rescue operations. By harnessing crowdsourcing Bernstein et. al [1] have built a MS Word plug-in that can help writers perform difficult copy editing tasks on their documents (for example for changing all of the active sentences to passive voices). Bigham et. al [2] use Amazon Mechanical Turk to help blind people locate objects in their environment.

For many of different crowdsourced tasks it is important to have an estimation of the completion time. It is known that the number of subtasks and the monetary rewards for a task are two main factors that contribute to the completion time for the task. For example Mason and Watts study the effect of financial incentives and the performance of online Turkers. Their study shows that even though the quantity of work increases by increasing the financial incentives, the quality of the work shows no significant increase [11]. In this paper we highlight other factors that contribute to this completion time. For example by using a topic model based on Latent Dirichlet Allocation (LDA) we show that in our dataset transcribing tasks are picked up faster than other groups of tasks. Using a survival analysis framework, we provide an extendable and well-studied approach for predicting the completion time for a crowdsourced task based on different factors of identical subtasks.

2. DATA SET

We first present the descriptive results about the distribution of completion times on Mechanical Turk. We estimated the completion time of the tasks by monitoring hourly the overall state of the Mechanical Turk market, and capturing the content and position of all available HITs. (See [8] for more details.) From January 2009 through April 2010, we collected 165,368 HIT groups, with 6,701,406 HITs total, from 9,436 requesters. The total value of the posted HITs was $529,259. To estimate the lifetime of a HITgroup, we counted the time since the first time we saw a particular HITgroup (each HITgroup has a unique id), until the last time.

Figures 1 and 2 show the count-count distribution and the CDF (Cumulative distribution function) of the completion times. We can observe that the completion times have power-law distribution. In contrast to the “well-behaving” systems with exponentially-distributed waiting times, a system with a heavy-tail distribution can frequently generate waiting times that are larger than the average waiting time.

Given that this is a power-law distribution, the sample mean is not the same as the mean of the distribution. To estimate the distribution mean, we rely on the maximum likelihood method for power-law distributions. The analysis works as follows. Given that the distribution is a discrete power-law distribution, we have:

$$Pr\{duration = x\} = x^{-\alpha}/\zeta(\alpha) \qquad (1)$$

where $\zeta(\alpha) = \sum_{n=1}^{\infty} \frac{1}{n^{\alpha}}$ is the Riemann zeta function, serving as normalization function, and α is the parameter of the distribution. For estimating the parameter α, fitting a regression line on the plot is not helpful. Instead it is better to use

Copyright is held by the author/owner(s). WSDM 2011 Workshop on Crowdsourcing for Search and Data Mining (CSDM 2011), Hong Kong, China, Feb. 9, 2011.
Copyright 2011 ACM 978-1-4503-0222-7 ...$10.00.

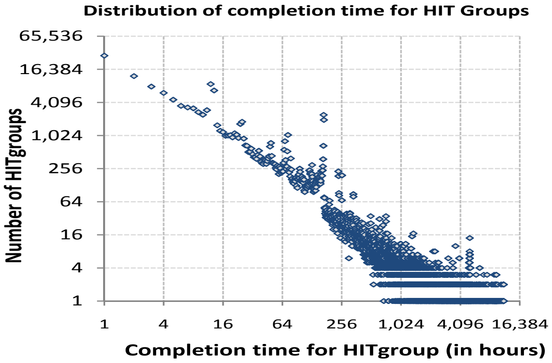


Figure 1: Completion time of a task depends on the number of individual subtasks (HITs) that constitute the task.

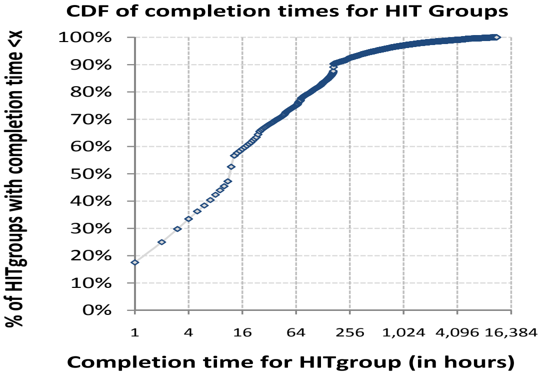


Figure 2: CDF for completion time $Pr(duration \geq x)$

the maximum likelihood estimator:

$$\frac{\zeta'(\hat{\alpha})}{\zeta(\hat{\alpha})} = -\frac{1}{n}\sum_{i=1}^{n}\ln(x_i)$$

In our case, the x_i values are the observed durations of the tasks. In the case of Mechanical Turk, we have:

$$\frac{\zeta'(\hat{\alpha})}{\zeta(\hat{\alpha})} = -2.3926$$

Using this result, we can estimate the distribution of the task completion time, and (by extension) its mean and variance. By looking up the table [13] with the values of $\frac{\zeta'(\hat{\alpha})}{\zeta(\hat{\alpha})}$, we find that the most likely value for $\hat{\alpha}_{MTurk} = 1.35$. A power-law distribution with $\alpha \leq 2$ does not have a well defined mean value. In other words it is infinite. In such cases, the sample mean is never a good representation of the distribution mean. Instead, the mean value for the sample is expected to increase over time, without ever converging to a stable value.

In a stable system, we expect the average completion time to increase initially (due to censoring effects, i.e. not observing long tasks) and then stabilize. On Mechanical Turk, though, the average completion time increases with time. This is mainly due to "abandoned" tasks that remain available until their expiration (which does *not* mean they were completed). This result indicates that we need to have some better estimates of completion time, in order to understand better what causes tasks to linger around for many days and weeks. Our preliminary analysis in this paper will provide some early clues.

3. STATISTICAL MODEL FOR EXPECTED COMPLETION TIME

We use survival analysis to build a predictive model for determining the expected completion time. When data, in our case, completion times, is not normally distributed, standard linear models cannot be used on the data. One important method that is used in this case by biologists, epidemiologists, and reliability engineers is survival analysis. Survival analysis is the analysis of the lifespan of an entity [5] (also see [9] for more). In this case we study the lifespan of a task. Additionally by using more sophisticated Shared Frailty Survival Models that are used for unobserved heterogeneity shared by clusters of tasks, we can improve our results. For more information on the shared frailty model see [6] and [7].

3.1 Variables used in the prediction model

In our study, we used variables that are time-independent and we extract these variables for the time the task was first posted. We use these variables as the main factors for the survival analysis models. These factors can be categorized into the following categories

Requester Characteristics: Activity of requester at time of submission (Number of total HITs/HITgroups by the requester, Total amount of money spent by requester so far), Existing lifetime of requester (how many days since first HIT posted), Average lifetime of prior HITs posted.

Market Characteristics: Day of the week, Time of the day, Total number of competing HITs/Rewards/HITgroups in the market at the time of posting

HIT Characteristics: Price, number of HITs, length in characters, HIT "topic" extracted using latent Dirichlet allocation (LDA) [3].

3.2 Generative topic model: LDA

To generate the "topics" for the available HITs, we used the keywords assigned to each HIT. (In the future we plan to use the words in the title, description, and in the actual HTML of the HIT.) Also, we add a "NoKeyword" category to represent HITs with no keywords input. The assumption for the LDA model is that each document is a mixture of topics, whose distribution has a Dirichlet prior. A topic has probabilities of generating various terms, characterized by a topic-dependent word distribution. We estimate the parameters of our topic model using variational EM algorithm [3]. The key parameter that we are interested in is $P(topic = k) = \theta_k = \exp(E[\log(\theta_k)]) = \exp[\digamma(\gamma_k) - \digamma(\sum_{k=1}^{K}\gamma_k)]$. In the analysis, we assign each HIT to the topic with highest probability (for the convenience of topic-stratified analysis). In Section 4 and 5, we used seven topics. In the future, we plan to use the hierarchical version of LDA, which does not require the specification of a predefined number of topics. Below, you can see a list of keywords for a few topics that were identified as important:

- **Topic1:** cw, castingwords, podcast, transcribe, english, mp3, edit, snippet, confirm
- **Topic2:** article, writing, write, data, review, collection, blog, writer, easy, freelance, rewrite, articles
- **Topic5:** editing, rewriting, paul, pullen, writing, sentence, dinkle

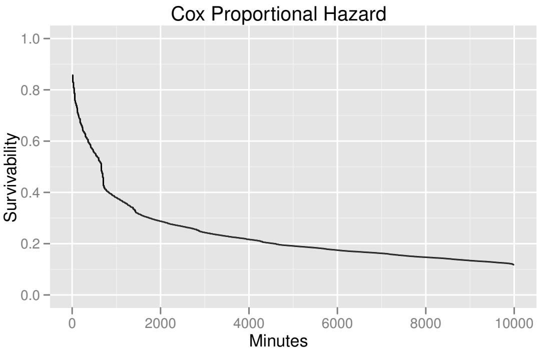


Figure 3: Fitting a Cox proportional hazards regression model to the data collection from Amazon Mechanical Turk

- **Topic6:** answer, question, writing, opinion, advice, research, questionswami, seo, contentspooling

4. SURVIVAL ANALYSIS

In survival analysis models it is assumed that completion times and censoring times are independent. "Censored" completion times are the completion times of the tasks that expired before being completed, or tasks that have been suddenly taken down by the requester. (We can detect that by observing the usual completion rate, and see if the disappearance of the task cannot be explained based on the prior completion rate.) This assumption holds for our problem. We perform the survival analysis using the `survival` package in R. Figure 3 is generated by fitting a Cox proportional hazards regression model to our data set. It implies that, in general, 75 percent of tasks are completed within two days.

4.1 Stratified survival analysis

The Cox regression model has a set of strict assumptions about the characteristic of the variables that can be used (the "proportional hazards" requirement.) Unfortunately, we also have useful variables that are not satisfying this requirement. A straightforward way to incorporate variables into a survival analysis is to use a stratified survival analysis, and examine the results without worrying about proportionality. We do such analysis for a set of variables: price, number of HITs, day of the week, time of the day, and HIT topic.

The experimental results in Figure 5 show that there are significant survival rate differences across groups stratified by HIT characteristics (price, number of HITs, and HIT topic). Not surprisingly, a higher price will shorten the completion time of the HIT, while a larger number of HITs will slow down the task completion. However, we can not see such big effects for market characteristics (day of the week, and time of the day). This was relatively surprising, given our prior "feeling" and experience in the market. A plausible explanation is that we are focusing on relatively long running tasks: the tasks in our dataset have been running for hours and days, not just minutes. So, our (still preliminary) analysis should also be interpreted as targeting longer running tasks. For a set of nice optimizations for handling tasks that require completion within very short periods of time, please see the techniques used by Bigham et. al [2].

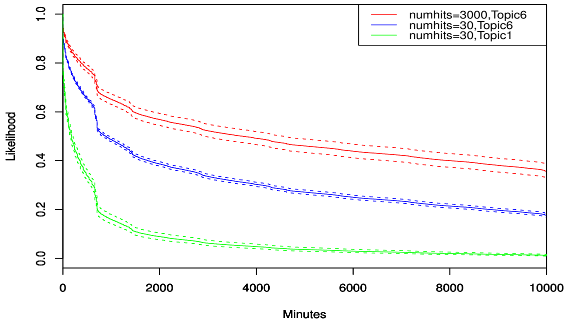


Figure 4: Model prediction for tasks with three different conditions.

5. PREDICTION

Our analysis generated the completion time for "normalized" HITs, removing the effect of other variables and allowing us to understand the effect of a single variable in the prediction time for the task. Using the results, we can generate predictions about the completion time of various HITs. We present a few examples here. We consider tasks under three different conditions: 1) A "Topic6" task with 3000 HITs; 2) A "Topic6" task with 30 HITs; 3) A "Topic1" task with 30 HITs. All the other variables are identical: we choose Monday for day of the week, 6pm-Midnight for time of the day, and median values for other non-binary variables. The prediction results are shown in Figure 4. Notice that the transcription tasks (posted mainly by CastingWords) are being finished up quickly, reflecting also the fact that CastingWords is a long-term reputable requester in the market. Notice that our analysis shows just the time for the HIT to be picked up by a worker, and cannot observe for how long a particular worker has been working on the actual task. (Prior work [10] indicates that the working time for HITs tends to follow a log-normal distribution, reflecting the unequal difficulty of the individual HITs.)

6. MODEL EVALUATION

We present some very preliminary results on model evaluation. For our experiments, we divided our data set randomly into two parts with almost equal number of HITs assigned to each part. We use the training set for estimating the parameters of the Cox model; we use the test set to evaluate whether the parameters of the model provide a good fit for the actual data in the test set.

We use the likelihood ratio (LR) test of statistical significance. (Note that due to the non-Gaussian lifetimes, measuring estimation errors is not ideal.) After generating the parameters $\hat{\beta}_{(train)}$ of the Cox model for the training set, we then estimate the Cox log partial likelihood $l^{(test)}\hat{\beta}_{(train)}$ of observing the lifetimes of the tasks in the test set. We also compute the likelihood $l^{(test)}(0)$ for the null model, i.e., predict a constant value for the lifetime of a task. The LR statistic is 8434 (larger than $\chi^2_{25} = 37.65$), which indicates a good model fit.

We should note that statistical significance does not automatically mean practical significance. We want to examine whether the types of HITs vary over time. Also, we want to examine whether predictions are possible to carry across requesters. Finally, we want to use time-varying characteristics

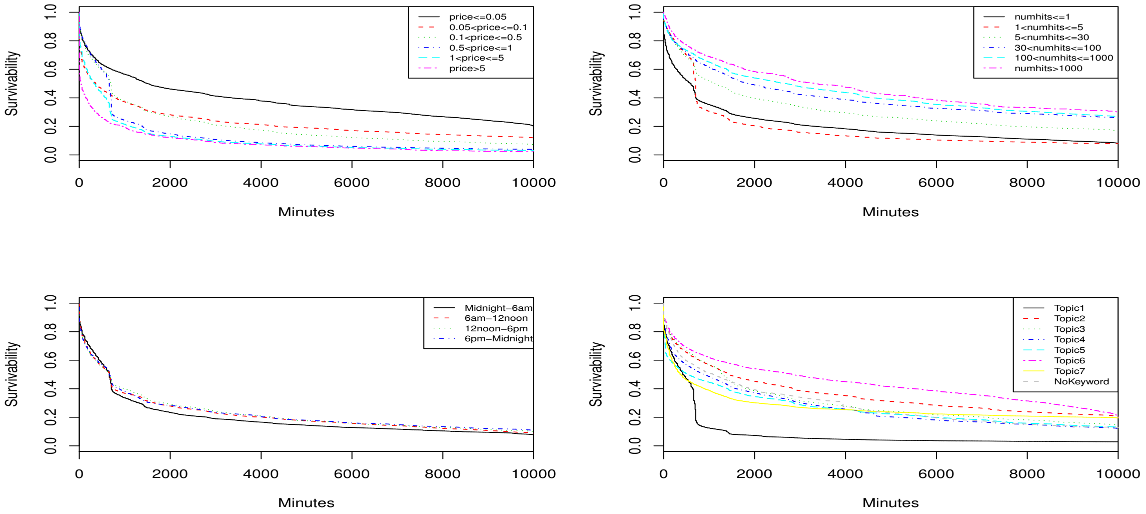


Figure 5: Stratified analysis for price, number of HITs, time of the day, and HIT topic.

of the HITs (e.g., in which page is currently the HIT listed?) to see their effect in the lifetime of the HITs.

7. FUTURE WORK: COMPLETION TIMES FROM THE QUEUING THEORY PERSPECTIVE

We would like to study the process of Turker arrivals to the system from the queuing theory perspective, to complement the survival analysis approach. Brown et. al [4] study a similar problem in a call center setting. They study three fundamental components in the service process. First, they provide a stochastic model for arrivals, they study customer abandonment behavior and lastly they study service durations. Interestingly Brown et. al, use the Kaplan-Meier model to characterize the waiting time for service or abandoning. Similar to Brown's model we can assume that Turkers arrive to Amazon as a Non-homogeneous Poisson Process (NHPP) with rate $\lambda(t)$. In another context Vulcano et. al [12] use a choice based model where arrivals are NHPP and each person follows a multinomial logit model to select one of the choices (here HITs). These assumptions can be verified empirically by looking at our data base.

8. CONCLUSION

We showed that completion times follow a heavy tail distribution. We demonstrated that sample averages cannot be used to predict the expected completion time of a task. We proposed a model based on survival analysis model and by fitting a Cox proportional hazards regression model to the data collected from Amazon Mechanical Turk, showing the effect of various HIT parameters in the completion time of the task. We believe that this work can serve as a good basis for building a more sophisticated and comprehensive system for this task.

9. REFERENCES

[1] M. Bernstein, G. Little, R. Miller, B. Hartmann, M. Ackerman, D. Karger, D. Crowell, and K. Panovich. Soylent: A Word Processor with a Crowd Inside. 2010.

[2] J. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R. Miller, A. Tatarowicz, B. White, S. White, and T. Yeh. VizWiz: nearly real-time answers to visual questions. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, pages 1–2. ACM, 2010.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, and L. Zhao. Statistical analysis of a telephone call center. *Journal of the American Statistical Association*, 100(469):36–50, 2005.

[5] P. Dalgaard. *Introductory statistics with R*. Springer Verlag, 2008.

[6] J. Fine, D. Glidden, and K. Lee. A simple estimator for a shared frailty regression model. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 65(1):317–329, 2003.

[7] R. Gutierrez. Parametric frailty and shared frailty survival models. *Stata Journal*, 2(1):22–44, 2002.

[8] P. G. Ipeirotis. Analyzing the amazon mechanical turk marketplace. *XRDS*, 17:16–21, December 2010.

[9] D. Kleinbaum and M. Klein. *Survival analysis: a self-learning text*. Springer Verlag, 2005.

[10] S. Kochhar, S. Mazzocchi, and P. Paritosh. The anatomy of a large-scale human computation engine. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pages 10–17, 2010.

[11] W. Mason and D. Watts. Financial incentives and the performance of crowds. *ACM SIGKDD Explorations Newsletter*, 11(2):100–108, 2010.

[12] G. Vulcano, G. van Ryzin, and R. Ratliff. Estimating primary demand for substitutable products from sales transaction data. Technical report, Working paper, 2008.

[13] A. Walther. *Acta Mathematic*, 48:393, 1926.

Crowdsourcing Interactions

A proposal for capturing user interactions through crowdsourcing

G. Zuccon, T. Leelanupab, S. Whiting, J. M. Jose, and L. Azzopardi
School of Computing Science, University of Glasgow
Glasgow, G12 8RZ, United Kingdom
{guido,kimm,stewh,jj,leif}@dcs.gla.ac.uk

ABSTRACT

Crowdsourcing has been proposed as an inexpensive and often efficient way of outsourcing work tasks to a large group of people. In this paper, we propose to use crowdsourcing as strategy for acquiring users' interactions within interactive information retrieval (IIR) systems. We are interested to understand whether crowdsourcing represents a robust strategy that can be used in conjunction with common approaches for capturing interactions, in particular laboratory user experiments. What are the similarities, differences, advantages and disadvantages of crowdsourcing interactions compared to traditional strategies? To investigate these issues, we outline the design of a procedure where interactions can be captured using crowdsourced workers. We expose the problematic issues that arise during the design process, together with preliminary statistics and results acquired by implementing our protocol within Amazon Mechanical Turk. This work opens up a number of research prospectives, the most appealing being a new methodology for the evaluation of IIR systems based on crowdsourcing.

1. INTRODUCTION

Collecting interactions between users and search system is fundamental for the analysis of user behaviours and system efficiency in the study of IIR systems. Users' interactions are often captured using search system logs. In practice, such interaction logs can be acquired in two ways: either from the system logs of search engines, in a completely naturalistic manner, or by set up experiments where users are invited to perform some pre-defined simulated information seeking tasks. Both techniques have several advantages, as well as disadvantages.

Obtaining search interactions through the analysis of query-logs generated by search engines whilst inexpensive, is virtually impossible, unless the organisation who owns the search engine grants access to this resource. Unfortunately, this is often not the case for academic researchers [5]. A further problem is that there is no control on the user population whose interactions have been captured. In fact, in these cases neither researchers have entry data about the user population, such as demographical information (e.g. age, sex, nationality, education, etc), level of confidence with the search technology and the information seeking task, nor can they obtain post-search task feedback from the users, such as their level of satisfaction about the search experience, level of achievement of the search goals, etc. These disadvantages are however mitigated by the availability of a large number of (often) heterogeneous user interactions, since everything users search for is logged by the retrieval systems.

Conversely, setting up laboratory user studies to capture search interactions with IIR systems is generally costly, as participants are usually paid at the minimum hourly wage. This limits the number of participants in laboratory-based user studies. Thus, the collected data is often several orders of magnitude smaller than what is acquired by search engines' query-logs. Moreover, participants are often recruited within an homogeneous user population. For example, in the case of researchers based within universities, users are often recruited within the university's student population. However, in laboratory user studies researchers have extensive control over the participants. Population observations such as demography, familiarity with search technologies/tasks, etc., can all be collected. Similarly, post search task feedback can be acquired explicitly from the users, e.g. using questionnaires or interviews.

In this paper, we propose an alternative approach to the IIR experiment methodology, based upon *crowdsourcing.* Crowdsourcing has been proposed as an inexpensive and often efficient way to conduct large-scale focused studies [7], and has been implemented in a number of web-based platforms such as Amazon Mechanical Turk (AMT) and CrowdFlower. The crowdsourcing paradigm has been recently used in information retrieval for performing a number of tasks. For example, Alonso et. al. crowdsourced relevance assessments by asking workers to evaluate the relevance of results retrieved by a geographical IR system [2]. While, Alonso and Mizzaro compared crowdsourced relevance judgements against the correspondent judgements obtained by TREC assessors [1] .

The intuition underlying the crowdsourcing-based user experiments we propose is that workers are asked to complete information seeking tasks within a web-based crowdsourcing platform. While workers perform information seeking tasks, researchers can capture logs of workers interactions with the IIR system. Furthermore, researchers have the possibility to acquire entry and post-search information and statistics, which would help to characterise (to some extent) the user population. This procedure might appear similar to laboratory-based experiments, and for this reason in this paper we focus on these two strategies. Note however that the inherent characteristics of crowdsourcing differentiates the two strategies. In section 3, we examine the diversities between crowdsourcing-based and laboratory-based (which are reviewed in section 2) IIR paradigms for capturing user

CSDM'11, WSDM 2011 Workshop on Crowdsourcing for Search and Data Mining (CSDM 2011), Hong Kong, China, Feb. 9, 2011
ACM 978-1-60558-896-4/10/07.

interactions in information seeking tasks. The presence of such diversities calls for the definition of a new protocol for crowdsourcing-based IIR experiments, which is outlined in section 3. Thereafter we describe how we plan to validate the protocol, and we report preliminary results and statistics, together with open issues of the IIR experiments we executed on AMT (section 4). Finally, in section 5 we discuss the prospectives that the protocol for IIR experiments based on crowdsourcing opens up for IR research, the most appealing being a novel methodology for the *evaluation* of IIR systems.

2. TRADITIONAL PARADIGMS FOR INTERACTIVE IR EXPERIMENTS

The acquisition of interaction data in laboratory based IIR settings follows the indications put forward by Borlund in the paradigm for the evaluation of IIR systems [3]. Within this scheme, information needs are treated for individual users with respect to search tasks and simulated situations. In a cognitive perspective, the knowledge and the perception of the search context is represented by the user's interactions with the IIR system. According to Borlund, the IIR evaluation model is composed of three main aspects:

a. Experimental *settings and protocols*, e.g. the Latin-square procedure, and sequences of pre-defined tasks and questionnaires.

b. Simulated *work task* situation, that provides adequate imaginative context to initiate user's information needs.

c. Set of alternative *performance measures* regarding user behaviours via logs (e.g. completion time, entered queries terms, read results, etc.), as well as user perception and search experience via questionnaires and interview (e.g. open question, Likert scale, or semantic differentials).

In the traditional IIR model the focus of the evaluation is on the behaviour of users performing search. During the search session, a user interactively searches, interprets, and modifies the search as well as the relevance assessment with respect to the perception of information needs and simulated task situations. Furthermore, Borlund suggests that participants should be from backgrounds similar to the simulated situation designed by the experimenters. However, this is not always the case, as university students are often employed by academic researchers for performing laboratory based IIR experiments, e.g. [9].

3. A PROTOCOL FOR INTERACTIVE IR BASED ON CROWDSOURCING

In the following we outline a protocol for conducting IIR experiments, and thus capture interaction data, within crowdsourcing platforms. Some of the considerations we develop in the following are based on the tools provided by AMT, but can be extended and adapted to other crowdsourcing platforms, such as CrowdFlower.

The protocol we propose prescribes that workers are asked to perform self-contained information seeking tasks within a unit of work (also known as HIT in AMT) advertised on the crowdsourcing platform. In the meantime, researchers can collect logs of workers' interactions with the IIR system as well as post-search information and statistics. Although this procedure might appear similar to laboratory based IIR experiments, a number of key factors affect important experimental aspects, thus effectively differentiating these two strategies. In particular, they differ in the way the following experimental aspects are tackled:

1. Characterise user population (section 3.1)
2. Define information seeking tasks (section 3.2)
3. Capture interactions (section 3.3)
4. Acquire post-retrieval information (section 3.4)

Before describing the experimental aspects that characterise the protocol based on crowdsourcing, we briefly outline some of the key factors that differentiate crowdsourcing-based from laboratory-based IIR experiments.

Heterogeneity. The user population that can be reached through crowdsourcing is highly heterogeneous with respect to location, nationality, education, employment, age, sex, language, etc (see [8] for a demographical study of the workers of AMT).

Cost. Crowdsourced IIR experiments are likely to be cheaper than laboratory-based ones: e.g. the average hourly rate of the experiments detailed in section 4 is $1.38, while the national minimum wage in UK is about $9.35.

Scale. Because researchers can access a large number of workers through crowdsourcing tools, and because of the associated low costs, crowdsourcing often provides the opportunity to reach a higher number of participants for IIR experiments than laboratory-based approaches.

Users' information quality. While it is often assumed that participants in laboratory-based experiments provide to researchers correct and detailed information about themselves[1], the same cannot be assumed for crowdsourced workers. In fact, usage regulations of web-based crowdsourcing platforms often forbid researchers to ask for personal details of users (e.g. see AMT policies[2]). Furthermore, it cannot be excluded that malicious users participate in crowdsourced tasks. Finally, crowdsourced workers likely optimise their working strategy for completing tasks, so as to achieve task completion with the minimum effort or within a minimum time.

Typology of IIR tasks. In section 2 we have pointed out that traditional IIR experimental paradigms prescribe the creation of simulated work task situations. This often requires participants to read instruction sheets that not only outline how to use the IIR system, but also describe the simulated situation the user has to imagine and the information need he is expected to satisfy. This procedure is unlikely to be suitable for crowdsourced workers, as previous studies noted that the instructions provided to workers have to be kept short and simple, and workers are unlikely to perform the cognitive effort required by simulated situations and information seeking tasks.

Quality of interactions/reliability of interactions. Previous studies suggested that crowdsourced workers tend to complete tasks as efficiently as possible [6]. Furthermore, others suggested that malicious workers might submit tasks without actually performing the requested operations. These aspects pose doubts on the quality and reliability of interactions captured through crowdsourcing. Interactions obtained via crowdsourcing should be validated and then compared against those acquired with traditional approaches.

[1] Researchers select a group of qualified subjects and ask their personal information.

[2] https://requester.mturk.com/mturk/help?helpPage=policies

3.1 Characterise User Population

Pre-experiment questionnaires and interviews are usually employed by researchers for acquiring demographical and self-perceptual information about participants in laboratory-based IIR experiments. This method is however inapplicable for crowdsourced IIR experiments. First, if workers are asked to fill in questionnaires[3] within a unit of work (i.e. HIT), then they will have to enter the same information several times: as many as the number of HITs they perform. This problem can be overcome by requiring workers to pass a *qualification test*. By employing qualification tests, researchers can acquire background information about the users to characterise the user population. Furthermore, experimenters can exclude from their HITs those workers that do not meet pre-defined criteria suitable for the experiment. Once workers are characterised through a qualification test, they can be classified within groups on the basis of similar scores. Groups can then be used to compare and contrast search behaviours and interactions of crowdsourced workers against the ones obtained by correspondent groups of laboratory based participants. This approach provides a means for comparing search behaviours and interactions between the two user populations.

However, crowdsourcing tools do not usually allow requesters to ask personal questions to users, such as their age, sex, etc. Moreover, it is yet unclear how to judge the truthfulness of answers related to self-perception questions, such as workers' confidence with search engines and search tasks, their expertise, etc [6]. Thus, qualification tests have to be carefully chosen in order (i) not to violate the crowdsourcing tool's policies, (ii) to avoid doubts on the truthfulness of the acquired data, (iii) but yet to obtain information that characterises users and their abilities. To address these points, we propose to use qualification tests based on aptitude or Intelligence Quotient (IQ) tests developed in Psychometrics [4]. A further use of this test is to assess whether workers are suitable for the typology of information seeking tasks that are used in the experiments (e.g. domain specific applications). The intuition is that these tests provide a measure of reasoning skills, language knowledge and problem solving skills of crowdsourced workers, as well as a measure of their attention when performing crowdsourced tasks. It is yet to be said whether high IQ scores correspond to higher abilities in solving IIR tasks: this has to be further investigated. However, we expect that there is not a predominant score (or range of scores) amongst the ones obtained by crowdsourced workers. Conversely, we expect that if the same tests were performed by participants of laboratory studies recruited amongst the student population of universities, the scores would be predominantly grouped within a high score range, mainly because of the level of education of the participants, and for the fact that participants have often been already screened by universities[4] according to IQ tests when beginning their university degrees.

3.2 Define Information Seeking Tasks

Information seeking tasks assigned to crowdsourced workers have to be clear and well defined, as no interaction is possible between workers and requesters. Workers are unlikely to perform the cognitive effort required by simulated situations and information seeking tasks, as workers' main goal is to complete tasks as efficiently and rapidly as possible. We suggest that in crowdsourced IIR environments, researchers should explicitly provide the topic that the search will be about, together with a number of specific informational questions the workers are expected to answer. For example, one of the topics contained in the experiments we report in section 4 is "Australian wines". With respect to this topic, workers are asked to answer the following questions[5]: "What winery produces Yellowtail?", "Where does Australia rank in exports of wine?", and "Name some of Australia's female winemakers". We argue that posing questions about a specific topic initiates in the workers the search requirements needed by the settings of IIR experiments. We thus posit that no simulated tasks are required. In fact the scenario in which the information seeking task is performed results clear: workers have to answer a number of questions, and to help themselves they can find information about these questions by searching through the provided IIR system. Topics and questions should be carefully chosen so that answers are not likely to be known, and search needs are thus effectively initiated.

3.3 Capture Interactions

Once topics and questions are assigned, workers can search with the provided IIR system in order to find useful information for formulating answers. It is imperative for the IIR system to capture the interactions between the workers and the system itself (e.g. issued queries, clicked results, time spent in reading/searching, etc). Crowdsourcing platforms, such as AMT, do not provide native tools for capturing these kind of user interactions. However, several solutions can be devised so as to direct the workers towards a tool that is controlled by experimenters, and thus records workers' interactions. For example, Field et al. used a proxy to achieve this goal [6]. In the experiments reported in section 4 a different solution was adopted: workers were shown the interface of the IIR system within a self-contained iFrame positioned in the page of the HIT. Through iFrames, interactions could be recorded, making them available for further analysis.

3.4 Acquire Post-Search Information

Self-perception information about the search task workers just performed can be acquired by means of a questionnaire within a unit of work. Questions can be related to the difficulty of the task, the level of satisfaction with both system and answers provided, etc. However, little can be said about the truthfulness of the acquired data [6]. Nevertheless, this problematic issue can be partially addressed by well known techniques, e.g. different phrasing of subsequent questions, so that answers cannot be inferred by the context.

4. EXPERIMENTING WITH THE NEW IIR PROTOCOL

For the purpose of setting up a preliminary investigation of the novel protocol for IIR experiments introduced in section 3, we asked AMT's workers to carry out 24 search tasks[6] extracted from the TREC 2006 and 2007 Question-Answering track[7]. For each topic, three questions were se-

[3] We ignore the possibility of performing interviews of workers, given the remote and asymmetric nature of crowdsourcing.

[4] At least in many European countries.

[5] Of course, making use of a search engine we provide for helping them find information useful for answering the questions.

[6] Each task was repeated by three different workers.

[7] `http://trec.nist.gov/data/qamain.html`

lected. Twenty-three workers performed our HITs (a worker completed on average 3.13 HITs). Workers were divided into two groups: the first needed to pass a 20-questions aptitude qualification test, while the second did not. Workers completed units of work by answering the posed questions using the provided IIR system to assist them in finding the answers on the Web. They were also asked to mark Web pages containing information useful for answering the questions. After the questions were answered, workers were asked to evaluate various aspects of their search experience in a post-search questionnaire: 4 five-point semantic differential questions focused on the performed task, while 3 five-point Likert scales assessed their background knowledge of the task and the search they performed.

4.1 System Platform

We embedded our IIR system within the crowdsourcing platform offered by AMT, using an iFrame within a standard HIT. Our IIR system was developed as a web-based front-end of the Microsoft Bing API[8] for web results. Each time a user began a search task our system was provided with AMT HIT details such as the work assignment ID and the corresponding question topic. Queries, result clicks and explicit feedback via an optional "Mark as Relevant" button were logged alongside the HIT information. Following completion of the batch of HITs for each experiment we then merged the provided search logs with the AMT logs to yield a rich source of individual worker data for analysis. AMT data provided statistics such as the search task duration, question answers, unique worker IDs and qualification scores that can be used to begin explaining behaviours observed through the related query logs.

4.2 Preliminary Results

In table 1 we outline preliminary interaction statistics that were acquired through the experiments performed with the novel protocol for IIR based on crowdsourcing. A definitive statement deriving from the analysis of the reported data cannot be made yet, since at the moment we have not performed a laboratory-based counterpart of the experiment. However, the statistics show how workers had to interact with the search system in order to find information that helped them formulating answers to the provided questions. Moreover, feedback from workers show that the tasks we developed based on the TREC Question-Answering track were clear, slightly difficult, moderately complex, but familiar. Workers also stated that they did not know the answers before performing the HITs. In addition, they felt they successfully answered the questions.

4.3 Open Issues and Future Work

A number of issues have still to be investigated in order to assess the validity of the protocol we outlined in this paper:

1. Are the interactions acquired through crowdsourcing similar to those acquired through laboratory experiments? And, how can they be compared?
2. Is a training session required for crowdsourcing based experiments, as suggested by Borlund [3]?
3. Is it legitimate to use an aptitude test (IQ) to characterise and compare users in IIR settings?
4. What is the role of a crowdsourcing-based experiment in IIR evaluation? Can this be used to replace or complement laboratory-based experiments for qualitative and quantitative assessment?

	MIN	AVG	MAX	STD
Search Queries	1.00	6.54	11.00	2.92
Total Unique Viewed Pg.	0.00	2.20	9.00	1.76
Unique Viewed Pg. from Wiki	0.00	0.45	3.00	0.73
Unique Viewed Pg. from Non-Wiki	0.00	1.75	7.00	1.64
Total Unique Rel. Pg.	0.00	0.73	4.00	1.17
Unique Rel. Pg. from Wiki	0.00	0.09	1.00	0.29
Unique Rel. Pg. from Non-Wiki	0.00	0.64	4.00	1.12
Time Spent in Seconds	36.00	459.86	776.00	195.68

Table 1: Statistics of user interactions on 24 HITs.

5. PROSPECTIVES FOR IIR

In this paper we have proposed a new strategy based on crowdsourcing for acquiring interactions between users and IIR systems. The acquisition of interaction data via crowdsourcing is not intended to act as a substitute in laboratory-based experiments, but complements it by offering additional data to analyse.

Moreover, if the validity of the proposed experimental protocol is confirmed by further studies, this work opens up a number of novel research prospectives for IIR. In fact, interaction data can be acquired following the proposed crowdsourcing protocol as to study querying behaviours, search strategies, and, ultimately, for comparing, contrasting and evaluating interactive IR systems.

Future work will be directed towards the consolidation and evaluation of the introduced crowdsourcing protocol for IIR, in particular by comparing the acquired information against that obtained through laboratory based experiments. Furthermore, we intend to explore the possibility of applying the protocol to the evaluation of IIR systems.

6. REFERENCES

[1] O. Alonso and S. Mizzaro. Can we get rid of TREC assessors? Using Mechanical Turk for relevance assessment. In *SIGIR 2009 Work. on The Future of IR Eval.*, 2009.

[2] O. Alonso, D. E. Rose, and B. Stewart. Crowdsourcing for relevance evaluation. *SIGIR Forum*, 42:9–15, 2008.

[3] P. Borlund. The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research*, 8(3), 2003.

[4] P. J. Carter. *IQ and Psychometric Tests.* Kogan Page, 2007.

[5] W. B. Croft, M. Bendersky, H. Li, and G. Xu. Query representation and understanding workshop. *SIGIR Forum*, 44(2):48–53, 2010.

[6] H. Feild, R. Jones, R. C. Miller, R. Nayak, E. F. Churchill, and E. Velipasaoglu. Logging the Search Self-Efficacy of Amazon Mechanical Turkers. In *SIGIR 2009 Work. on Crowdsourcing for Search Eval.*, 2009.

[7] J. Howe. The Rise of Crowdsourcing. (accessed July 20, 2010), June 2006.

[8] J. Ross, A. Zaldivar, L. Irani, B. Tomlinson, and M. S. Silberman. Who are the crowdworkers? shifting demographics in mechanical turk. In *Proceedings CHI 2010*, pages 2863–2872, 2010.

[9] R. W. White. *Implicit Feedback for Interactive Information Retrieval.* PhD thesis, University of Glasgow, 2004.

[8] `http://www.bing.com/developers`