

# SIGIR 2010

Geneva, Switzerland  
July 19-23, 2010

## Crowdsourcing for Search Evaluation

Workshop of the 33<sup>rd</sup> Annual International  
**ACM SIGIR Conference**  
*on Research and Development  
in Information Retrieval*

**Organised by**  
Vitor Carvalho, *Microsoft Bing*  
Matthew Lease, *University of Texas at Austin*  
Emine Yilmaz, *Microsoft Research*



Association for  
Computing Machinery



**SIGIR**  
**Geneva 2010**

## **Crowdsourcing for Search Evaluation (CSE 2010)**

A Workshop at the 33rd Annual International ACM SIGIR Conference on  
Research and Development in Information Retrieval

Geneva, Switzerland – July 23, 2010

### **Organized by**

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

### **Program Committee**

Eugene Agichtein, Emory University  
Ben Carterette, University of Delaware  
Charlie Clarke, University of Waterloo  
Gareth Jones, Dublin City University  
Michael Kaisser, University of Edinburgh  
Jaap Kamps, University of Amsterdam  
Gabriella Kazai, Microsoft Research  
Mounia Lalmas, University of Glasgow  
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, University of Sheffield  
Mark Smucker, University of Waterloo  
Siddharth Suri, Yahoo! Research  
Fang Xu, Saarland University

# Crowdsourcing for Search Evaluation (CSE 2010)

## Workshop Proceedings

<i>Crowdsourcing for Affective Annotation of Video: Development of a Viewer-reported Boredom Corpus.</i> Mohammad Soleymani and Martha Larson. ....	4
<i>Crowdsourcing Preference Judgments for Evaluation of Music Similarity Tasks.</i> Julian Urbano, Jorge Morato, Monica Marrero and Diego Martin. ....	9
<i>Ensuring quality in crowdsourced search relevance evaluation.</i> John Le, Andy Edmonds, Vaughn Hester and Lukas Biewald. ....	17
<i>An Analysis of Assessor Behavior in Crowdsourced Preference Judgments.</i> Dongqing Zhu and Ben Carterette. ....	21
<i>Logging the Search Self-Efficacy of Amazon Mechanical Turkers.</i> ..... Henry Feild, Rosie Jones, Robert C. Miller, Rajeev Nayak, Elizabeth F. Churchill and Emre Velipasaoglu	27
<i>Crowdsourcing a News Query Classification Dataset.</i> Richard M. C. McCreddie, Craig Macdonald and Iadh Ounis. ....	31
<i>Detecting Uninteresting Content in Text Streams.</i> Omar Alonso, Chad Carson, David Gerster, Xiang Ji, Shubha U. Nabar. ....	39

# Crowdsourcing for Affective Annotation of Video: Development of a Viewer-reported Boredom Corpus

Mohammad Soleymani  
Computer Vision and Multimedia Laboratory  
University of Geneva  
mohammad.soleymani@unige.ch

Martha Larson  
Multimedia Information Retrieval Lab  
Delft University of Technology  
m.a.larson@tudelft.nl

## ABSTRACT

Predictions of viewer affective response to video are an important source of information that can be used to enhance the performance of multimedia retrieval and recommendation systems. The development of algorithms for robust prediction of viewer affective response requires corpora accompanied by appropriate ground truth. We report on the development a new corpus to be used to evaluate algorithms for prediction of viewer-reported boredom. We make use of crowdsourcing in order to address two shortcomings of previous affective video corpora: small number of annotators and gap between annotators and target viewer group. We describe the design of the Mechanical Turk setup that we used to generate the affective annotations for the corpus. We discuss specific issues that arose and how we resolve them and then present an analysis of the annotations collected. The paper closes with a list of recommended practices for the collection of self-reported affective annotations using crowdsourcing techniques and an outlook on future work.

## Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing

## General Terms

Algorithms, Measurement, Design, Experimentation,

## Keywords

Affective computing, multimedia benchmarking, internet video

## 1. INTRODUCTION

Developing video processing algorithms capable of predicting viewer boredom requires suitable corpora for development and testing. This paper reports on the development of the MediaEval 2010 Affect Task Corpus for boredom prediction of Internet video. Standard limitations on viewer affective response annotation are overcome by making use of crowdsourcing. Using Mechanical Turk (MTurk), we rapidly gather self-reported boredom scores from a large user group that is demographically diverse and also represents our target population (Internet video viewers).

Ultimately, our boredom-prediction algorithms will be used to improve multimedia retrieval and recommendation. Relatively little research has investigated topic-independent factors that contribute to the relevance of multimedia content to the user information need. In the area of text-based retrieval, incorporation of quality information has been used to improve results, as, for example in [14]. Our larger goal is to extend such techniques to multimedia information retrieval and recommendation.

Copyright is held by the author/owner(s).  
Workshop on Crowdsourcing for Search Evaluation  
SIGIR'10, July 19–23, 2010, Geneva, Switzerland.

We focus on viewer affective response, and in particular on boredom, as a reflection of perceived video quality. We are also interested in variation of affective response among viewers that will help us to develop recommendation and retrieval systems that incorporate information on personal preference.

Our starting point is a set of specifications that our corpus was required to fulfill. The annotation process needed to control as much as possible for extraneous effects, such as reaction of the annotators to the topic of the video, tiredness or underlying mood of the annotators. We wanted to have a relatively large number of annotators for each video, but also a certain number of annotators who annotated the whole collection. We wanted to avoid violating copyright law in order to be able to license our corpus for public use the MediaEval 2010 benchmark. Finally, we had limited resources to invest in corpus development. After a short section on to related work, this paper describes the MediaEval 2010 Affect Task and then the MTurk task that was used to annotate the Affect Task Corpus. We discuss how we fulfilled the specifications of the corpus and met other challenges arising along the way. Finally, we present an analysis of the collected annotations and we distill our experience into a list of recommendations for using crowdsourcing for viewer affective response annotation.

## 2. RELATED WORK

There are two notable efforts by psychologists to create standard affective video corpora for emotional studies [8][6]. In both studies, movie excerpts extracted from Hollywood movies were used. Because only the time codes of the excerpts and their description are published, the datasets are difficult to re-use. Moreover, use of copyrighted video material depends on the regulations of individual countries. In general, it cannot be shared between researchers or shown to the public for purposes of conducting experiments, gathering annotations or demonstrating systems.

The research in the field of multimedia content analysis for affective understanding of videos lacks significant user studies and only relies on the feedback from limited number of participants [5][10][13]. Multimedia corpora with affective annotations make it possible to investigate interesting research questions and develop useful algorithms, but are time-consuming to generate. The number of participants contributing annotations is a significant factor that limits their usefulness. We describe the 2009 Affect Task in the VideoCLEF (now called MediaEval) benchmark [5] as an example of such a case. The 2009 Affect Task involved narrative peak detection – automatic identification of points within a video at which users experience a heightened sense of dramatic tension. Narrative peak detection is related to highlights detection in sports videos cf. [2], but cannot rely on the presence of audience reaction (the roar of the crowd) in the video. The 2009 Affect Task corpus contains 45 eight-minute videos that are

documentaries on the visual arts, hosted by a well-known Amsterdam professor, Henk van Os. Three assessors watched each video in its entirety and marked the start and end points of the segments that they identified to be the top three narrative peaks. The annotation is necessary time-consuming. In order to understand peaks against the background of their narrative context, it is necessary to watch the video as a whole. Generally, annotating the videos took 2-3 times the run-time of the video. In the 45 videos, there were only 22 peaks that all three assessors identified as among the top three. Although the agreement might have been higher, had we examined a longer top-N list, the annotations generated strongly suggest that there is a personal component determining where viewers perceive narrative peaks. In order to gain a deeper understanding of this component it would be necessary to have more than 3 assessors watch the entire video set. Moreover, assessors reported a familiarity effect. Their sensitivity to narrative peaks developed the more of Prof. van Os' material that they watched. The familiarity effect seemed to be related to a better understanding and appreciate the narrator's style, e.g., sense of humor. More annotations are necessary in order to understand better how affective response changes or develops with familiarity.

To our knowledge there has been only one effort to gather online affective annotations with a large set of participants [11]. During that study more than 1300 annotations from 40 volunteer participants were gathered for 155 video excerpts extracted from Hollywood movies. Although the number of participants is among the largest population size in its kind, the dataset is not redistributable due to the copyright violations issues. The participants who usually volunteer to participate in academic studies are from a certain age group and limited geographical locations or cultural background. In the current dataset, both copyright problems and population size and diversity are addressed.

### 3. AFFECT PREDICTION TASK

The MediaEval 2010 Affect Task involves automatically predicting the level of user boredom for a video. The Affect Task is running in 2010 within the MediaEval benchmarking initiative [5], which offers tasks to the multimedia research community that help consolidate and synchronize research effort and concentrate it on forward-looking, challenging research areas. Research groups build systems that predict affect and test them on the Affect Task Corpus. For the purpose of the Affect Task and related research, we adopt a fairly simply definition of boredom. We take boredom to be related to the viewer's sense of keeping focus of attention and to be related to the apparent passage of time [4]. We understand boredom to be a negative feeling associated with viewer perceptions of the viewer-perceived quality (viewer appeal) of the video being low.

We are interested in studying two aspects of viewer-reported boredom. First, the 2010 Affect Task corpus will be used to investigate universal aspects of user boredom. On the Internet, certain videos emerge as being more popular than others (as reflected by views, links or viewer-contributed ratings). This popularity can be taken as a reflection of an underlying consensus of an inherent quality of the video, i.e., in some sense it is "worth watching." If this quality is at least in part related to the video content, then we believe that it is worthwhile investigating the extent to which it can be predicted using automatic methods. We know that Internet videos differ not only in subject material, but also with respect to other factors. Among the factors that influence the creation process of a video are: novelty of videographic style, resources avail-

able, production skill of the film maker and amount of care invested in planning and realization.

Second, the corpus will be used to investigate personal variation. Affective reaction to video content differs widely from viewer to viewer. We are interested in determining if it is possible to build user-specific models for prediction of self-reported boredom. Additionally, we would like to investigate whether affective reaction can be modeled at a level between the universal and the personal. In other words, we would like to determine whether predictive models can be built for certain groups of users.

The dataset selected for the corpus is Bill's Travel Project, a travelogue series called "My Name is Bill" created by the film maker Bill Bowles (<http://www.mynameisbill.com/>). The series consists of 126 videos between two to five minutes in length. This data was chosen since it represents the sort of multimedia content that has risen to prominence on the Internet. Bill's travelogue follows the format of a daily episode related to his activities and as such is comparable to "video journals" that are created by many video bloggers. We believe that results of investigations on Bill's Travel Project will extend to other video bloggers, and also perhaps to other sorts of semi-professional user generated video content. Because we are interested in aspects of the data that are independent of topic and genre, we were careful to choose data related to the same topic (travel) and genre (video blog). Further, the fact the video predominantly involves only a single speaker (Bill) helps to abstract away from personal preferences of the viewer that might be based on the gender or appearance of the central figure(s) rather than on the content of the video. The focus is kept squarely on pacing, narrative devices and manner of presentation. Finally, since the video is not Creative Commons licensed we contacted Bill, who kindly granted us permission to use it for the Affect Task. In this way, we were able to develop the corpus without concerns about copyright violations.

The relationship of the 2010 boredom prediction task to the 2009 narrative peak predication task also requires a note of explanation. We would like to investigate if there is a relationship between affective reactions within the video (i.e., their magnitude and timing) to the overall appeal of the video for users. As a result of the experience with the creation of the 2009 corpus, in 2010, we will be investigating possible "familiarity" effects in viewer-reported affective response. In other words, we are interested in whether there is a trend in viewer's reactions to Bill's videos as they grow more acquainted with his material. Specifically, we would like to know whether viewers report increasing boredom as they watch more of Bill's material or whether we find evidence a "fan of Bill" effect, namely, that they report less boredom with growing familiarity with Bill, his journey and his personal style.

The participants carrying out the Affect Task in MediaEval 2010 are various international research groups involved in multimedia information retrieval and affective computing research. The groups are free to design their own algorithms for automatic boredom detection and can make use of features derived from the visual channel, audio channel or speech recognition transcripts. Speech recognition transcripts were supplied with the corpus and generously donated to the benchmark by ICSI and SRI International [12]. Groups approach the tasks in multiple ways. Generally, they first formulate an idea of what properties of the video contribute to user perceived boredom and then build a model that captures these properties. In a typical model, the focus is on properties of the video related to production, for example the cutting or audio mixing, but they also include a wide range of factors.

We were also able to formulate theories about the sources of possible viewer interest in the video by interviewing Bill Bowles concerning the strategies that he makes use of as a film maker to add interest to his videos. In particular he mentioned, that he keeps shots short (< 1 minute), he varies the rhythm of the shot length, he doesn't make the videos any longer than necessary and he varies between close ups and distance shots. Finally, and perhaps presenting the biggest challenge to capture in an automatic algorithm, he attempts to continuously surprise his viewer with a novel approach to his subject material. For example, he switches his role (e.g., between observer, interviewer and commentator) and uses word play and comic devices. Bill also mentioned that how he makes the video is affected by his own mood at the time. This point is not relevant for the Affect Task, which deals with viewer-reported mood, but is an interesting vista for future work.

#### 4. DESIGN OF CROWDSOURCING TASK

We approached the design of the MTurk task by first reading crowdsourcing literature, for example [3], searching for information the Internet on the subject of using MTurk and reflecting on our past experience collecting annotations online. We decided for a two-step approach. The first step was the pilot that consisting of a single micro-task (HIT) involving one video would be used for the purposes of recruiting and screening MTurk users (referred to as "workers"). The second step was the *main task* and involved a series of 125 micro-tasks, one for each of the remaining videos in the collection. We discuss each step in turn.

##### 4.1 Pilot

The pilot contained three components corresponding to qualities that we required of our recruits. The first section contained questions about the personal background (age, gender, cultural background). We made judicious use of MTurk's ability to block workers from certain countries in order to maintain the overall balance. The second section contained questions about viewing habits: we asked the workers if they were regular viewers of Internet video. The third section tested their seriousness by asking them to watch the video, select a word that reflected their mood at the moment and also write a summary. The summary constituted a "verifiable" question, recommended by [3]. The summary offered several possibilities for verification. Its length and whether it contained well-formulated sentences gave us an indication of the level of care that the worker devoted to the HIT. Also, the descriptive content indicated to us whether the worker had watched the entire video, or merely the beginning. We also checked seriousness by ensuring that workers did not complete the HIT faster than the run-time length time of the video. A final question enquired if they were interested in further HITs of the same sort. We were interested in deflecting the attention of the worker away from the main goal of the task, i.e., collecting affective annotations. For this reason we placed the summary box prominently in the HIT. We also believe it was an effective distracter since it was the element of the HIT that was the longest and most intellectually challenging to answer.

##### 4.2 Main Task

We chose the workers for the main task from the participants of the pilot by considering the quality of their description and choosing a diverse group of respondents. The qualification was only granted to the participants who answered all the questions completely. We invited workers to do the main study by sending them an invitation e-mail invitation via their ID number on the MTurk platform. The e-mail informed the users that we had assigned

them our MTurk qualification. Use of a qualification serves to limit those workers that carry out the HIT to invitees only. Each HIT in the main study consisted of three parts. In the first part, the workers were asked to specify the time of day, which gave us a rough estimate of how tired they were. Also the workers were asked to choose a mood word from a drop down list that best expressed their reaction to an imaginary word, such as those used in [7]. The mood words were *pleased, helpless, energetic, nervous, passive, relaxed, and aggressive*. These questions gave us an estimate of their underlying mood. In the second part, they were asked to watch the video and give some simple responses. They were asked to choose the word that best represented the emotion they felt while watching a video from a second list of emotion words in the drop down list. The emotion list contained Ekman six basic emotions [1], namely, *sadness, joy, anger, fear, surprise, and disgust*, in addition to *boredom, anxiety, neutral and amusement*, which cover the entire affective space, as defined by the conventional dimensions of valence and arousal [9]. The emotion and mood word lists contained different items, which were intended to disassociate them for the user. Next, they were asked to provide a rating specifying how boring they found the video and how much they liked the video, both on a nine point scale. Then, they were asked to estimate how long the video lasted. Here, we had to rely on their full cooperation in order not to cheat and look at the video timeline. Finally, they were asked to describe the contents of the video in one sentence. We emphasized the description of the video rather than the mood word or the rating, in order to conceal the main purpose of the HIT. Quality control of the responses was carried out by checking the description of the video and also by ensuring that the time that they took to complete the HIT was reasonable.

##### 4.3 Issues and solutions

The most important issue with the MTurk task arose because we needed each worker to finish all 125 videos. In the invitation to the main task we named the total sum workers would earn by completing all 125 HITs as an enticement, but we also mentioned that we would only accept the HITs if they completed all 125. Approximately half of the workers we invited to do the task responded positively to this arrangement. Many wrote personal e-mails with specific questions or asking for assurances from our side that we would accept their HITs. The personal communication with the workers was a key factor in collecting the annotations. We were surprised at workers' willingness to give up their anonymity by writing us e-mails and also revealing to us their worker IDs. Many also mentioned their base location in their e-mails. This evolving openness gave us more confidence in trusting the original demographic information collected in the pilot, since by revealing their identities the workers showed themselves willing to provide us with the opportunity to verify at least some of the personal information provided in the pilot. We noticed that many workers were not willing to make the commitment to do all 125 HITs. Building trust was very important. It quickly became clear that some workers were reluctant to risk starting on the series out of fear that we would reject their hits and ruin their reputations on MTurk. Receiving the payment seemed to be secondary. We noticed that at least one person really appreciated that completing the whole series gave them a substantial goal to work for and that the sum that they earned could then be used to buy a particular book. Personal communication via e-mail was essential when the video server that we were using developed a technical problem and the videos did not load. We fielded many e-mails on

those days, and on the whole were surprised at the patience that and cooperative spirit of the workers in the face of the problems.

## 5. ANALYSIS OF ANNOTATIONS

Our pilot HIT was initially published for 100 workers and finished in the course of a single weekend. We re-published the HIT for more workers when we realized we needed more people in order to have an adequate number of task participants. Only workers with the HIT acceptance rate of 95% or higher were admitted to participate in the pilot HIT. In total, 169 workers completed our pilot HIT, 87.6% of which reported that they watch videos on the Internet. We took this response as confirmation that our tasks participants were close to the target audience of our research. Out of 169 workers, 105 were male and 62 were female and two did not report their gender. Their age average was 30.48 with the standard deviation of 12.39. The workers in the pilot HITs identified themselves by different cultural backgrounds from North American, Caucasian to South and East Asian. Having such a group of participants with a high diversity in their cultural background would have been difficult without using the crowd-sourcing platforms. Of the 169 pilot participants, 162 had interest in carrying out similar HITs. Of the interested group, the 79 workers were determined to be qualified and assigned our task-specific qualification within MTurk. This means only 46.7% of the workers who did the pilot HIT were able to answer all the questions and had the profile we required for the main task.

In total, 32 workers have participated and also annotated more than 60 of the 125 videos in the main task HIT series. This means only 18.9% of the participants in the pilot and 39.0% of the qualified participants committed to do the main task HIT series seriously. Of this group of 32 serious participants, 18 are male and 11 are female with ages ranging from 18 to 81 (average 34.9; standard deviation 14.7).

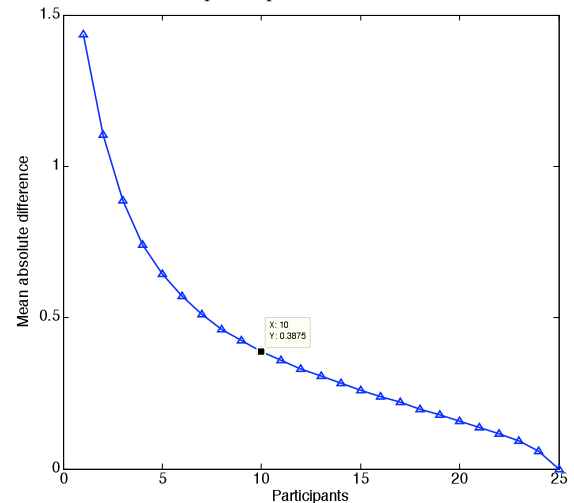
To evaluate the quality of the annotations, the time spent for each HIT was compared to the video length. In 81.8% of the completed HITs the working duration for each HIT was longer than the video length. This means that in 18.2% of the HITs the workers did not follow the instructions. Also, their reported perception of the time is invalid. This shows the importance of having workers with the right qualifications and trustworthy pool of workers in annotation or evaluation hits. Even after the pilot task and disqualifying 60% of the first participants, 16 participants or 39.0% of our final pool did not watch at least 10% of their submitted HITs' videos completely. Rejecting those HITs reduced the number of workers who carried out more than 60 videos in the main series of HIT to 25 from which 17 are male and 8 are female ages ranging from 19 to 59 (average 33.9, standard deviation 11.8).

Three questions were asked about each video to assess the level of boredom. First, how boring the video was on nine-point scale from the most to the least boring. Second, how much the user liked the video on the nine-point scale and third how long the video was. Boredom was shown to have on average a strong negative correlation,  $\rho = -0.86$  with liking scores. The time perception did not show a significant correlation for all users and it varied from 0.4 down to -0.27. Although positive correlation was expected from boredom scores and the perception of time seven participants' boredom scores have negative correlation with the time perception.

The correlation between the order of watching the videos for each participant and the boredom ratings was also examined. No positive linear correlation was found between the order and boredom

score. This means that watching more videos did not increase the level of boredom and in contrary for 2 of participant it decreased their boredom. Additionally, the correlation between the video length and boredom scores was investigated. No positive correlation was found between the boredom scores and videos' duration. We can conclude that the lengthy videos are not necessarily perceived as more boring than the shorter videos.

To measure the inter-annotator agreement, the Spearman correlation between participants' pairwise boredom scores was computed. The average significant correlation coefficient was very low  $\rho = 0.05$ . There were even cases where the correlation coefficients were negative, which shows complete disagreement between participants. For each worker we then grouped videos into two rough categories, above and below the mean boredom score of that worker. We computed the average pair-wise Cohen's kappa for these categories and here found only slight agreement ( $\kappa = 0.01$ ). We also compared agreement on the emotion words workers associated with viewers. Here, again Cohen's kappa indicated only slight agreement ( $\kappa = 0.07$ ). The strong correlations suggest that it is indeed important to investigate personalized approaches to affective response prediction.



**Figure 1** The mean absolute difference (on the vertical axis) versus number of participants.

One of the key questions in such studies is the number for participants for a significant result. In order to address this question, the situation of having fewer participants was simulated and the mean absolute difference with the final average was computed (see Figure 1). In this simulation, participants were randomly drawn and added to the pool of participants with the pool size of one to the maximum possible size of 25. This random simulation was performed 1000 times and the mean absolute difference between the participants' average annotations and the average scores of all 25 participants were computed. As it can be seen in Figure 1, with 10 participants the difference between the averaged scores is smaller than 5% of the possible range,  $0.05 \times 8 = 0.4$ . Although the gain of having more participants gets smaller after 10, in the real world applications a larger pool of annotators is always a valuable asset for information retrieval and recommendation studies.

## 6. BEST PRACTICES

Crowdsourcing using MTurk provided an effective means of collecting the viewer affective response annotations needed to create a corpus to be used in the development of automatic prediction of

viewer reported boredom. Our experience can be distilled into a list of recommendations that will enable the development of additional such corpora to proceed smoothly.

- The MTurk task should consist of two steps, the first identifies appropriate workers to invite and the second involves the generation of the annotations.
- For a long HIT series tasks, invite five times as many workers to do the pilot as you wish to have complete the main task.
- Expect that up to 75% of the workers you will invite will not be interested in carrying out a HIT that has the feeling of traditional "work", i.e., requires a long time commitment. In the invitation e-mail, specify a date by which they need to reply so that you can disinvite them and invite others if necessary.
- Consider breaking down long HIT series into packages and giving a small reward to the completion of individual packages in addition to a larger bonus for completing the whole series in order to prevent fatigue of the workers.
- As suggested by [3], we use multiple methods to verify that the workers are doing a good job on the question, for example, as a verifiable question and also check time.
- Include dummy questions to veil the purpose of the HIT.

Establishing trust with workers is a key factor in getting the same users to do a long HIT series. It is important to remember that they are concerned about maintaining their reputation on MTurk. Trust can be built by accepting HITs as quickly as possible and also being prompt with the bonuses. We suggest making the payment for each HIT very small and then accepting the HIT relatively indiscriminately. Workers who complete the entire series and do it well then receive the bonus. Effort invested in establishing trust accumulates since users exchange information on requesters on Turker Nation (<http://www.turkernation.com/>) concerning the HITs and the bonuses rewarded.

Our future work will concentrate on scaling up to be able to collect annotations for a larger set of videos with less intervention on our part. We now realize that for long HIT series, such as the ones necessary for a single person to annotate many videos, MTurk does not "run by itself", but rather requires constant attention in terms of contacting workers and answering e-mail. In the future, we plan to be highly active during the initial stage of our main task to help speed up the process. In the future, we would like to develop a more complex pilot HIT that provides a more effective recruitment tool for workers. We are considering including more videos in the pilot HIT, or implementing a two-stage pilot, involving two HITs. A key factor here might be to use the MTurk API more extensively to achieve a higher level of automation. Addressing a practical problem, we would also like to work on developing a mechanism to deal elegantly with the failure of external resources. If a video fails to load, then the HIT is lost for the worker and needs to be manually reinitiated. The speed of the response depends on the amount of the reward offered. We paid viewers US \$37.50 for watching 125 short videos. Paying less might have been possible. It would be worthwhile to determine if we can offer lower rewards without compromising quality. We also would like to investigate the bias introduced into the system by the fact that a certain type of personality is attracted to MTurk tasks and in particular to our Affect Task. Finally, we would like to move from boredom detection to other affective annotations. Our experiences with the MediaEval 2010 Affect Task Corpus suggest that crowdsourcing is a valuable technique to collect affective annotations and we have just begun to tap its potential.

## 7. ACKNOWLEDGMENTS

The research leading to these results has received funding from the EC FP7 under grant agreement n° 216444 (PetaMedia Network of Excellence). The work of the first author is supported in part by the Swiss National Science Foundation.

## 8. REFERENCES

- [1] P. Ekman et al., Universals and cultural differences in the judgments of facial expressions of emotion, *Journal of Personality and Social Psychology*, 53(4):712-717, 1987.
- [2] A. Hanjalic and L.-Q. Xu. A selective video content representation and modeling. *IEEE Transactions on Multimedia*, 7(1):143-154, 2005.
- [3] A. Kittur, E.H. Chi, and B. Suh. Crowdsourcing user studies with Mechanical Turk. In *Proceedings of the 26<sup>th</sup> Annual ACM conference on Human Factors in Computing Systems (CHI'08)*, 453-456, 2008.
- [4] J.D. Laird. *Feelings: The Perception of Self*. Oxford University Press, 2007.
- [5] M. Larson, E. Newman, G.J.F. and Jones, G. J. F. Overview of Video-CLEF 2009: New perspectives on speech-based multimedia content enrichment. In C.Peters et al. (eds.), *Multilingual Information Access Evaluation, Vol. II Multimedia Experiments*, Springer, to appear 2010.
- [6] P. Philippot. Inducing and assessing differentiated emotion-feeling states in the laboratory. *Cognition & Emotion* 7(2):171-193, 1993.
- [7] M. Quirin, M. Kazen, and J. Kuhl, J. When nonsense sounds happy or helpless: The Implicit Positive and Negative Affect Test (IPANAT). *Journal of Personality and Social Psychology*, 97:500-516, 2009.
- [8] J. Rottenberg, R.D.Ray and J.J.Gross. Emotion elicitation using films. In A.Coan and J.J.B.Allen (eds.), *The Handbook of Emotion Elicitation and Assessment*. Oxford University Press, 2007.
- [9] J. Russell and A. Mehrabian. Evidence for a 3-factor theory of emotions, *Journal of Research in Personality*, 11(3):273-294, 1977.
- [10] M. Soleymani, G. Chanel, J.J.M. Kierkels, and T. Pun. Affective characterization of movie scenes based on content analysis and physiological changes. *International Journal of Semantic Computing*, 3(2):235-254, 2009.
- [11] M. Soleymani, J. Davis, and T. Pun. A collaborative personalized affective video retrieval system. In *Proceedings of the International Conference on Affective Computing & Intelligent Interaction (ACII 2009)*, 588-589.
- [12] A. Stolcke, X. Anguera, K. Boakye, Ö Çetin, A. Janin, M. Magimai-Doss, C. Wooters, and J. Zheng. In Steifelhagen et al. (eds.) *The SRI-ICSI Spring 2007 Meeting and Lecture Recognition System*. LNCS vol. 4625, Springer, 450-463, 2008.
- [13] H.L. Wang and L.F. Cheong Affective understanding in film, *IEEE Transactions on Circuits and Systems for Video Technology*, 16(6):689-704, 2006.
- [14] X. Zhu and S. Gauch. Incorporating quality metrics in centralized/distributed information retrieval on the World Wide Web. In *Proceedings of the 23<sup>rd</sup> Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '00)*, 288-295, 2000.



# Crowdsourcing Preference Judgments for Evaluation of Music Similarity Tasks

Julián Urbano, Jorge Morato, Mónica Marrero and Diego Martín

University Carlos III of Madrid  
Department of Computer Science  
Leganés, Madrid, Spain

{jurbano, jmorato, mmarrero, dmandres}@inf.uc3m.es

## ABSTRACT

Music similarity tasks, where musical pieces similar to a query should be retrieved, are quite troublesome to evaluate. Ground truths based on partially ordered lists were developed to cope with problems regarding relevance judgment, but they require such man-power to generate that the official MIREX evaluations had to turn over more affordable alternatives. However, in house evaluations keep using these partially ordered lists because they are still more suitable for similarity tasks. In this paper we propose a cheaper alternative to generate these lists by using crowdsourcing to gather music preference judgments. We show that our method produces lists very similar to the original ones, while dealing with some defects of the original methodology. With this study, we show that crowdsourcing is a perfectly viable alternative to evaluate music systems without the need for experts.

## Categories and Subject Descriptors

H.5.1 [Multimedia Information Systems]: Evaluation/methodology; H.3.3 [Information Search and Retrieval]; H.3.4 [Systems and Software]: Performance evaluation (efficiency and effectiveness).

## Keywords

Crowdsourcing, relevance judgment, music information retrieval.

## 1. INTRODUCTION

Evaluation experiments are the corner stone of Information Retrieval (IR), as they are the main research tool for scientifically comparing retrieval techniques and figuring out which improve the state-of-the-art and which do not [1]. These evaluations have traditionally followed the so called Cranfield paradigm, where the set of relevance judgments are the most important and most expensive part of test collections. Usually, these ground truths take the form of a matrix containing information, assessed by humans, about the relevance of each document for each information need.

Music Information Retrieval (MIR) is a relatively young discipline, and this kind of evaluations has been somewhat scarce until the arrival of the Music Information Retrieval Evaluation eXchange (MIREX) in 2005, as a first attempt to perform TREC-like evaluations in the musical domain [2]. Evaluation in Music IR differs greatly from evaluation in Text IR, mainly with regard to the construction and maintenance of test collections [3]. On the one hand, MIR has been traditionally biased toward classical music because of many issues concerning copyright laws and royalties. On the other hand, many retrieval tasks defined for the music domain are inherently more complex to evaluate. This is

the case of the Symbolic Melodic Similarity (SMS) and Audio Music Similarity (AMS) tasks, as defined in MIREX, in which systems are asked to retrieve a ranked list of musical pieces deemed similar to some piece of music acting as query. In particular, it is unclear how to assess the relevance of a document for a given query.

Ground truths are traditionally based on a fixed scale of relevance with levels such as “relevant” and “not relevant”. However, several studies indicate that relevance is continuous for information needs involving music similarity [4][5][6]. Single melodic changes such as moving a note up or down in pitch, or extending or shortening its duration, are not perceived to change the overall melody. However, the relationship with the original melody is gradually weaker as more changes are applied to it. There are no common criteria to split the degree of relevance into different levels, so assessments based on a fixed scale do not seem suitable as it would be difficult to draw the line between levels.

Major advancements in this matter were achieved by Typke et al. by the beginning of 2005. They developed a methodology to create ground truths where the relevance of a document does not belong to any prefixed scale, but it is rather implied by its relative position in a partially ordered list [5]. These lists have ordered groups of candidates assumed to be equally relevant to the query, so that the earlier a group appears in the list, the more relevant its documents are (see Figure 2). That way, the ideal retrieval technique should return these documents in order of relevance, and permutations within the same group are not penalized. With this new form of ground truth, there does not need to be any prefixed scale of relevance, and human assessors only need to be sure that any pair of documents is well ordered according to their similarity to the query.

In the first edition of MIREX, a Symbolic Melodic Similarity task was run using ground truths based on partially ordered lists [7]. These lists have also been widely accepted by the research community as the most comprehensive means to evaluate new retrieval techniques, such as [8][9][10] and [11]. However, they have proven to be expensive to generate, which forced the MIREX evaluations to move to traditional level-based relevance judgments in the 2006, 2007 and 2010 editions.

In this paper we propose a modification of the original methodology followed to create these lists, by means of crowdsourced preference judgments that allow the candidate documents to arrange and aggregate themselves into relevance groups [12]. We implemented it with Amazon Mechanical Turk (AMT), as an attempt to explore its suitability for music tasks. Indeed, we show that our method generates lists very similar to the original ones with far less cost and no need for music experts.

Copyright is held by the author/owner(s).  
SIGIR '10, July 19–23, 2010, Geneva, Switzerland.

The rest of the paper is organized as follows. In Section 2 we describe the issues that motivate our work, reviewing the current methodology followed to create these ground truths and some of its problems. Section 3 presents our alternative methodology, and Section 4 shows how we implemented it with Mechanical Turk. In Section 5 we summarize the results obtained, showing that our alternative leads to very good results in terms of cost, consistency and agreement between assessors. The paper finishes in Section 6 with conclusions and lines for future research.

## 2. MOTIVATION

Ground truths based on partially ordered lists have two main drawbacks: they are hard to replicate and expensive to generate in terms of man-power, and they have several inconsistency problems where equivalent music pieces are judged differently.

### 2.1 Expensiveness

In the original lists created by Typke et al. [5], 35 music experts were needed for 2 hours to generate the ground truth for just 11 queries, and only 11 of them were able to work on all queries. This exceeds MIREX's human resources for a single task [2]. In part because of this restriction, the official MIREX evaluations were forced to move to traditional level-based relevance judgments from 2006 on. Two different scales were used: BROAD and FINE. The BROAD scale contained 3 levels: not similar (NS), somewhat similar (SS) and very similar (VS). The FINE scale was numerical, ranging from 0.0 to 10.0 with one decimal digit (note that this is not different than an ordinal scale with 101 levels). This choice of relevance scales presented several issues concerning assessor agreement, and the line between levels was again found to be very diffuse [6][2].

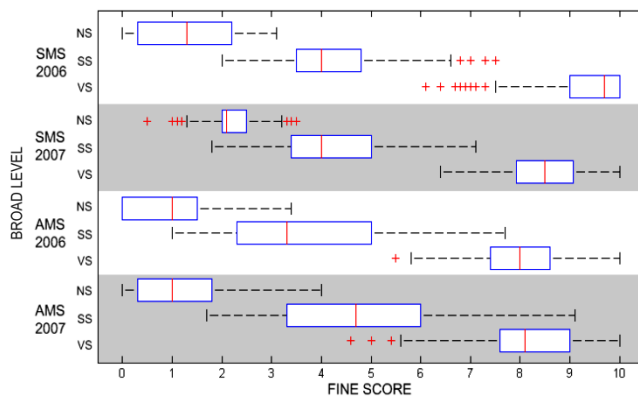


Figure 1. Distribution of FINE scores across BROAD levels, for the SMS and AMS tasks in 2006 and 2007. Taken from [2].

Figure 1 shows the distribution of FINE scores given across BROAD levels, both for the 2006 and 2007 editions of MIREX's Symbolic Melodic Similarity and Audio Music Similarity tasks. It can be seen that there was a great overlap between the FINE scores corresponding to the SS BROAD level and the NS and VS levels, as well as a large number of outliers, indicating that assessors were not very consistent when facing two different relevance scales. This is, again, evidence on the difficulty that relevance assessment poses for these tasks.

### 2.2 Inconsistencies Due to Ranking

The original method to generate ground truths based on partially ordered lists, as described in [5], was used with the RISM A/II collection [13], which at the time contained about half a million musical incipits (short excerpts from the beginning of musical

pieces). The methodology followed may be divided in four steps: filtering, ranking, arranging and aggregating:

1. *Filtering.* Several musical features were calculated for each document (musical incipits in this case). Filtering by these features and using several melodic similarity algorithms, the initial collection was gradually narrowed down to about 50 candidate incipits per query.
2. *Ranking.* For each query, 35 experts ranked its candidates in terms of melodic similarity to the corresponding query. Incipits that seemed very different from the query could be left unranked. A limit of 2 hours per expert was imposed, so not every expert could work on every list.
3. *Arrangement.* Incipits were arranged according to the median of their rank sample, using the means to solve possible ties. Therefore, the incipits that on average were ranked higher by the experts appeared with higher ranks in the ordered list.
4. *Aggregation.* Incipits with similar rank samples were aggregated within a group, so as to indicate that they were similarly relevant to the query. Thus, a retrieval system could return them with their ranks swapped and still be considered correct. The Mann-Whitney U test (also known as Wilcoxon Rank-Sum test) [14] was used between the rank samples of two incipits to tell whether they were similar or not.

Several works have noted the presence of odd results in these lists [5][10][15]. The experts were instructed to disregard changes that do not alter the actual music perception, such as changes in clef or in key and time signatures. To compare, the textual counterpart of these changes would be something like changing the language of the text or replace some words with their synonyms, which do not change the actual contents but only its form [8]. Experts were also told to consider two incipits as equally relevant if one of them was part of the other.

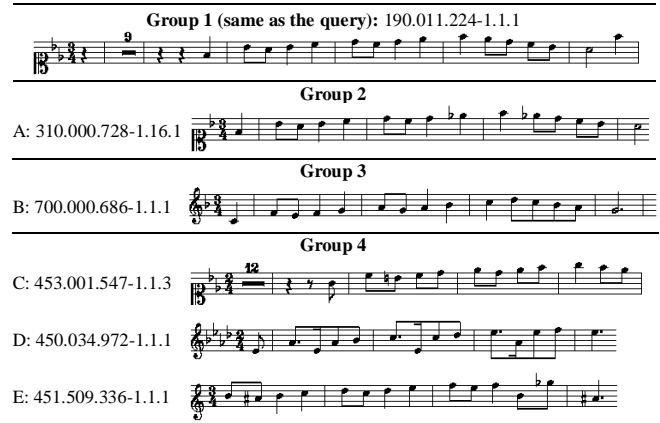


Figure 2. Excerpt of the ground truth for query 190.011.224-1.1.1.

However, incipits with such irrelevant differences ended up in different groups. For example, the second result (incipit A) expected for query 190.011.224-1.1.1 is like the query itself, but with the key signature changed (see Figure 2). Ignoring the leading silence, no listener would be able to tell the difference between this melody and the query, because they are the same note by note. Nonetheless, it was judged as less similar when compared to the query itself. The third result (incipit B) is like the second one, but with a change both in clef and key signature (see Figure 2). Again, these two melodies should be considered as equally similar to the query, but they ended up in different groups of relevance anyway.

<b>Group 1 (same as the query)</b>	
600.053.475-1.1.1	
<b>Group 2</b>	
A: 550.018.151-1.1.1	
<b>Group 3</b>	
B: 600.500.641-1.1.1	

**Figure 3.** Excerpt of the ground truth for query 600.053.475-1.1.1

The top three results for query 600.053.475-1.1.1 show similar problems (see Figure 3). The third one (incipit *B*) is just like the second one (incipit *A*) but with 3 notes missing at the end, and this one is just like the first one (the query itself) with 3 notes missing at the end too. These three results ended up in different groups of relevance according to the rankings they were given by the experts, when they were instructed to judge them as equally similar. There are also cases where incipits with virtually the same changes in the melody were placed in different groups, as with the second and third results for query 000.111.706-1.1.1.

Despite they are no longer used in MIREX, ground truths based on partially ordered lists are still used to date for the evaluation of new retrieval techniques, as they are clearly more suitable for similarity tasks than traditional assessments. However, as no new lists have been generated since 2005, in house evaluations may be overfitting to this single collection. Therefore, we strongly believe that partially ordered lists should be brought back to the official MIREX evaluations so that new test collections are adopted. For that, further research should focus on new and more affordable ways to generate them. In a previous work we analyzed and dealt with inconsistencies originated in the latter steps of the methodology to generate the lists [15], and in this paper we deal with inconsistencies originated by the experts from the very beginning, while cheapening the whole process.

It has been hypothesized that with sufficient description of the information need sought by these tasks, any reasonable person should concur as to whether a given returned item satisfies the intention of the query (in our case, whether a returned piece is similar to another one). This is called the “reasonable person assumption” [3]. We decided to use Amazon Mechanical Turk to examine whether crowdsourcing alternatives can be used to gather accurate relevance judgments without the need for experts [16][17]. Doing so, we review the reasonable person assumption, evaluate crowdsourcing for a task very different from the usual ones focused on text, and study whether this alternative is doable and produces reliable results to evaluate music similarity tasks with partially ordered lists.

### 3. ALTERNATIVE METHODOLOGY

In a first attempt to bring partially ordered lists back to the evaluation of music similarity tasks, we explored alternatives in the current methodology to make the process more affordable and work toward large-scale evaluations, while trying to minimize inconsistencies. We opted for two changes: allow assessors to indicate that certain incipits are equally relevant, and have them perform simple preference judgments [12].

#### 3.1 Equally Relevant Incipits

Reviewing the inconsistencies due to ranking (see Section 2.2), the reason seems to be clear: experts were not allowed to judge two incipits as equally relevant in the first place, they were only

able to rank one above or below the other. Under this condition, for the example list in Figure 2 they will rank first the same incipit as the query, as it is identical. Even though incipit *A* is perceived as the same melody, they will surely rank it below and not above, as it has a change in key signature, even if they are told to ignore it. Same thing happens with incipit *B* (a change in clef). One would expect the experts to randomly assign opposite orders to such pairs of incipits for their medians to average out, but that is hardly the case. For instance, half the experts might rank incipit *A* as the second most similar, and incipit *B* right after, while the other half might rank them the other way around. However, any person looking at the staves would rank *A* before *B* because its image is more similar to the query’s. In the example of Figure 3, the three incipits should be equally ranked, but the experts ranked them according to the number of notes missed. In no case should we expect such incipits to have similar ranks if we do not allow the experts to give them similar ranks in the first place.

The immediate solution to this problem would be to allow experts to specify groups of relevance from the very beginning. Also, the query-candidate pairs could be given as audio files to listen instead of as images of the corresponding staves. That way, the irrelevant changes indicated in Section 2.2 would be undistinguishable to the assessors, besides other misleading changes such as different arrangements of the stems of a group of eighth notes (quavers).

#### 3.2 Preference Judgments

It is also important to note that the experts had to judge all candidates at once for each query. That is, they had to return a list of relevant candidates ranked by similarity. It is normal to guess that they would have more problems to set up a new incipit as the list grows: the first two candidates can be easily ordered, but once the list has, say, 15 incipits, it is clearly more difficult to decide where between those 15 should the next one be placed. This phenomenon could clearly accelerate assessor fatigue, and it was already observed for the level-based relevance judgments gathered in the 2006 and 2007 editions of MIREX [2][6]. Some experts had to go back and re-judge some documents, surely after they were presented a candidate which made them realize that a previous judgment was not very congruent. This agrees with the overlapping of FINE scores across BROAD levels shown in Figure 1, and indicates, again, that the relevance for music similarity is rather continuous and the differences between levels is certainly not clear.

To alleviate this problem we propose to ask for preference judgments of the form “incipit *A* is more similar to the query than incipit *B*” ( $A < B$  for short). Carterette et al. studied the use of preference judgments for text IR and showed that they are better than traditional level-based judgments, both in terms of agreement and time to answer [12]. However, in their study they decided not to allow an option like “*A* and *B* are equally relevant” ( $A = B$  for short), which we must permit in our case to form groups (see Section 4.1). Using preference judgments, we could implement a modified QuickSort algorithm to make the incipits auto-organize themselves following the preferences of the assessors. Such an algorithm has been shown to reduce dramatically the number of judgments needed to fully order a list, as the rate of growth in the number of comparisons is  $O(n \lg n)$ , much slower than the  $O(n^2)$  growth rate of all comparisons [12]. Table 1 shows an example.

In the first iteration of the algorithm, we choose the last document as the pivot, which is *F* in this case. The assessors would have to

answer preference judgments between  $F$  and each of the other documents. In this case, every document was judged as more similar, except for  $G$ , which was judged equally similar (or dissimilar). Therefore, a new segment appears to the left of  $F$  with all the candidates judged more relevant, and  $G$  is set up in the same group as  $F$ . For the second iteration, in the rightmost segment no judgment is needed because  $F$  and  $G$  were already compared, and  $B$  would be the pivot for the leftmost segment. Incipits  $A$  and  $C$  are judged similar to  $B$ , but  $D$  and  $E$  are judged as less similar, so they are set up in a segment to the right of  $B$ . At the end, there are 3 ordered groups of relevance formed with preference judgments. Note that not all the 21 judgments were needed to arrange and aggregate every incipit (e.g.  $G$  is only compared with  $F$ ).

**Table 1.** Example of self-organized partially ordered list. Pivots for each segment appear in bold face. Documents that have been pivots already appear underlined.

Iteration	Segments	Preference Judgments
1	$\langle C, D, E, A, G, B, F \rangle$	$C < F, D < F, E < F, A < F, G = F, B < F$
2	$\langle \langle C, D, E, A, \mathbf{B} \rangle, \langle \underline{E}, G \rangle \rangle$	$C = B, D > B, E > B, A = B$
3	$\langle \langle \underline{B}, C, A \rangle, \langle D, \underline{E} \rangle, \langle \underline{E}, G \rangle \rangle$	$C = A, D = E$
4	$\langle \langle \underline{A}, \underline{B}, C \rangle, \langle \underline{E}, D \rangle, \langle \underline{E}, G \rangle \rangle$	-

With preference judgments, the sample of rankings given to each candidate is less variable than with the original method. Whenever a candidate is preferred over another one, it would be given a rank of 1 and -1 otherwise. In case it was judged equally similar, a rank of 0 would be added to its sample. With the original methodology, on the other hand, the ranks given to an incipit could range from 1 to well beyond 20, which increases the variance of the samples. Note that, with our scheme, the two samples of rankings given to each pair of documents are the opposite and therefore have the same variance. Signed Mann-Whitney U tests can be used again to decide whether two rank samples are different or not. Because the samples are less variable, the effect size is larger, which increases the statistical power of the test and makes it more likely for it to find a true difference where there is one. As a consequence, fewer assessors are needed overall.

## 4. CROWDSOURCING PREFERENCES

The use of a crowdsourcing platform seems very appropriate for our purposes. If the reasonable person assumption holds, we could use non experts to generate a ground truth like these. Because we no longer show the image of the staves, but offer an audio file instead, no music expertise is needed. We have also seen how to use preference judgments to generate partially ordered lists instead of having assessors rank all candidates at once. Therefore, the whole process can be divided into very small and simple tasks where one incipit has to be preferred over the other, which seems perfectly doable for any non expert. Also, the number of judgments between pairs of documents can be smaller, and given that we use non experts, the overall cost should be much less.

We are not aware of any work examining the feasibility of music related tasks with crowdsourcing platforms like Amazon Mechanical Turk (AMT), so we decided to use it for our experiments. AMT has been widely used before for tasks related to Text IR evaluation. HITs (each of the single tasks assigned to a worker) have traditionally used the English language, but it has been shown recently that workers can also work in other languages such as Spanish [18]. Other multimedia tasks, such as image tagging, have also been proved to be feasible with crowdsourcing [19].

## 4.1 HIT Design

The use of preference judgments is prone to have a very simple HIT design (see Figure 4). We asked workers to listen to the query or “original melody”, as we called it. Then, they had to listen to what we called “variations”, that is, the two incipits to compare. Next, they were asked what variation was more similar to the original melody, allowing 3 options:  $A$  is more similar,  $B$  is more similar, and they are either equally similar or dissimilar. We indicated them that if one melody was part of another one, they had to be considered equally similar, so as to comply with the original guidelines. As optional questions, they were asked for their musical background, if any, and for comments or suggestions to give us some feedback.

**Evaluate Melodic Similarity**

Listen to the following original melody (beware, this melody is not always the same one):

•

And now to these two variations:

• Variation A:

• Variation B:

Which melody is more similar to the original one?

Variation A.

Variation B.

They are both equally similar or dissimilar to the original melody.

NOTE: If one melody is part of another one, they shall be considered equally similar.

Optional:

Do you have a strong musical background? Which one and for how long?  
You may answer to this question just once.

Do you have any comments or suggestions? We appreciate your input!

**Figure 4.** Example of HIT for music preference judgment.

The evaluation collection used in MIREX 2005 (*Eval05* for short) had about 550 short incipits in MIDI format, which we transformed to MP3 files as they are easier to play in a standard web browser. The average duration was 6 seconds, ranging from 1 to 57 seconds. However, many incipits start with rests (see query and incipit  $C$  in Figure 2), which would make workers lose a lot of time. Therefore, we trimmed the leading and trailing silence, which resulted in durations from 1 to 26 seconds, with an average of 4 seconds. With this cuts, the average time needed to listen to the 3 files in a HIT at least once was 13 seconds, ranging from 4 to 24 seconds. This decision agrees with the initial guidelines that were given to the experts, as two incipits should be considered equally relevant despite one of them having leading or trailing rests (i.e. one would be just part of the other). We uploaded all these trimmed MP3 files to a private web server, as well as the source of a very simple Flash player to play the queries and candidate incipits. Therefore, our HIT template was designed to display the MP3 players and stream the audio files from our server.

We created a batch of HITs for each of the iterations calculated with our methodology, and paid every answer with 2 cents of dollar (plus half a cent for Amazon’s fee). After downloading the results and analyzing them, we calculated the next preference judgments to perform and uploaded a new batch to AMT,

**Table 2.** Summary of batches submitted to Mechanical Turk.

Iteration	Pairs judged	Unique workers	Previous workers (%)	Median time per judgment (seconds)	Time to completion	Inter-agreement per pair	Cost (US \$)
1	107	32	-	26	13h 29m	0.656	26.75
2	83	20	4 (20%)	14	3h 2m	0.822	20.75
3	51	15	11 (73%)	19	3h	0.72	12.75
4	19	17	10 (59%)	30	10h 3m	0.644	4.75
5	10	16	11 (69%)	21	3h 29m	0.663	2.5
6	5	12	8 (67%)	24	2h 48m	0.732	1.25
7	4	11	7 (64%)	15.5	1h 21m	0.569	1
8	2	11	4 (36%)	24.5	28m	0.506	0.5
Total/Avg.	281	79	55%	21.75	37h 40m	0.664	70.25

corresponding to the next iteration. Whenever all pairs of incipits within the same segment had been judged, we considered that segment closed, and whenever all segments were closed, the list was completed.

## 4.2 Threats to Validity

The initial order of candidates in the first iteration and the choice of the pivot element could clearly affect the results. If the pivot chosen were the query itself, most of the incipits would be judged less similar and go to the right segment, which would not provide much information. Therefore, we randomized the initial order of incipits in the first iteration. Moreover, we always chose the last incipit of a segment as the current pivot, and for the next iteration this element would be the first one of the equally-similar segment. See for example incipit *A* from iterations 3 to 4, in Table 1.

Workers could be tempted to stop listening to the original melody (i.e. the query) after a few HITs have been answered. Then, whenever the query changes as they start judging for another list, all answers given from that point on would be plainly useless. Even within the list of a single query, there will usually be several pivots, each of which will be compared with different incipits. Likewise, if the pivot is always kept as the first or second variation, workers could stop listening to them and just listen to the other variation, which would again make every answer useless after the pivot is changed when a new segment begins to be evaluated. See for example the 3<sup>rd</sup> iteration in Table 1, where both *A* and *E* are pivots. Again, we addressed this problem by randomizing the HITs: not all HITs from the same queries were presented together, and pivots were sometimes the variation *A* and some others the variation *B*. The HIT design explicitly warned the workers about this randomization anyway.

We also have to deal with carelessness of the workers. In first trials of our experiment we found that sometimes they judged some incipits as more similar to the query than the query itself, in cases where it was clearly different. We tried to alleviate this problem by accepting workers only with a 95% or higher rate of acceptance, and by using a sufficiently large number of answers per HIT. We chose to ask for 10 different workers per HIT, which we considered enough given that fewer answers are needed to begin with (see Section 3.2). This decision was also successfully taken by Alonso et al. when crowdsourcing relevance judgments for TREC documents [17]. We also found 2 workers that always gave the “Equal” answer in exactly 8 seconds. It seemed clear to us that we were dealing with some kind of robot, so we directly blocked them from our experiments and re-assigned their HITs.

## 5. RESULTS

The 11 lists in the *Eval05* collection account for a total of 119 candidate documents to judge for relevance, ranging from 4 to 23 documents per query. In order to complete the judgments, we had

to submit 8 batches to Mechanical Turk, each corresponding to an iteration of the self-organizing algorithm. These batches were submitted from April 14<sup>th</sup> to April 17<sup>th</sup>, with some time taken between iterations to semi-automatically calculate what documents to compare for the next batch.

### 5.1 Summary of Submissions

The 119 candidate documents in the 11 lists sum up 740 pairs of candidates (i.e. the  $O(n^2)$  case). We only needed to judge a total of 281 (38%) pairs of documents to completely organize the 11 lists, which account for a total of 2810 preference judgments by the workers (see Table 2). A total of 79 unique workers performed those judgments, with an average of 55% of the workers in an iteration having worked in previous ones. It took for them almost 22 seconds in median to submit the judgments, although this time reflects only how long they took to complete the assignment since they accepted it, rather than since it was displayed to them. Summing up the time to complete all iterations, the 2810 judgments took about a day and a half.

For all the 2810 judgments the total cost of generating the ground truths was about 70 dollars. The original lists needed 35 music experts for 2 hours, and during this time only 11 of them were able to work on the 11 queries. This accounts for roughly 70 hours of the time of one single expert, which is about twice as much as we needed using non-expert workers from Mechanical Turk.

### 5.2 Worker Feedback and Music Background

Out of the 79 unique workers, 23 gave us feedback. Four of them reported very positive comments about the HITs, one asked for more money and two reported problems loading one of the MP3 files for two HITs (the other workers did not report to have such problems for the same HITs).

Five workers explicitly indicated not to have any musical background, but fourteen did. Six of them had formal musical education, mainly in college and high school, while nine reported to have been practitioners for several years. Nine played an instrument, mainly piano, and six others performed in a choir.

### 5.3 Agreement between Workers and Experts

For each of the 281 HITs (i.e. pairs of candidates) we have 10 judgments made by workers. We calculated their inter-agreement score for each HIT as follows. Consider the 45 pairs of answers given for a single HIT, adding 2 points to the score if the two workers agreed (complete agreement); adding 1 point if one judged “Equal” and the other judged either document (partial agreement), and adding nothing if they judged both documents (no agreement). The perfect agreement would sum up 90 points, so we divided the score obtained by 90 to normalize from 0 (no agreement at all) to 1 (perfect agreement). Table 2 shows the mean agreement for every HIT judged in each iteration. We can

see that the agreement among workers is very high, ranging from 0.506 to 0.822, averaging to 0.664.

It is also interesting to measure the agreement between the workers of AMT and the music experts that ranked the original lists. We compared each of the resulting 281 preference judgments (aggregating the 10 corresponding answers of the workers, see Section 3.2) with the rankings given by the experts, inferring their preference judgments as well with signed Mann-Whitney U tests over the rankings they gave to each document. Table 3 shows the results.

**Table 3.** Agreement between workers (columns) and experts (rows) for aggregated judgments. Percentages are calculated over the row total.

		Workers		
		Less (56)	Equal (110)	Greater (115)
Experts	Less (91)	38 (42%)	37 (41%)	16 (18%)
	Equal (55)	11 (20%)	31 (56%)	13 (24%)
	Greater (135)	7 (5%)	42 (31%)	86 (64%)

Not surprisingly, the agreements are fairly high. There were 155 (55%) cases of complete agreement, 102 (36%) cases of partial agreement and only 23 (8%) cases of no agreement at all. Computing a global score as before, rewarding complete agreements with 2 points and partial agreements with 1 point, the agreement between workers and experts results in 0.735. These figures serve as empirical verification of the reasonable person assumption, indicating that the notion of musical similarity, though not formally formulated, appears to be common between experts and non experts.

**Table 4.** Agreement among single workers with no music background and experts. Percentages are calculated over the row total.

		Workers with no music background		
		Less (81)	Equal (97)	Greater (193)
Experts	Less (100)	55 (55%)	27 (27%)	18 (18%)
	Equal (92)	16 (17%)	35 (38%)	41 (45%)
	Greater (179)	10 (6%)	35 (20%)	134 (75%)

We also calculated the agreement between the original experts and the 5 workers that explicitly reported no music background, the 14 that reported to have some background, and the other 60 that did not answer. The workers that reported no background fully agreed with the experts 60% of the times, partially agreed 32% and did not agree in 8% of the judgments, which accounts for a total agreement of 0.764 (see Table 4).

**Table 5.** Agreement among single workers with music background and experts. Percentages are calculated over the row total.

		Workers with music background		
		Less (70)	Equal (80)	Greater (116)
Experts	Less (70)	45 (64%)	18 (26%)	7 (10%)
	Equal (67)	15 (22%)	32 (48%)	20 (30%)
	Greater (129)	10 (8%)	30 (23%)	89 (69%)

When considering the workers that reported some background, the agreement rises to 0.78, having 62% cases of total agreement with the experts, 31% of partial agreement and 6% of no agreement at all (see Table 5).

**Table 6.** Agreement among single workers with unknown music background and experts. Percentages are calculated over the row total.

		Workers with unknown background		
		Less (426)	Equal (1230)	Greater (517)
Experts	Less (390)	218 (56%)	152 (39%)	20 (5%)
	Equal (941)	127 (13%)	707 (75%)	107 (11%)
	Greater (842)	81 (10%)	371 (44%)	390 (46%)

The 60 workers that did not report anything about musical background had an agreement score with the experts of 0.777, with 60% of total agreement, 34% of answers with partial agreement and 5% of no agreement (see Table 6). All these results support again the reasonable person assumption, as very similar agreement scores can be found not only between groups of workers, but also between single workers with and without music background. As a consequence, they also support the use of crowdsourcing platforms to gather music relevance judgments.

## 5.4 Comparison with the Original Lists

Given the high agreement scores obtained by the workers of Mechanical Turk, one would expect to obtain lists very similar to the original ones generated with experts. To measure the similarity, we considered the original lists as ground truths and the crowdsourced lists as if they were the results of a system, evaluating the ADR score that would be obtained in a real evaluation [20]. Moreover, we considered the original lists as aggregated with the *Any-1* function we proposed in [15], as the resulting lists proved to be the most consistent. Finally, and to compare lists in both directions, we considered the crowdsourced lists as ground truths and the original ones as results.

There is one important detail to note, though: both the ground truth list and the results list have groups of relevance, but the latter will be considered as a fully ranked list (i.e. a sequence without groups) when computing the ADR score. For example, consider the list  $L1 = \langle (A, B, C), (D, E) \rangle$  is taken as ground truth and the list  $L2 = \langle (A, B), (D, E, C) \rangle$  as results. When evaluating  $L2$ , it would be considered as  $\langle A, B, D, E, C \rangle$ , which results in an ADR score of 0.933 because at position 3 the document retrieved is  $D$ , when  $C$  was expected. However,  $C$  and  $D$  were judged as equally relevant. These cases depend directly on the order the documents were randomly arranged at the beginning. If the results list were  $L3 = \langle (A, B), (C, D, E) \rangle$ , which is equivalent to  $L2$ , the ADR score would be 1. To account for the random effect of the initial arrangement, we generated 1000 random versions of the lists obtained with Mechanical Turk, by randomly permuting the order of documents within the same group. The results of the comparisons appear in Table 7, with the minimum, mean and maximum ADR scores obtained for the 1000 random sets of equivalent lists.

**Table 7.** Comparison between the original lists and the lists crowdsourced, in terms of average ADR score. Columns represent lists acting as ground truth, rows for lists acting as results. The numbers between square brackets indicate the minimum and maximum scores.

		Ground truth		
		All-2	Any-1	MTurk
Results	All-2	1	0.872 [0.830-0.927]	0.824 [0.785-0.872]
	Any-1	1	1	0.850 [0.828-0.873]
	MTurk	0.943 [0.915-0.977]	0.840 [0.812-0.881]	1

When compared to the original lists generated by Typke et al. (i.e. *All-2*), the crowdsourced lists performed exceptionally well, with very high ADR scores across the 11 queries, between 0.915 and 0.977. As expected, the *Any-1* lists reduce the scores because they are more restrictive than the *All-2* alternatives, although the averages are still high over 0.812. When using the crowdsourced lists as ground truth, the average across the 11 queries is still high. The *Any-1* lists would obtain a higher score than the *All-2*, showing that the crowdsourced lists are also more restrictive than

the original ones. These results confirm that the lists generated with Mechanical Turk workers are, in fact, very similar to the ones generated by experts, as already anticipated by the high agreement scores.

## 5.5 Judgments Consistency

We examined the crowdsourced lists to check whether inconsistent results like the ones described in Section 2.2 did still appear or not, and in several cases they did not. For example, the first two incipits in Figure 3 ended up in the same group of relevance, at the top of the list, as did the first three incipits in Figure 2. Other lists, like the one for query 600.054.278-1.1.1, also showed such correct variations.

## 5.6 MIREX 2005 Results Revisited

The question is whether those small variations in the lists would affect the evaluation of real systems or not [1]. We re-evaluated the 7 systems that participated in the MIREX 2005 Symbolic Melodic Similarity task with the crowdsourced ground truth lists. In addition, we also re-evaluated and compared the *Splines* method we proposed in [8] (see Table 8). Again, we compare also with the Any-1 version of the original lists.

**Table 8.** ADR results of the systems that participated in MIREX 2005 with the original and crowdsourced lists. GAM = Grachten, Arcos and Mántaras; O = Orio; US = Uitdenbogerd and Suyoto; TWV = Typke, Wiering and Veltkamp; L(P3) = Lemström (P3), L(DP) = Lemström (DP); FM = Frieler and Müllensiefen. Best scores appear in bold face. \* for significant difference at the 0.05 level and \*\* at the 0.01 level.

	Splines	GAM	O	US	TWV	L(P3)	L(DP)	FM
All-2	<b>0.71</b>	0.66	0.65	0.642	0.571	0.558	0.543	0.518
Any-1	<b>0.646*</b>	0.583	0.593*	0.594*	0.556	0.515	0.494*	0.483*
MTurk	<b>0.6**</b>	0.574*	0.572*	0.546**	0.517*	0.51*	0.467*	0.462*

As with the *Any-1* version, the crowdsourced lists seem to be more restrictive than the original *All-2*. All systems get reductions in average ADR score between 9% and 15%, and all these differences were statistically significant according to 1-tailed paired Mann-Whitney U tests. The important result is, however, that the ranking of systems is exactly the same as with the original lists. That is, the crowdsourced lists ranked the 7 systems in terms of average ADR score as the original lists did. This, again, supports the use of our methodology for evaluation of music similarity tasks.

## 6. CONCLUSIONS AND FUTURE WORK

Ground truths based on partially ordered lists represented a big leap towards the scientific evaluation of music similarity tasks. They have been widely accepted by the community, but their use in the MIREX evaluations was interrupted mainly because of their expensiveness in terms of man-power and need for music experts.

In this paper we have proposed a modification of the methodology followed to generate these lists, and we have implemented it with Amazon Mechanical Turk to gather music relevance judgments, showing that crowdsourcing platforms are viable alternatives for the evaluation of music retrieval systems. This allowed us to review the reasonable person assumption, which may lead to more affordable and large-scale evaluations without the need for music experts. We provided empirical evidence supporting it, showing high agreement scores between workers and experts.

Our methodology has several advantages. Fewer assessors are needed to judge, so more queries can be evaluated with the same man-power. Preference judgments are easier to perform, and the number of actual judgments made by the assessors is far less,

because they do not need to assess where between several candidates should a new incipit be placed. Allowing judgments of the form "A and B are equally similar", we avoid inconsistency problems where incipits equal for all purposes were judged differently. Offering the incipits as audio files instead of images, also helps in this matter, and it seems to avoid the necessity of having experts.

Further research should focus on the sorting algorithm used to organize incipits. The choice of good pivots is essential, and more empirical research should focus on the nature of music similarity to assess whether it is transitive or even symmetrical. That is, if *A* is preferred over *B* and *B* is preferred over *C*, will *A* be preferred over *C*? And if *A* is preferred over *B* for query *C*, will *C* be preferred over *B* for query *A*? So far it has been assumed that these properties hold, but such assumption should be subject of further experimental studies. In case it were valid, more work in the line of Carterette et al. should be carried out to minimize the number of judgments needed to sort all candidates and find true differences in the performance of retrieval systems [21].

## 7. ACKNOWLEDGMENTS

We would like to thank Omar Alonso for his thoughtful comments on Mechanical Turk and the paper itself. We also thank Carlos Gómez, Rainer Typke, and the IMIRSEL group, especially Stephen Downie and Mert Bay, for providing us with the MIREX 2005 evaluation data.

## 8. REFERENCES

- [1] Voorhees, E.M. The Philosophy of Information Retrieval Evaluation. *Workshop of the Cross-Language Evaluation Forum* (2002), 355-370.
- [2] Downie, J.S., Ehmann, A.F., et al. The Music Information Retrieval Evaluation eXchange: Some Observations and Insights. *Advances in Music Information Retrieval*. W.R. Zbigniew and A.A. Wierzchowska. Springer. 2010. 93-115.
- [3] Downie, J.S. The Scientific Evaluation of Music Information Retrieval Systems: Foundations and Future. *Computer Music Journal*. 28, 2 (2004), 12-23.
- [4] Selfridge-Field, E. Conceptual and Representational Issues in Melodic Comparison. *Computing in Musicology*. 11, (1998), 3-64.
- [5] Typke, R., den Hoed, M., et al. A Ground Truth for Half a Million Musical Incipits. *Journal of Digital Information Management*. 3, 1 (2005), 34-39.
- [6] Jones, M.C., Downie, J.S., et al. Human Similarity Judgments: Implications for the Design of Formal Evaluations. *International Conference on Music Information Retrieval* (2007), 539-542.
- [7] Downie, J.S., West, K., et al. The 2005 Music Information Retrieval Evaluation Exchange (MIREX 2005): Preliminary Overview. *International Conference on Music Information Retrieval* (2005), 320-323.
- [8] Urbano, J., Lloréns, J., et al. Using the Shape of Music to Compute the Similarity between Symbolic Musical Pieces. *International Symposium on Computer Music Modeling and Retrieval* (2010), 385-396.
- [9] Pinto, A. and Tagliolato, P. A Generalized Graph-Spectral Approach to Melodic Modeling and Retrieval. *International ACM Conference on Multimedia Information Retrieval* (2008), 89-96.
- [10] Hanna, P., Ferraro, P., et al. On Optimizing the Editing Algorithms for Evaluating Similarity Between Monophonic Musical Sequences. *Journal of New Music Research*. 36, 4 (2007), 267-279.

- [11] Grachten, M., Arcos, J., et al. A Case Based Approach to Expressivity-Aware Tempo Transformation. *Machine Learning*. 65, 2 (2006), 411-437.
- [12] Carterette, B., Bennett, P.N., et al. Here or There: Preference Judgments for Relevance. *European Conference on Information Retrieval* (2008), 16-27.
- [13] Saur Verlag, K. Répertoire International des Sources Musicales (RISM). Serie A/II, Manuscrits Musicaux après 1600.
- [14] Mann, H.B. and Whitney, D.R. On a Test of Whether One of Two Random Variables is Stochastically Larger than the Other. *Annals of Mathematical Statistics*. 18, 1 (1947), 50-60.
- [15] Urbano, J., Marrero, M., et al. Improving the Generation of Ground Truths based on Partially Ordered Lists. *International Society for Music Information Retrieval Conference* (2010).
- [16] Alonso, O., Rose, D.E., et al. Crowdsourcing for Relevance Evaluation. *ACM SIGIR Forum*.
- [17] Alonso, O. and Mizzaro, S. Can We Get Rid of TREC assessors? Using Mechanical Turk for Relevance Assessment. *SIGIR Workshop on the Future of IR Evaluation* (2009), 15-16.
- [18] Alonso, O. and Baeza-Yates, R. An Analysis of Crowdsourcing Relevance Assessments in Spanish. *Spanish Conference on Information Retrieval* (2010).
- [19] Nowak, S. and Rüger, S. How Reliable are Annotations via Crowdsourcing? A Study about Inter-annotator Agreement for Multi-label Image Annotation. *International ACM Conference on Multimedia Information Retrieval* (2010), 557-566.
- [20] Typke, R., Veltkamp, R.C., et al. A Measure for Evaluating Retrieval Techniques based on Partially Ordered Ground Truth Lists. *IEEE International Conference on Multimedia and Expo* (2006), 1793-1796.
- [21] Carterette, B. and Allan, J. Incremental Test Collections. *ACM International Conference on Information and Knowledge Management* (2005), 680-687.



# Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution

John Le<sup>1</sup>, Andy Edmonds<sup>2</sup>, Vaughn Hester<sup>1</sup>, Lukas Biewald<sup>1</sup>

<sup>1</sup>CrowdFlower  
455 Valencia Street  
San Francisco, CA, 94103  
+1-415-621-2343  
{john, vaughn, lukas}@crowdfLOWER.com

<sup>2</sup>eBay Search Science  
2065 Hamilton Avenue  
San Jose, CA, 95125  
+1-206-619-0100  
aedmonds@ebay.com

## ABSTRACT

The use of crowdsourcing platforms like Amazon Mechanical Turk for evaluating the relevance of search results has become an effective strategy that yields results quickly and inexpensively. One approach to ensure quality of worker judgments is to include an initial training period and subsequent sporadic insertion of predefined gold standard data (training data). Workers are notified or rejected when they err on the training data, and trust and quality ratings are adjusted accordingly. In this paper, we assess how this type of dynamic learning environment can affect the workers' results in a search relevance evaluation task completed on Amazon Mechanical Turk. Specifically, we show how the distribution of training set answers impacts training of workers and aggregate quality of worker results. We conclude that in a relevance categorization task, a uniform distribution of labels across training data labels produces optimal peaks in 1) individual worker precision and 2) majority voting aggregate result accuracy.

### Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Systems and Software—performance evaluation

### General Terms

Performance, Design, Experimentation, Human Factors.

### Keywords

Crowdsourcing, search relevance evaluation, quality control.

## 1. INTRODUCTION

Crowdsourcing is the use of large, distributed groups of people to complete microtasks or to generate information. Because traditional search relevance evaluation requiring expert assessment is a lengthy process [2, 3, 5], crowdsourcing has gained traction as an alternative solution for these types of high volume tasks [2, 1]. In some cases, crowdsourcing may provide a better approach than a more traditional, highly-structured judgment task because it facilitates the collection of feedback from a wide variety of viewpoints for the same comparison. Feedback from varying viewpoints naturally captures the myriad interpretations a particular problem may have.

Quality assurance is a major challenge of crowdsourcing [8, 9]. Without a rigorous quality control strategy, workers

often produce an abundance of poor judgments. Poor judgments by a worker can occur when a worker is ethical but misinterprets the designer's intent for the task. This is a case of a worker's bias introducing error. Unethical workers, who do not attempt to honestly complete tasks but simply answer as many questions as quickly as possible, are another source of erroneous judgments. In the first case we are left with some erroneous judgments which can adversely affect our results. The second case we are left with completely unusable judgments which invalidate our results.

To deal with each of these cases, we train our workers on previously defined gold standard data (training data) in a dynamic learning environment that gives instant feedback for why the answer they chose was incorrect. If a worker answers too many questions incorrectly, suggesting they are an unethical worker, we ban them from returning to the task.

By running tasks like this we saw that worker responses were influenced by the distribution of correct answers in the training data. Ethical workers naturally developed notions on how the data was distributed and actively applied what was learned to future questions. This situation is similar to that in machine learning, where classifiers develop bias towards the training data. When testing machine learning algorithms, training data must be fairly sampled from the underlying population distribution to ensure minimal bias. Unethical workers optimize their responses to maximize revenues while minimizing the time spent making judgments. For example, if a worker perceives 80% of the answers are of label *A* then they will answer *A* every time.

In this paper we attempt to quantify the influence of the dynamic learning environment by examining how the distribution of correct answers in the training data affects worker responses. We hypothesize that training data in which the distribution of correct answers is more uniform yields optimal results with respect to worker quality and aggregate majority vote result quality. We test this hypothesis on a task where we ask workers to categorize query results into four categories. We compiled a test set that had a skewed underlying distribution (a higher proportion of one label), and then trained different sets of workers on five different training sets. This will be explained in more detail in Sections 3 and 4.

## 2. BACKGROUND

Amazon Mechanical Turk (AMT) is a platform offered

by Amazon Web Services that facilitates online work between job requesters and workers from around the world. CrowdFlower is a labor-on-demand product that facilitates the completion of online microtasks among a number of labor channels including AMT. CrowdFlower provides the infrastructure for training workers via training data. In combination, these two products allow high throughput while ensuring judge quality.

Current strategies for evaluating and ensuring quality in crowdsourced tests include measurement of agreement, qualification questions, and worker trust algorithms [10, 7, 6]. When measuring quality with agreement, either by majority vote or similar methods, it is important to consider that high agreement among multiple judges may reflect a variety of factors, particularly:

1. Correctness of the chosen label
2. Cultural bias of the workers
3. Interpretation/ambiguity of the question
4. Cheating and collusion [3, 9].

Agreement assessment is often used in conjunction with worker error estimation on a previously defined gold standard [10, 7, 9]. In Snow [10] and Ipeirotis [7], gold standard answers are hidden from the workers and used in post-processing to estimate the true worker error-rate. For a movie rating categorization task, Ipeirotis showed that weighting worker responses by their error-rate on a previously defined gold standard improved accuracy from 95% to 99.8% [7].

As stated in Section 1, we use previously-defined gold standard data in an dynamic learning environment to provide instant feedback to workers when they answer these questions incorrectly. The gold standard data used in training will be referred to hereinafter as training data (interchangeably with training set). Analogously, testing data (test set) is the gold standard data against which results are reported.

### 3. DATA

The dataset came from a major online retailer’s internal product search projects. It consisted of 256 queries with 5 product pairs associated with each query. In other words, the dataset contained 1,280 search results. We will refer to each batch of five search results to a query as a result set. There were 164 distinct queries which included product queries such as: “LCD monitor,” “m6600,” “epiphone guitar,” “sofa,” and “yamaha a100.”

The training data was sampled from a dataset previously judged by a set of experts from the online retailer. The test set was taken from the same dataset without repetition. We ran five tasks where the test set had a highly skewed distribution towards “Not Matching” results; 82.67% of results were “Not Matching”, 14.5% “Matching”, 2.5% “Off Topic” and 0.33% “Spam”.

We varied the distribution of answers in the training set from one skew to the other, particularly as seen in Table 1. We attempted to vary “Matching” and “Not Matching” results as symmetrically as possible, but as we decreased the number of “Not Matching” results, the number of “Off Topic” results increased significantly as well.

**Table 1: Training Data Skew**

Experiment	1	2	3	4	5
Matching	72.7%	58%	45.3%	34.7%	12.7%
Not Matching	8%	23.3%	47.3%	56%	84%
Off Topic	19.3%	18%	7.3%	9.3%	3.3%
Spam	0%	0.7%	0%	0.7%	0%

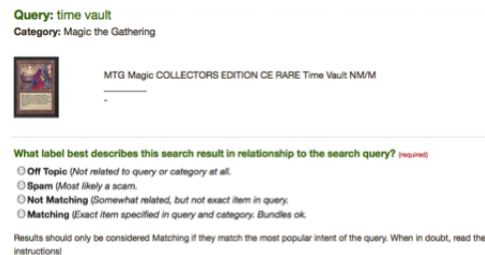
## 4. EXPERIMENTAL DESIGN

### 4.1 Amazon Mechanical Turk

We set up five tasks via CrowdFlower to be run in parallel on Amazon Mechanical Turk. The task instructions, layout, title, pricing, and design were all the same, and hence all appeared as the same task on Mechanical Turk. We paid workers 20 cents to judge the relevance of five result sets, or 25 search results.

### 4.2 Task Design

When a worker comes to our task or HIT, they see a set of instructions followed by five queries with five corresponding search results. Each query is accompanied by the category in which it was searched, if available.



**Figure 1: Example of one query-product pair in a HIT**

The instructions detail how to label search results as one of four categories: “Off Topic”, “Spam”, “Not Matching”, and “Matching”. The instructions include examples and reasons to guide workers as they make judgments. We tell workers that search results in the above categories should follow the following guidelines (shortened here for brevity):

- Matching is a result that matches the “most likely intent” of the query. We define this as the core product in the search. For instance a search for “iphone” may yield results for iPhone cases; only results for an actual iPhone are matching.
- Not Matching is a result that does not match the most likely intent of the query, but is still relevant. In the above example, we would consider iPhone cases non-matching to the query “iPhone.”
- Spam is a result that appears to be a solicitation or pornographic in nature. An image which does not picture the product but instead scantily clad women usually indicates spam.
- Off Topic is a result that is completely unrelated to the given query, though the query may appear in the result. A query for “iPod” may have a result for a car

where “iPod” is in the result string. The main product in the result is a car which has nothing to do with the iPod.

The full instruction set and task interface are available here: <<http://crowdfunder.com/judgments/mob/13838>>.

### 4.3 Dynamic Learning for Quality Control

This experiment used the training data first in an entry training module, in which each worker has to complete 20 query-result pairs successfully before proceeding to test-set questions. The workers are notified that only upon passing this section will they receive payment. We inform workers of their mistakes. After this training period, training data is used as periodic screening questions [4] to provide live feedback when workers err. The feedback explains what the correct answer should be and why. For every 20 query-result pairs a worker saw, they also were exposed to five training data questions in periodic screening.

As a worker answers these training data questions, we calculate their accuracy and use it as an estimate for the worker’s “true” accuracy. We rely on a simplified metric, the percent of correct responses, as described by Snow et al [10]. Workers are blocked from continuing on a task if their accuracy is poor. Before being blocked, a worker will receive a warning that their accuracy is too low and that they should reread the instructions to correct mistakes. Unlike in [10, 7], we did no post-processing to refine worker error estimation.

## 5. RESULTS

### 5.1 Workers

There were 255 unique workers who participated in these five experiments. There were no AMT qualifications to exclude certain workers from this task. The workers were split randomly into one of the tasks that were live simultaneously such that test group sizes were uniform. We stored the task assignment of each worker on our servers; if a worker had previously been working on a task and then resumed the session, he/she would be returned to the same task.

Routing to a task stopped if the task fulfilled its judgment needs of five trusted judgments per result. The distribution of unique workers across each task is affected by a variety of factors: individual worker output, untrusted workers, the number of judgments needed to complete each task, changes in the routing of workers away from tasks that had fulfilled their judgment needs, etc. Table 2 shows the distribution of worker involvement.

**Table 2: Worker Distribution**

Experiment	1	2	3	4	5
Came to the task	43	42	42	87	41
Did Training	26	25	27	50	21
Passed Training	19	18	25	37	17
Failed Training	7	7	2	13	4
Percent Passed	73%	72%	92.6%	74%	80.9%

### 5.2 Individual Worker Quality

In the experiments where the underlying distribution skewed toward “Not Matching,” individual worker test accuracy increased as the training set more closely reflected the underlying distribution. Optimal worker accuracy is achieved

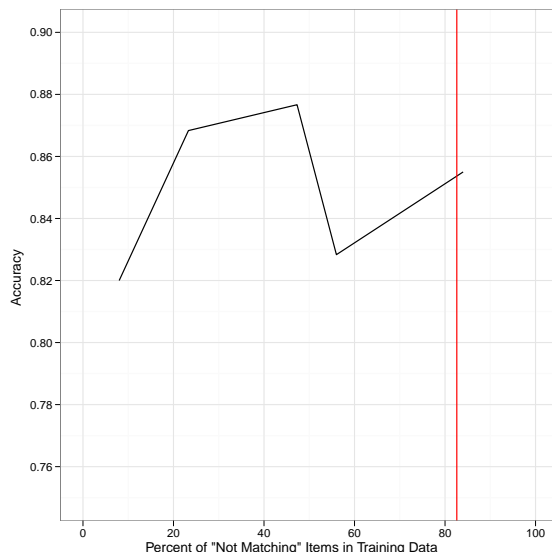
when training distributions match the population distribution. But when the test set is highly skewed, other measures may be more effective since a worker can achieve 82% accuracy by answering all “Not Matching.” Worker precision on “Not Matching” labels peaked when the training answers were uniform over the labels (Table 3).

**Table 3: Average Worker Performance Measures**

Worker \ Experiment	1	2	3	4	5
Accuracy (Overall)	0.690	0.708	0.749	0.763	0.790
Precision (Not Matching)	0.909	0.895	0.930	0.917	0.915
Recall (Not Matching)	0.704	0.714	0.774	0.800	0.828

### 5.3 Aggregate Result Quality

The aggregate majority vote results have the greatest accuracy (87.67%) when the distribution of training data answers is the most uniform. This accuracy is 5% greater than baseline accuracy (82.67%) and 12.77% greater than individual worker accuracy as shown in Table 3. Baseline accuracy for the aggregate results would be defined as a majority of workers answering all questions as “Not Matching.” When the training data distribution is the same as the underlying distribution accuracy was 85% (3% above baseline accuracy).



**Figure 2: Aggregate Accuracy on Test Data**

In these experiments the underlying distribution is so heavily skewed towards one label that we may want to optimize other performance metrics. As seen in Table 4, each measure is maximized in Experiment 3 (which contained a uniform training set distribution).

**Table 4: Aggregate Performance Measures**

Experiment	1	2	3	4	5
Precision	0.921	0.932	0.936	0.932	0.912
Recall	0.865	0.917	0.919	0.863	0.921

Note that as the training distribution more closely reflects the skewed population distribution in experiment 5, recall exceeded precision. A majority of workers were optimizing for the “Not Matching” label, labeling more items as “Not Matching” at the cost of being less precise.

## 6. DISCUSSION

The task environment is one in which workers can learn what is expected of them as they progress through the task. This enables the task designer to incorporate more detailed instructions with the expectation that workers can and will skip the instructions to immediately begin answering questions. We anticipate that many workers will only reference the instructions upon notification of their mistakes.

The training method is analogous to training a classifier with a machine learning algorithm. Most machine learning algorithms are applied using a randomly selected training set, which would be expected to approximate the underlying distribution.

We found that workers yielded greater precision on identifying “Not Matching” items when they were trained on a training set with a more uniform distribution of correct answers. Results aggregated by majority vote had greater accuracy even though the test set had a skewed distribution towards “Not Matching” items.

Workers are very adept at realizing that items are heavily skewed to a certain label (an anchoring effect) and may be predisposed to select the label with the highest prior. Workers would then be more likely to miss items that deviate from their expectations. Thus in this learning environment, training questions should predispose no bias.

This phenomenon may be due in part to workers’ ability to learn testing data as they are exposed to it. Machine-learned classifiers generally cannot learn from test data as it is processed, which is why it is so important to have robust training sets. Humans are not machines, so when doing machine-learning-like tasks where we use humans as classifiers, we must apply different techniques to train them. Tong et al [11] noted that incorporating active learning methods in training machine-learned classifiers may offer improvements to traditional methods. This result may also imply that strategies for training humans could inform future research on machine learning algorithms.

This experiment suggests broader implications for practitioners; namely that a dynamic learning environment can be used strategically to: 1) identify unethical workers and 2) train ethical workers more effectively. However, the attributes of the learning environment are critical. In particular the choice of training examples will affect worker output. Further development and application of these principles will enable us to approach search relevance tasks involving ambiguous queries or even more complex tasks that require domain-specific knowledge.

## 7. FUTURE WORK

We shall run more experiments to further validate these results. Future research should also extend the learning environment, possibly by incorporating active learning methods to train workers on similar examples of items they got incorrect and by developing a more refined model for estimating the “true” error rate of workers using a full multinomial model [10]). Having such a model for worker responses may

better show why workers are getting questions wrong and may also point to difficulty and ambiguity in our task. If we differentiate workers by demographics we may also be able to identify cultural differences, which could in turn improve task design.

## 8. NOTES

We have run this type of task numerous times over the past year on AMT, and as such workers may have entered the job with expectations as to what answers allow them to pass training questions. On previous runs of this task, items were overwhelmingly “Matching” (about 80%). Because repeat workers can learn the training data through repeated exposure, our experiments skewed the distribution of items towards “Not Matching.” We point out that the training data used for these experiments had not been used previously on any crowdsourcing platform.

We priced this task at an extremely high rate for an AMT task. An unusually high price tends to attract many opportunistic or untrustworthy workers. Part of the goal of this experiment was to engage a diverse set of both ethical and unethical workers.

## 9. ACKNOWLEDGMENTS

We would like to thank Brian Johnson (eBay), James Rubinstein (eBay), Aaron Shaw (Berkeley), Alex Sorokin (CrowdFlower), Chris Van Pelt (CrowdFlower) and Meili Zhong (PayPal).

## 10. REFERENCES

- [1] O. Alonso. Guidelines for designing crowdsourcing-based relevance evaluation. In *ACM SIGIR*, July 2009.
- [2] O. Alonso, D. E. Rose, and B. Stewart. Crowdsourcing for relevance evaluation. *SIGIR Forum*, 42(2):9–15, 2008.
- [3] B. Carterette, P. N. Bennett, D. M. Chickering, and S. T. Here or there: Preference judgments for relevance. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, 2008.
- [4] J. S. Downs, M. B. Holbrook, S. Sheng, and L. F. Cranor. Are your participants gaming the system?: screening mechanical turk workers. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 2399–2402. ACM, 2010.
- [5] G. Dupret, V. Murdock, and B. Piwowarski. Web search engine evaluation using click-through data and a user model. In *Proceedings of the Workshop on Query Log Analysis WWW2007*, May 2007.
- [6] C. Grady and M. Lease. Crowdsourcing document relevance assessment with amazon’s mechanical turk. In *NAACL/HLT 2010 Workshop on Creating Speech and Language Data With Amazon’s Mechanical Turk (at the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics)*, 2010.
- [7] P. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *KDD-HCOMP '10*, 2010.
- [8] J. Kamps, S. Geva, C. Peters, T. Sakai, A. Trotman, and E. Voorhees. In *Report on the SIGIR 2009 workshop on the future of IR evaluation*, volume 43, pages 13–23. ACM, 2009.
- [9] G. Kazai and N. Milic-Frayling. On the evaluation of the quality of relevance assessments collected through crowdsourcing. In *Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation*, pages 21–22, 2009.
- [10] 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 '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics, 2008.
- [11] S. Tong, D. Koller, and P. Kaelbling. Support vector machine active learning with applications to text classification. In *Journal of Machine Learning Research*, pages 999–1006, 2001.

# An Analysis of Assessor Behavior in Crowdsourced Preference Judgments

Dongqing Zhu and Ben Carterette  
Department of Computer & Information Sciences  
University of Delaware  
Newark, DE, USA 19716  
[zhu | carteret]@cis.udel.edu

## ABSTRACT

We describe a pilot study using Amazon’s Mechanical Turk to collect preference judgments between pairs of full-page layouts including both search results and image results. Specifically, we analyze the behavior of assessors that participated in our study to identify some patterns that may be broadly indicative of unreliable assessments. We believe this analysis can inform future experimental design and analysis when using crowdsourced human judgments.

## Categories and Subject Descriptors

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

## General Terms

Experimentation, Measurement, Human Factors

## Keywords

crowdsourcing, preference judgements, experimental design

## 1. INTRODUCTION

The search engine results page (SERP) layout has a great impact on users’ searching experience. With the emergence of images, ads, news, blog posts, and even micro-blog posts in the search results, how to smartly integrate them into web pages to assist searching becomes an interesting though challenging problem. As the first step, we explore how to make the search results layout more user-friendly by varying the positions of images relative to ranked results. Images can assist users in finding the information they want and make searching more effective. On the other hand, images may take up too much valuable space on the SERP due to their sizes.

This work describes a pilot study we performed to determine the optimal placement of images among search results. The objects to be assessed are *full page* layouts consisting of both ranked results and results from image search verticals.

It is the positions of images and ranked results relative to one another rather than the relevance of individual items on the page that we are interested in assessing. Because we believe the quality of a full layout is difficult to assess on an absolute scale, we used preference judgments: assessors see two different possible layouts and choose the one they prefer. Over a large number of assessors, we should be able to determine the most preferred general layout, as well as specific queries that deviate from the overall most preferred.

Preference judgments can be made quickly and are less prone to disagreements between assessors than absolute judgments [2]. Using preference judgments to evaluate is also more robust to missing judgments than using absolute judgments to evaluate [1]. Preferences between full page layouts seem to correlate well to traditional evaluation measures based on absolute relevance judgments on documents [9]. Finally, preference judgments can be mapped to much finer-grained grades of utility than is possible with absolute judgments [8]. Preference judgments do have some problems: the number of pairs to be judged grows quadratically rather than linearly, and assessors seem to find the increased number much more tedious [2]. Furthermore, when assessing full layouts, the number of objects that need to be assessed can grow factorially as individual items are rearranged relative to one another. And depending on how much layouts are allowed to vary, there can be a “credit assignment problem”: it is difficult to tell *why* an assessor prefers one layout to another.

Nevertheless, preference judgments seem an ideal tool for this task, and because we both need many of them and they can be made quickly, they seem to be an ideal candidate for crowdsourcing via Amazon’s Mechanical Turk<sup>1</sup> (MTurk) or some other system. The fact that they can be made quickly, however, may lead assessors being paid very low rates per judgment to “cheat” or produce otherwise unreliable data in various ways so that they can make more money without expending much effort. When data is unreliable, it leads to bias in judgments and possibly severe errors in evaluations [4]. Experimenters naturally would like to be able to prevent, detect, and compensate it [6][10][7]. But this cheating creates an adversarial relationship with the experimenter, in that as experimenters learn how to detect cheating, the assessors find new ways to cheat them. In our pilot study, we discovered that assessors seem to be cheating in ways that are not initially obvious, and further that they will sometimes cheat on one task while seemingly taking another

Copyright is held by the author/owner(s).  
SIGIR’10, July 19–23, 2010, Geneva, Switzerland.

<sup>1</sup><http://www.mturk.com>

seriously.

The rest of this paper is organized as follows: in Section 2, we describe our experimental design. In Section 3, we provide some analysis on the behavior of MTurk workers, and in Section 4 we summarize our results and describe directions for future work.

## 2. EXPERIMENTAL DESIGN

We set up an online survey that asks assessors to give their preferences on variations of the Yahoo! SERP layout for 47 queries formed from 30 topics taken from the TREC 2009 Web track [5] and the TREC 2009 Million Query track [3]. To limit the space of possibilities, we selected only queries that have results from an image vertical but not from any other vertical. We keep the search results fixed (we always use the same 10 URLs with the same summaries), and we insert image vertical results into one of three places: above all search results (top), below the top five results (middle), and below all ten results (bottom). In addition to results from the image vertical, some URLs have an inline image associated with them as well. These we displayed to either the left or the right of the summary. In all, each query had up to six different layout variants: queries with only inline images had two variants, queries with image vertical results had three variants, and queries with both had six. Two layout variations are shown side by side to the assessors as illustrated in Fig. 1.

We take the advantage of Amazon Mechanical Turk as a platform to involve search engine users with different backgrounds around the world. In Mechanical Turk, “requesters” submit “HITS” (Human Intelligence Tasks) to be completed by “MTurkers” (Mechanical Turk Workers). We redirect MTurkers from our HIT (Human Intelligent Task) question to our own survey website, which allows us to show each MTurker a sequence of preferences and to log additional information such as time-on-task. MTurkers complete the survey, submit the confirmation code at the end of the survey via the HIT, and get paid US\$0.13 for every 17 preferences they complete once their submissions are validated.

There are three different batches of survey questions, corresponding to queries with inline images only, image vertical results only, and both types. Each batch consists of 17 queries, and for each query there is a single preference judgment. We limited to one preference judgment per query per assessor to control possible learning on topic effects, even though it prevents us from acquiring all preference judgments for a query from a single assessor. The order of showing those 17 queries is randomized to control possible order effects and to allow analysis on whether assessors’ behavior was changing as they learned about the task. For each query, two result pages are randomly selected and presented to the user, who may choose the “left” layout, the “right” variant, or express “no preference”. Though each batch has different number of layout variants, we can easily control the data size required for each batch by letting user groups of different sizes to complete the survey.

We added an additional absolute-scale rating task for each query. Users not only give preferences for layouts but also rate the pictures by relevance on a ternary scale (“poor”,



Figure 1: A sample survey question page. The first question asks assessors to make their preference. The second question asks assessors to rate the images in the SERP.

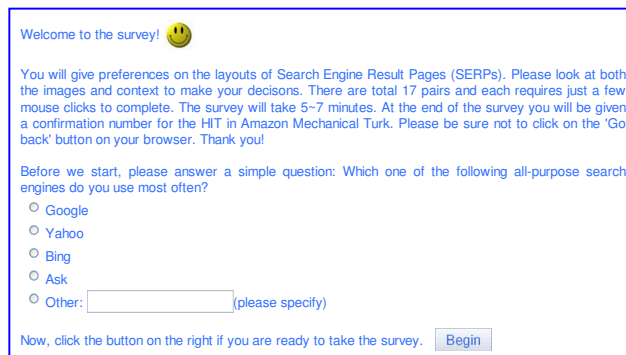


Figure 2: Welcome page of the survey with a simple question to identify user’s favorite search engine.

“OK”, “good”). In this way we can test whether the relevance of the pictures is also an important factor that influences the layout preference.

Out of the 17 queries, two are “trap” questions (following Sanderson et al. [9]) that have two identical result pages. We put them at the 6th and 15th respectively in the survey question sequence. Thus, assessors should have no preference for those pairs. The purpose of setting up the trap questions is to detect dishonest workers—if they do not se-

lect “no preference”, they are probably not paying attention.

Finally, we add a simple question at the beginning of the survey to identify a user’s favorite search engine (Fig. 2). The purpose of this is to determine whether expectation plays any role in preference: Yahoo!’s default for inline images is to display them on the right, while Google’s is to display them on the left (Bing’s seems to vary by query). Our hypothesis is that Yahoo! users may prefer the right while Google users may prefer the left, at least initially.

### 3. DATA ANALYSIS

First we rejected the HITs that failed our trap question. After that, this pilot study produced a total of 25 approved HITs for which we had timing information (seconds per judgment), preferences, image ratings, and search engine preference. 24 of 25 preferred Google, so we were not able to test our hypothesis about expectations.

#### 3.1 Time analysis

We plot the time in seconds that assessors spend on each preference judgment against the query sequence. Three general types of assessors are found and three representative curves are shown in Fig. 3. Fig. 3(a) shows the **Normal** pattern: the assessor starts out slow, quickly gets faster as he or she learns about the task, and then roughly maintains time. This is expected because assessors who never did this survey before require more time on the first few questions, but as they get more familiar with the questions, they make quicker responses. 17 out of the 25 assessors (68%) fall into this category.

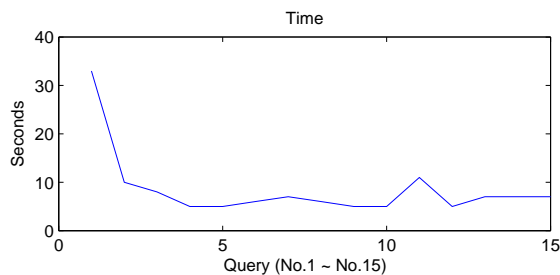
Fig. 3(b) shows the **Periodic** pattern which indicates very strange user behavior: some judgments are made fast and some are made slow, but the fast and slow tend to alternate. One possible explanation might be that these assessors were not fully dedicated to doing the survey. They are probably not purposely cheating, but they might be absent-minded periodically, calling their data into question. 6 out of the 25 assessors (24%) fall into this Periodic category.

Fig. 3(c) shows the **Interrupted** pattern in which occasional peaks appears under the background of Normal pattern. Users who have this kind of pattern might be interrupted in the middle of the survey, e.g., receiving a phone call while doing the survey; it seems unlikely they are cheating or allowing the interruption to affect their results significantly. Only 2 assessors fall into this category.

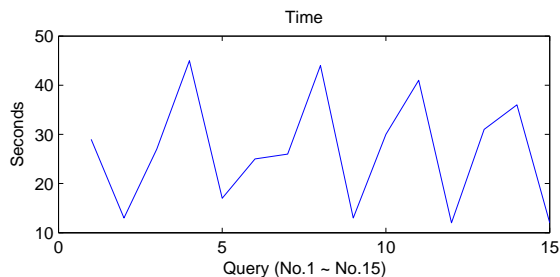
#### 3.2 Image rating analysis

As to the image ratings in Fig. 4, they also exhibit 3 different patterns. Fig. 4(a) shows the **Normal** pattern. Users give rating 2 or 3 most often and give rating 1 occasionally (suggesting that Yahoo!’s image results are pretty good, as we expect). 21 of 25 assessors (84%) have this Normal time pattern.

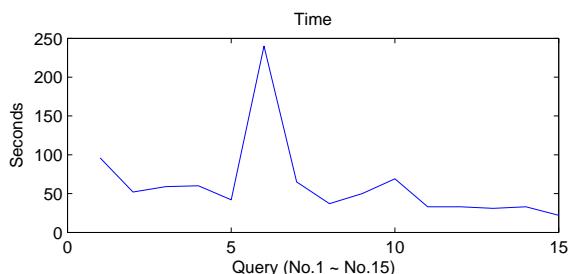
Fig. 4(b) shows a **Periodic** pattern similar to that observed for timing: a user shows a tendency to alternate between a subset of ratings. These assessors may be cheating, but exhibiting an advanced cheating behavior by purposefully trying to “randomize” their responses so that it would be



(a) **Normal** assessors start out slow and quickly get up to a consistent judgment speed.



(b) **Periodic** assessors vacillate between relatively fast judgments and relatively slow judgments.



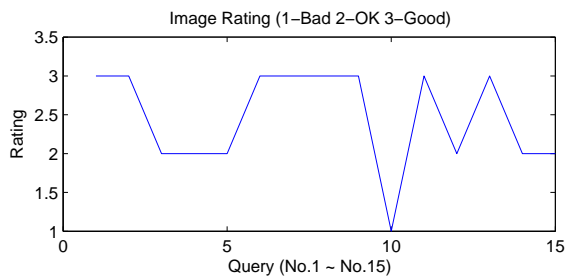
(c) **Interrupted** assessors look like **normal** assessors except for a large spike in time.

**Figure 3: Each plot shows the time an assessor took to make a preference judgment for each of 15 queries in a randomized order. “Trap” queries have been excluded. The plots are not averages; each is an exemplar of one of the cases.**

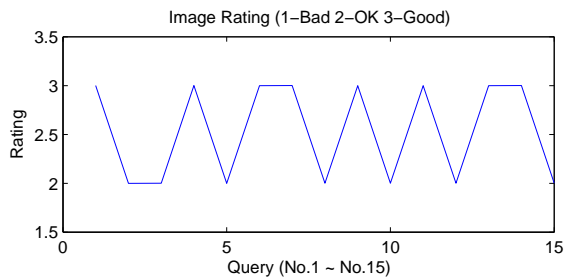
difficult for requesters to discover their dishonesty. 2 out of 25 assessors (8%) fall into this category and we find that one of these two assessors also shows a periodic time pattern.

Note that Figures 4(a) and 4(b) actually have roughly equal numbers of “good” judgments. If indeed images for 8 of the 15 queries can be given the highest rating, then because the order is randomized, there is some chance that an honest assessor will actually produce such a periodic pattern. However, the probability of any periodic or even quasi-periodic (i.e., with some short repetitions) pattern being observed due to chance alone is very low—we estimate it to be less than one in a thousand. It therefore seems safe to conclude that both assessors produced invalid data.

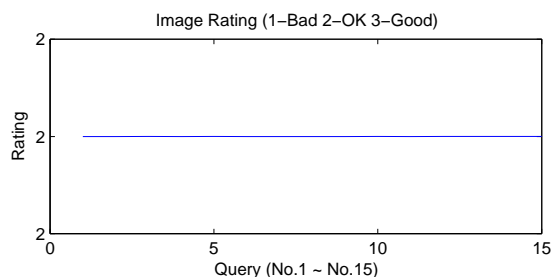
Fig. 4(c) shows the **Fixed** pattern in which all or most of the images rating are the same. 2 out of 25 assessors (8%) have the Fixed pattern. This type of assessor seem clearly not interested in rating the images and thus almost always



(a) **Normal** assessors demonstrate no clear pattern in their image relevance judgments.



(b) **Periodic** assessors vacillate between two or more ratings in a consistent way.



(c) **Fixed** assessors give every image set the same judgment.

**Figure 4: Each plot shows the rating an assessor gave to the set of images retrieved by the image vertical.**

give a fixed rating.

### 3.3 Preference judgment analysis

In this section we analyze preference judgments for the mixed set (both inline and vertical image results) to determine whether we can identify one or the other as the primary factor in the assessor’s preference. If so, we may be able to (partially) address the “credit assignment problem” described in Section 1.

We separately analyzed the inline placement and vertical placement preferences for the time being. We assign TMB (top/middle/bottom vertical variants) and LR (left/right inline variants) scores to indicate the layout preferences according to the following method:

- Given a T-B pair, if the user prefers T, we assign 1.5 as TMB score of that pair. Otherwise, we assign -1.5.
- Given an M-B pair, if the user prefers M, we assign 1 as TMB score of that pair. Otherwise, we assign -1.

- Given a T-M pair, if the user prefers T, we assign 1 as TMB score of that pair. Otherwise, we assign -1.
- Given an L-R pair, if the user prefers L, we assign 1 as LR score of that pair. Otherwise, we assign -1.

In the example in Figure 1, if an assessor preferred the left layout to the right, the judgment would have a TMB score of 1 because the assessor prefers vertical results in the middle rather than the bottom, and an LR score of 1 because the assessor prefers inline images on the left to the right. Note that the first three items above are mutually exclusive. Thus, higher TMB and LR scores indicate the preference for higher vertical images position and left inline image position.

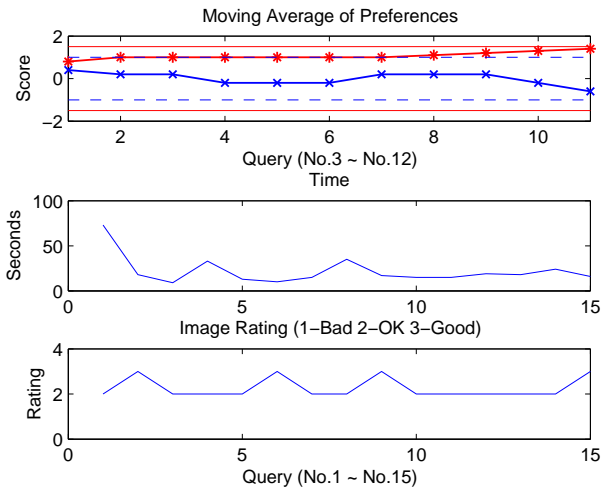
Figures 5 and 6 show moving averages of the TMB and LR scores (red and blue lines, respectively) against query number; we call these curves layout preference curves. They are moving averages to control for variance in which layout variants the assessor saw. There are roughly two patterns of layout preference curves: either only one of the scores changes over time, or both do. Fig. 5 shows two representative curves of the first pattern along with judgment time and image rating curves (note that both assessors are more-or-less **normal** types as defined above). In Fig. 5(a), the inline image preference gradually goes from a left-preference to a right-preference while the vertical image preference remains high. In Fig. 5(b), the inline image preference stays fixed on the right while the vertical image preference seems to be changing periodically (with a relatively long wavelength). We infer from this that it is the layout preference associated with the varying curve that is the leading factor in making preferences: assessor 5(a) definitely prefers vertical results towards the top, so bases his or her judgments on the position of the inline images, while assessor 5(b) definitely prefers inline images on the right, so bases his or her judgments on the position of the vertical results. The alternative possible explanation, that the assessor simply doesn’t care about one or the other type, is probably not the case: since each variant for each type was equally likely to occur in either the left or right position, the assessor had to consciously choose their consistent preference every time.

Fig. 6 shows two representative curves of the second pattern along with judgment time and image rating curves (again, both assessors seem **normal** w.r.t. time, though there is a sense that assessor 6(a) stopped doing the image rating task while assessor 6(b) has somewhat periodic ratings). In this pattern, both TMB and LR curves vary over time, so we cannot conclude that one or the other is responsible for the preference. However, we may take this as an indicator of expectations shifting as the assessor becomes more familiar with the different layouts. Assessor 6(a) starts out with a preference for inline images on the right but gradually comes to have no preference; he or she also starts with no preference for vertical image placement but gradually comes to prefer them on the top. In these cases we may need to look at each SERP pair individually to determine if TMB and LR positions have a combinational effect on layout preference.

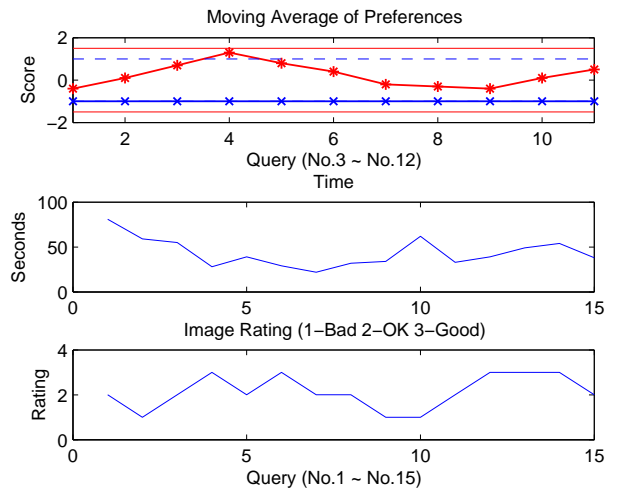
### 3.4 Analysis of rejected data

Finally, we looked at the assessors that failed the “trap” question by expressing a preference when the layouts were



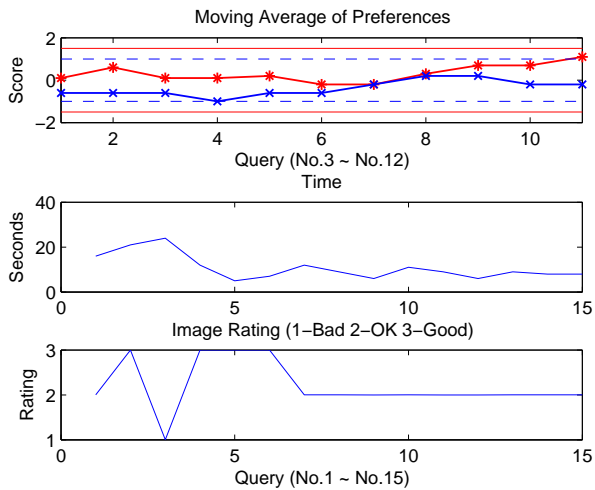


(a) Varying inline image preference curve (LR curve).

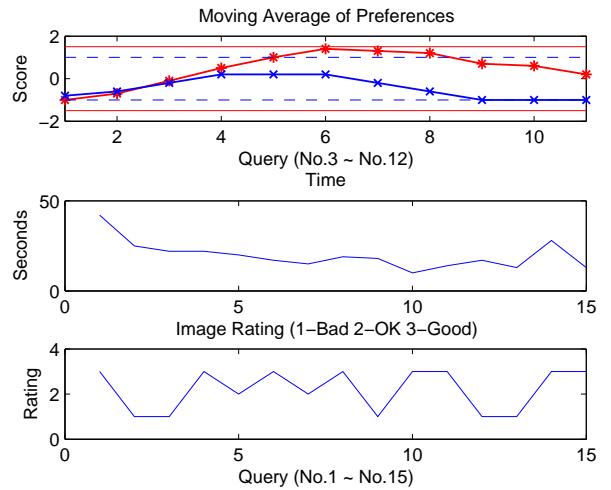


(b) Varying vertical image preference curve (TMB curve).

**Figure 5: One layout preference curve (top plots) changes dramatically over time while the other changes slightly. The solid red line with \* indicates TMB score (for vertical image preference) and the solid blue line with × indicates LR score (for inline image preference). The solid red lines are the upper and lower bounds for TMB score while the dashed blue lines are bounds for LR score. The moving window size is 5. Thus, the curve starts at Query No.3 and ends at No.12. The middle and bottom plots show the judgment time and image rating curves respectively.**



(a)



(b)

**Figure 6: Both layout preference curves (top plots) change over time. The solid red line with \* indicates TMB score (for vertical image preference) and the solid blue line with × indicates LR score (for inline image preference). The solid red lines are the upper and lower bounds for TMB score while the dashed blue lines are bounds for LR score. The moving window size is 5. Thus, the curve starts at Query No.3 and ends at No.12. The middle and bottom plots show the judgment time and image rating curves respectively.**

identical. Eight assessors were rejected for this reason; of these, one showed a periodic time pattern (Fig. 7(a)) and one showed an interruption (Fig. 7(b)). Two showed an *abnormal* time pattern of taking longer on the last few queries (Fig. 7(c), Fig. 7(d)); this was not observed among assessors that passed the trap question. One showed a fixed rating pattern (Fig. 7(e)) one showed a partially periodic rating pattern (Fig. 7(f)), and two seemed normal in both time and rating (Fig. 7(g) and 7(h)).

## 4. CONCLUSIONS

We performed a pilot study to determine whether Amazon Turk could produce useful preference judgments for distinguishing between different layouts including both search engine results and image results. Though we did not discuss it in the main body of this work, the overall results were overwhelmingly in favor (by a factor of roughly 2:1) of vertical image results near the top of the page and inline image re-

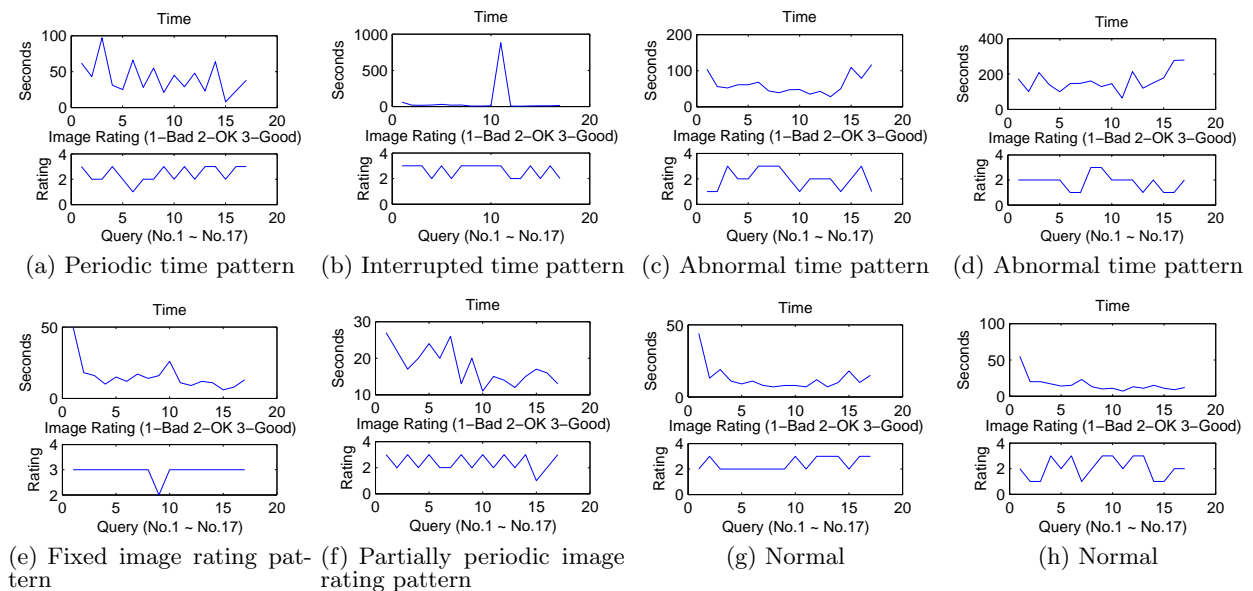


Figure 7: Analysis of assessors that failed the “trap” question.

sults on the left. However, we believe our analysis provides some interesting data for designing future studies:

1. Unreliable assessors may reveal their unreliability in different ways: some through periodic timings, some through abnormal timings, some through periodic ratings, some through fixed ratings.
2. When asked to perform more than one task, assessors may be reliable on one without necessarily producing reliable data for the other.
3. Trap questions are useful for identifying unreliable assessors, as 6 of the 8 responses rejected for failing the trap question also exhibited unusual behavior patterns.
4. Trap questions alone may not identify all the unreliable assessors, as 6 of 25 passed the trap question but showed periodic timings and 4 of 25 passed the trap question but showed strange image rating behavior.
5. The use of periodic image ratings may suggest MTurkers learning how to avoid being detected when cheating.

Certainly there is more analysis that can be done, particularly in terms of the total number of assessments needed, whether it is “safe” to have a single assessor make multiple preference judgments for the same query, and how to aggregate preferences over assessors to learn about particular queries. These are all directions we are pursuing currently.

## Acknowledgements

This work was supported in part by a gift from Yahoo! Labs and in part by the University of Delaware Research Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

## 5. REFERENCES

- [1] B. Carterette and P. N. Bennett. Evaluation measures for preference judgments. In *Proceedings of SIGIR*, 2008.
- [2] B. Carterette, P. N. Bennett, D. M. Chickering, and S. T. Dumais. Here or there: Preference judgments for relevance. In *Proceedings of ECIR*, pages 16–27, 2008.
- [3] B. Carterette, V. Pavlu, H. Fang, and E. Kanoulas. Overview of the trec 2009 million query track. In *Proceedings of TREC*, 2009.
- [4] B. Carterette and I. Soboroff. The effect of assessor errors on ir system evaluation.
- [5] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2009 web track. In *Proceedings of TREC*, 2009.
- [6] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 453–456, New York, NY, USA, 2008. ACM.
- [7] V. C. Raykar, S. Yu, L. H. Zhao, A. Jerebko, C. Florin, G. H. Valadez, L. Bogoni, and L. Moy. Supervised learning from multiple experts: Whom to trust when everyone lies a bit. In L. Bottou and M. Littman, editors, *Proceedings of the 26th ICML*, pages 889–896, Montreal, June 2009. Omnipress.
- [8] M. E. Rorvig. The simple scalability of documents. *JASIS*, 41(8):590–598, 1990.
- [9] M. Sanderson, M. L. Paramita, P. Clough, and E. Kanoulas. Do user preferences and evaluation measures line up? In *Proceedings of SIGIR*, 2010.
- [10] 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*, pages 254–263, Morristown, NJ, USA, 2008. Association for Computational Linguistics.

# Logging the Search Self-Efficacy of Amazon Mechanical Turkers

Henry Feild\*

Dept. of Computer Science  
University of Massachusetts  
Amherst, MA 01003  
hfeild@cs.umass.edu

Rosie Jones

Yahoo! Labs  
4 Cambridge Center  
Cambridge, MA 02142  
rosie.jones@acm.org

Robert C. Miller

MIT CSAIL  
32 Vassar St  
Cambridge, MA 02139  
rcm@mit.edu

Rajeev Nayak

MIT CSAIL  
32 Vassar St  
Cambridge, MA 02139  
jeev@mit.edu

Elizabeth F. Churchill

Yahoo! Research  
4301 Great America Parkway  
Santa Clara, CA  
elizabeth.churchill@yahoo-  
inc.com

Emre Velipasaoglu

Yahoo! Labs  
701 First Ave  
Sunnyvale, CA 95054  
emrev@yahoo-inc.com

## ABSTRACT

Conducting focused but large-scale studies and experiments of user search behavior is highly desirable. Crowd-sourcing services such as the Amazon Mechanical Turk allow such studies to be conducted quickly and cheaply. They also have the potential to mitigate the problems associated with traditional experimental methods, in particular the relatively small and homogenous participant samples used in typical experiments. Our current research project addresses the relationship between searcher self-efficacy assessments and their strategies for conducting complex searches. In this work-in-progress paper, we describe our initial tests of using Amazon Mechanical Turk to conduct experiments in this area. We describe a platform for logging the actions taken by Turkers, and a questionnaire we conducted to assess search self-efficacy of average Turkers. Our results indicate Turkers have a similar range of search self-efficacy scores to undergraduate students, as measured by Kelly [8]. We were able to reach a large number of searchers in a short time and demonstrated we can effectively log interactions for rigorous log-based evaluation studies. Changing the amount of remuneration Turkers received had a significant effect on the time spent filling out the questionnaire, but not on the self-efficacy assessments. Finally, we describe the design of an experiment to use Turkers to evaluate search assistance tools.

---

\*Work completed while at Yahoo!

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*SIGIR '10 Workshop on Crowdsourcing* July 19–23, 2010, Geneva, Switzerland.

Copyright 2010 ACM Copyright is held by the author/owner(s). ...\$10.00.

## General Terms

### Keywords

search, self-efficacy, crowd-sourcing, Amazon Mechanical Turk, user study, iterative method design

## 1. INTRODUCTION

This work is motivated by our current research project to study the effectiveness of search assistance. Following Compeau and Higgin's [5] research suggesting that a determinant of computer use is people's self-efficacy assessments around computer literacy or competence, our hypothesis is that search self-efficacy may affect users' willingness to interact with search assistance tools such as relevance feedback and query suggestions. Self-efficacy is defined by Bandura to be "concerned with people's beliefs in their capabilities to produce given attainments" [2]; self-efficacy measures offer an assessment of a person's confidence in their ability to perform task(s). We would like to study users with a range of search self-efficacy levels, and log their interactions with a search engine, including a variety of search assistance tools.

We have several desiderata when we attempt to evaluate the quality of search engines for web search users. Firstly we would like to evaluate over a representative sample of search users. An effective way of doing this is with live tests on a search engine such as described by Anick [1]. However, live tests have two draw-backs: they are risky in that a bad test could alienate users. In addition, the meaning of user click and interaction behavior is still an area of active research, and its relationship to goal success and user satisfaction is still only approximately understood [7, 3].

A second desirable property is to understand the range of users well. Finding study populations in universities allows us to study the users in detail, including surveying their demographics, and other properties, but these users may not be representative of general web searchers. In particular for our study we would like to sample web searchers with a range of search abilities, orientations to Internet use, and search self-efficacy levels. Getting participants who are

representative along all these dimensions is unlikely in an easily accessible, and relatively homogenous population like students at a university or workers in an Internet company. For example, considering our current area of interest, self-efficacy, Kelly [8] measured the search self-efficacy of undergraduate students, and found that they had generally high search self-efficacy. To evaluate search on such a population may overestimate the ease with which people find things, by under-representing low search self-efficacy users.

Running tests on crowd-sourcing services such as the Amazon Mechanical Turk (AMT) may mitigate the problems with both university-based and live-search based evaluations. However, there are two challenges in evaluation using workers on AMT (“Turkers”). The first is logging the searches and clicks they perform during the task. Turkers prefer not to download software or toolbars that could be used to track their interactions. The second is understanding how representative Turkers are of general web searchers.

In Section 2 we describe search self-efficacy in more detail, and give the details of the search efficacy scale we use in this work. In Section 3 we describe our preliminary results from the questionnaire on search self-efficacy of Turkers and compare it to the results obtained by Kelly [8] on undergraduate students. In Section 4 we give details of the platform which we will use to log search interactions in our full study. In Section 5 we describe some open design issues for our full study, arising from this preliminary study of Turkers. Finally in Section 6 we describe the full study we are preparing, which will measure the effectiveness of search assistance tools for searchers with different levels of search self-efficacy, and different levels of frustration.

## 2. SEARCH SELF-EFFICACY

Kelly’s search self-efficacy scale, presented below, covers a range of activities involved in searching, from general query formulation to query refinement to results filtering and management. Users are asked to rate their self-confidence on a number of tasks using a numerical scale from 1 to 10, where 1 is *totally unconfident* and 10 is *totally confident*. Questions on the scale are as follows:

I can...

1. Identify the major requirements of the search from the initial statement of the topic.
2. Correctly develop search queries to reflect my requirements.
3. Use special syntax in advanced searching (e.g., AND, OR, NOT).
4. Evaluate the resulting list to monitor the success of my approach.
5. Develop a search query which will retrieve a large number of appropriate articles.
6. Find an adequate number of articles.
7. Find articles similar in quality to those obtained by a professional searcher.
8. Devise a query which will result in a very small percentage of irrelevant items on my list.
9. Efficiently structure my time to complete the task.
10. Develop a focused search query that will retrieve a small number of appropriate articles.
11. Distinguish between relevant and irrelevant articles.
12. Complete the search competently and effectively.
13. Complete the individual steps of the search with little

	\$0.50 HIT	\$0.05 HIT
Min	47.00	28.00
Median	117.50	92.50
Mean	134.89	99.06
Max	503.00	123.75
Stddev	74.92	50.02

**Table 1: Statistics about the time (in seconds) each user spent on the surveys.**

difficulty.

14. Structure my time effectively so that I will finish the search in the allocated time.

In presenting our results, we use these numbers to reflect efficacy assessments across our participant population.

## 3. RESULTS

We ran a questionnaire on AMT two different times, each with 100 Turkers. The questionnaire asked users their age and gender in addition to the fourteen search self-efficacy questions presented in section 2. For our first presentation of the questionnaire we paid workers \$0.50 to fill out the questionnaire. The second presentation of the questionnaire paid only \$0.05. While the time of month varied, the day of the week and time of day when the AMT human-intelligence task (HIT) was released was the same. We compare the differences between these two presentations below.

### 3.1 Demographics

The populations showed very similar gender splits and a somewhat similar spread in ages. The workers that completed the first HIT consisted of 57 males and 43 females. Their ages ranged from 18 to 81, with a mean of 32 years. For the second HIT, there were 55 males and 45 females ranging in age from 18 to 62 years old, with a mean of 30.

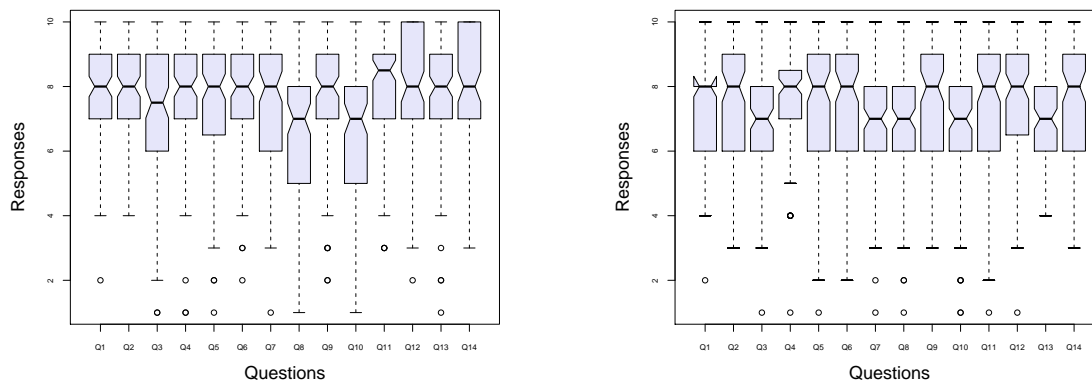
### 3.2 Time to Completion

Each HIT was released at 8:30pm American Eastern Daylight Savings Time (EDT) on two different Mondays during June 2010. The \$0.50 HIT was released first. Within 106 minutes, all 100 assignments were accepted by workers. The second HIT was issued a few weeks later and it took 540 minutes for all 100 assignments to be accepted—*five times as long*. Table 1 shows the statistics for per-worker survey completion in seconds for each of the survey versions. The means are statistically different according to a Welch’s two-sided t-test ( $p < 0.0002$ ). We see that workers spent significantly more time on the questionnaire in the first presentation, when workers were paid \$0.50 rather than \$0.05.

### 3.3 Self-efficacy Responses

Users were asked to rate their confidence in being able to perform each of the fourteen search self-efficacy questions using the scale described in Section 2. Figure 1 shows the range of responses for each question for the two presentations of the questionnaire. We can see that both plots are skewed towards the higher end, suggesting a ceiling effect.

The mean over average scores per user was 7.63 (sd=1.38; min=3.74; max=10.00) for the \$0.50 version of the questionnaire and 7.26 (sd=1.35; min=3.86; max=10.00) for the \$0.05 version. The average scores for our two HITs did not differ significantly according to a two-sided T-test ( $p = 0.054$ ). The scores seem consistent with the mean found



**Figure 1: Boxplots of the responses for each questions from the \$0.50 (left) and \$0.05 (right) versions of the questionnaire.**

by Kelly [8] over 23 students: 7.319 (sd=1.38; min=5.14; max=9.79).

#### 4. EXPERIMENT PLATFORM

Our previous study of search frustration [6] was conducted in a lab setting, which allowed a variety of custom software and sensors to be deployed and logged participants’ searching and browsing behavior in detail. Transitioning to an on-line experimental platform like AMT brings with it a cost in the richness of the information that can be logged, because the experimenter can no longer completely control and track the participant’s environment.

For our current experiments, we will be using a platform designed for AMT that also retains some of the benefits of a lab study. In the lab, we log searching and browsing behavior using an open-source browser toolbar, the Lemur Query Log toolbar, which records not only queries and result click-throughs on the search engine, but also page views on target sites. Unfortunately, it is too much to expect AMT workers to install new software in their browsers, especially software that may inadvertently violate the worker’s privacy in other browsing, unrelated to the assigned search task.

Instead, we log the search session by requiring the worker to issue searches and browse results through a proxy that we control. We show each Turker a page with a form and an imbedded frame, which points to the proxy. A diagram of the setup is shown in Figure 2. The form, located in the task pane, consists of the task to be completed and a text area where the user is required to respond to the task. The proxy frame is directed to a modified search engine interface made for the study. The proxy rewrites all links on every page that passes through so that those pages are redirected via the proxy as well. It injects JavaScript calls so that events, such as pages visited and mouse movements, can be logged. When the user has completed a task, they click the “Next” button in the task pane. This causes several hidden fields in the form to be populated with the events logged by the injected JavaScript. This data can either be uploaded to a database or sent to the outer frame in the AMT HIT page.

Alternatively, the proxy server could record a search log as pages pass through it, associating the log with a session identifier. We chose the JavaScript injection approach instead because it allows us to capture client side information,

such as mouse movements. One could inject an off-the-shelf analytics package like userfly<sup>1</sup>, which generates videos of browsing sessions, but we feel it would be more valuable to store low-level events directly in the search log for further analysis.

We have tested a variation of this platform by posting dozens of simple search tasks on AMT (such as “What is the record for the fastest mile run?” and “Who is the president of Harvard University?”), with a proxy frame included in the task, and successfully captured search logs from users on a variety of browsers. Two preliminary observations can be made, relevant to running these kinds of experiments on AMT. First, AMT workers copy-and-paste heavily, in order to work as efficiently as possible. As a result, the first query in many logs is the exact wording of the question, copied directly from the task frame into the search box. In experiments, it may be desirable to inhibit direct copying by presenting the task as an image, rather than as text. Second, a few workers answered the task without generating any search log, suggesting either that they already knew the answer or that they searched for it outside the proxy frame (contrary to the instructions of the task). This problem could be addressed by requiring use of the proxy frame before the answer can be submitted – e.g., by requiring that some part of the answer be selected, copied, and pasted from the proxy frame, which can be observed by selection events.

#### 5. OPEN DESIGN ISSUES

We are currently in the process of completing the design for our search self-efficacy studies, and while we have determined there is much to be gained from using AMT to conduct this study, we have some open design issues to address.

##### 5.1 Screening

We would like to include web searchers with both low and high self-efficacy in our study. AMT has the advantage of allowing a very large potential pool of study participants. We can administer the search self-efficacy scale as a screening tool, then administer our search assistance experiment to a stratified sample of users at different levels of search self-efficacy, ensuring that we screen sufficient numbers of users

<sup>1</sup><http://www.userfly.com>

<b>Task:</b>	<b>Answer:</b>
<input style="width: 100%; height: 30px;" type="text"/>	<input style="width: 100%; height: 30px;" type="text"/>
(a.) The task pane	Next

(b.) The proxy frame

1. Re-writes links to pass back through the proxy
2. Injects JavaScript calls to functions in the outer window
  - this can keep track of, with time stamps:
    - + mouse movements
    - + queries entered
    - + pages clicked
    - + pages viewed
    - + scrolling
    - + anything else that JavaScript can capture

Figure 2: The proxy setup for the study.

to find enough at low self-efficacy levels. However, from our initial investigations, the range of search self-efficacy seen in Turkers appears to be similar to those of the undergraduate population [8].

## 5.2 Time to Completion

Time to completion raises questions of several kinds. One is consideration of the hourly wage being paid. An important ethical question we may want to examine is how much people are being paid for their work [9]. If the median time to complete a questionnaire is used as a marker, the hourly wage paid for the two HITs issued for this study were \$15.31 and \$1.95 per hour, respectively. The latter was an alarming number, and so we paid a bonus of \$0.17 to each worker to increase the median rate to \$8.55 per hour, the highest minimum wage in the United States as reported by the U.S. Department of Labor<sup>2</sup>.

The second concern with time to completion is how much attention users give to their answers. For example, is the Turker answering the self-efficacy part of the questionnaire in 28 seconds actually reading the questions, or just filling it in arbitrarily as quickly as possible. This point is related to “instrument reliability”, our next design issue.

## 5.3 Instrument Reliability

The questionnaire we presented to the Turkers contained only questions for which a high score means high self-efficacy. This does not provide us with any error checking. As mentioned in Section 4, Turkers have shown a tendency to complete tasks as efficiently as possible, which may include minimizing mouse movements. This means that one way to complete the questionnaire is simply to select items by location on the screen. We intend to experiment with different question wording in our full study. One technique that potentially enables identification of people who may not be engaging in depth with questions on surveys has been to provide questions with both positively and negatively phrased versions [2]. In our own work, this approach has allowed us to identify and filter out survey respondents whose answer profiles suggest they are selecting options so as to optimize time-to-completion and are therefore unlikely to be providing useful data [4].

<sup>2</sup><http://www.dol.gov/whd/minwage/america.htm>

## 5.4 Truthfulness

While self-efficacy relates only to self-perception and not performance, we would like users’ honest opinions about their self-efficacy. If Turkers view the questionnaire as a screening mechanism akin to a job interview, they may be incentivized to report higher self-efficacy than they truly feel. This is clearly an issue with all surveys of this kind, where participants often has a sense of what are “desirable” responses [2]. One way for us to address this is again through a slightly different phrasing on the questions such that desirable responses are not so clearly implied by the context (e.g., search ability is clearly a good skill to have and strongly aligned with being online—so Turkers likely skew towards seeing search prowess as desirable).

## 6. NEXT STEPS

The work described here gives us the ingredients needed for our full study. Our next steps are to complete the study using the following design:

- Modify the search self-efficacy scale so we can estimate reliability
- Screen Turkers with cross-checked search self-efficacy assessments to create a stratified sample by search self-efficacy
- Integrate search assistance mechanisms into the Turker search logging platform
- Design a post-survey about the level of task difficulty
- Evaluate the effects of search assistance, taking into account searcher self-efficacy and task difficulty

## 7. REFERENCES

- [1] P. G. Anick. Using terminological feedback for web search refinement: a log-based study. In *SIGIR*, pages 88–95. ACM, 2003.
- [2] A. Bandura. Self-efficacy: Toward a unifying theory of behavioral change. *Psych. Review*, 84(2):191–215, 1977.
- [3] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *Proc. of WWW*, pages 1–10, 2009.
- [4] C. Cheshire, J. Antin, and E. Churchill. Behaviors, adverse events and dispositions: An empirical study of online discretion and information control. *JASIST*, 61(7):1487–1501, 2010.
- [5] D. Compeau and C. Higgins. Computer self-efficacy: Development of a measure and initial test. *MIS Quarterly*, 189(211), June 1995.
- [6] H. Feild, R. Jones, and J. Allan. Predicting searcher frustration. In *Proc. of SIGIR*, 2010.
- [7] A. Hassan, R. Jones, and K. L. Klinkner. Beyond DCG: user behavior as a predictor of a successful search. In *Proc. of WSDM*, pages 221–230, 2010.
- [8] D. Kelly. A preliminary investigation of search self-efficacy. Technical Report TR-2010-01, U. of North Carolina School of Information and Library Science, 2010.
- [9] K. Mieszkowski. “I make \$1.45 a week and I love it”. *Salon.com*, 2006.

# Crowdsourcing a News Query Classification Dataset

Richard M. C. McCreddie  
Department of Computing  
Science  
University of Glasgow  
Glasgow, G12 8QQ  
richardm@dcs.gla.ac.uk

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

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

## ABSTRACT

Web search engines are well known for aggregating news vertical content into their result rankings in response to queries classified as news-related. However, no dataset currently exists upon which approaches to news query classification can be evaluated and compared. This paper studies the generation and validation of a news query classification dataset comprised of labels crowdsourced from Amazon's Mechanical Turk and details insights gained. Notably, our study focuses around two challenges when crowdsourcing news query classification labels: 1) how to overcome our workers' lack of information about the news stories from the time of each query and 2) how to ensure the resulting labels are of high enough quality to make the dataset useful. We empirically show that a worker's lack of information about news stories can be addressed through the integration of news-related content into the labelling interface and that this improves the quality of the resulting labels. Overall, we find that crowdsourcing is suitable for building a news query classification dataset.

**General Terms:** Performance, Experimentation

**Keywords:** News Query Classification, Crowdsourcing, Vertical Search

## 1. INTRODUCTION

General-purpose Web search engines are well known for integrating focused news vertical content into their search rankings when the user query is judged as holding some news-related intent [8, 12]. In particular, every user query submitted is classified as either having a news-related intent or not. If so, the Web search engine will aggregate appropriate news content into the search ranking returned. However, while it is likely that commercial search engines have large internal datasets for evaluating their news query classification performance, there currently exists no publicly available dataset upon which news query classification approaches can be evaluated and compared.

On the other hand, *crowdsourcing* [13] has been championed as a viable method for creating datasets both quickly and cheaply, whilst still maintaining a reasonable degree of quality [1]. We hypothesise that crowdsourcing is a suitable means to generate a news query classification dataset, which will be useful when investigating news query classification using evidence from various sources, e.g. the full query-log. In this paper, we detail the generation and validation of a such a dataset comprised of real user queries from a search engine query log and associated news classification labels crowdsourced using Amazon's Mechanical Turk.

In our study, we follow an iterative design methodology, as recommended in [2], during the creation our news query classification

dataset. We propose multiple interfaces for crowdsourced query labelling and evaluate these interfaces empirically in terms of the quality of the resulting labels on a small representative sample of user queries from a Web search engine query log. Later, we use the best performing of these interfaces to generate our final news query classification dataset comprised of a larger query sample from the same log. We report the quality of our resulting news query classification dataset in terms of inter-worker labelling agreement and accuracy with regard to labels created separately by the authors. Moreover, we further investigate its quality in the form of an additional agreement study, in which crowdsourcing is leveraged for quality assurance.

Notably, one of the most interesting aspects of news query classification labelling is the temporal nature of news-related queries [16]. In particular, a query should only be labelled as news-related if there was a relevant noteworthy story in the news around the time each query was made. However, the query log we employ dates back to 2006 [7], hence there is no guarantee that our workers will remember what the major news stories were from the time of each query. In this work, we empirically investigate the effect that this has on labelling quality. Moreover, we propose the integration of news headlines, article summaries and Web search results into the interface seen by the workers to address this problem.

The main contributions of this paper are four-fold: firstly, we examine the suitability of crowdsourcing for the creation of a news query classification dataset; secondly we both propose and evaluate methods to overcome the temporal nature of news queries described above; thirdly, we investigate a novel application of crowdsourcing as a quality assurance tool; and lastly we propose some best practices based on experience gained from creating this dataset.

The remainder of the paper is structured as follows. In the next section, we further motivate the need for a news query classification dataset as well as describe related work in news-aggregation. Section 3 provides a brief background into crowdsourcing in general, as well as recent studies that have used crowdsourcing. In Section 4, we define the methodology used to create our news query classification dataset, discuss query sampling and describe the interfaces evaluated later. Section 5 covers our experimental setup. In Section 6, we empirically determine the best of our proposed interfaces, report on the quality of the news query classification dataset produced and provide some best practices when crowdsourcing. Finally, in Section 7, we provide concluding remarks.

## 2. NEWS AGGREGATION

News aggregation is an important problem in Information Retrieval (IR), with as much as 10% of queries possibly being news-related [5]. Moreover, classifying queries as news-related or not is a challenging task. In particular, the news-relatedness of a query is not solely dependent upon the terms it contains, but also the main news stories of the time. Hence, two identical queries made at dif-

ferent times may not always receive the same label. For example, the query ‘ash’ normally would not be seen as being news-related, since the dominant interpretation is the rock band with the same name. However, in April 2010 the query suddenly became so, as an ash cloud grounded aircraft in Europe and the United States. Furthermore, there are also notable challenges when classifying queries as news-related or not soon after a story breaks. Indeed, as news-related queries are now being submitted to Web search engines mere minutes after a newsworthy event occurs [19], new approaches need to be developed which can correctly classify such ‘breaking news’ queries [16]. Hence, the need is clear for a dataset as a basis upon which to investigate these classification challenges.

In addition, news aggregation has recently become a somewhat hot topic in IR. In particular, Arguello *et al.* [3] investigated the construction of offline classifiers for vertical content in the presence of user feedback, while Diaz *et al.* [8] investigated the online aggregation of news content using explicit user feedback. However, these studies use private datasets to evaluate performance.

Our goal is to produce a standard dataset such that news query classification approaches both old and new can be easily evaluated and compared. We examine the suitability of crowdsourcing to build such a dataset from a query log provided by a real Web search engine. In the following section, we provide a brief background into crowdsourcing and motivate its application for creating a news query classification dataset.

### 3. CROWDSOURCING

In this work, we propose to build a news query classification dataset using crowdsourcing [13]. Crowdsourcing is an attractive option for researchers and industry alike as a method for dataset generation. In particular, simple repetitive *jobs*, e.g. query labelling, can be completed at a relatively small cost, and often very quickly [1]. 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 [4], the lack of motivation for workers due to below-market wages [6] and susceptibility to malicious workers [10]. In general, the advantages of generating a news query classification dataset using crowdsourcing are easily quantified. Indeed, the total cost of the experiments reported in this paper is less than \$200, with even the longest single job taking less than 3 days to complete. Still, the quality of the resulting labels may be questionable, due either to insufficient worker understanding of the job given to them, or a lack of important information needed to complete the job satisfactorily.

In this work, we use CrowdFlower, which is an on-demand labour website providing job creation, monitoring and analytical services on top of crowdsourcing marketplaces, most notably Amazon’s Mechanical Turk (MTurk). MTurk is a service that can provide real human judgements for a variety of simple repetitive jobs. In particular, the crowdsourcer defines a *human intelligence task* (HIT), where a *worker* views an interface containing instructions on how to complete the HIT, typically along with some content to be processed, and then uses the same interface to provide feedback to the system. Workers are normally paid a small sum of money or micro payment for each HIT they complete.

Notably, there have been multiple studies into crowdsourcing with MTurk to date, which provide useful information for those wishing to use crowdsourcing for IR-related tasks. In particular, Snow *et al.* [21] and Callison-Burch [6] investigated the accuracy of labels generated using MTurk within a natural language processing context. They concluded that ‘expert’ levels of labelling quality can be achieved by having multiple workers complete each crowdsourcing job and taking a majority label. Indeed, for the experiments reported in this paper, we require that three individual workers label each query, taking the majority result.

Date	Time	Query	Query ID
2006-05-01	00:00:08	What is May Day?	37afe7af832649d2
2006-05-08	14:43:42	protest in Puerto rico	71ddb381f574410e

**Table 1: Two example queries from the MSN Web search query log for May 2006.**

Kittur *et al.* [14] also examined labelling quality when moderating Wikipedia pages. Their results highlight the need to validate the output produced by each worker, pointing out that some workers produce random or malicious labels. Following this recommendation, for both query samples used in this paper we create ground truth labels to vet our workers’ output in an online manner. In this way, we can detect and then eject poorly performing workers from our jobs early on in the evaluation, saving us money and hopefully improving the quality of the final labels produced. Indeed, we later empirically show through experimentation the effect that validation has on the quality of our resulting labels.

Lastly, Alonso *et al.* [1] tackled the related task of building a TREC-style ad-hoc test collection using crowdsourcing. Our work differs from this, in that we propose to have workers label queries as news-related or not, instead of labelling a document’s relevance to a query. Moreover, our news query classification task also has a novel temporal component which needs to be addressed, i.e. that a query’s news-relatedness is dependent not only on the terms it contains, but also on the news stories that were important at the time the query was issued. Furthermore, unlike in [1], we also perform an empirical analysis of the produced labels’ quality.

Taking this prior work into account, in the next section, we detail the methodology and data used to generate our news query classification dataset. Most notably, the creation of our two query sets, including validation queries, and the five job interfaces that are used during evaluation.

### 4. METHODOLOGY

The task that we address in this paper is the creation of a general, high quality dataset to evaluate approaches to news query classification. In particular, the dataset in question is comprised of a set of queries, each to be labelled as holding a news-related intent or not. The performance of any news query classification approach can then be evaluated on how well it performs against the dataset. In this work, we use real queries sampled from American users of the MSN Search engine (now Microsoft Bing) query log from 2006 [7], while we propose to generate the news query classification labels through crowdsourcing. Two example queries from the MSN query log are shown in Table 1. In particular, for each user query  $q$  from the set of all queries to be included in the dataset, we want our crowdsourced workers to classify  $q$  as holding a news-related intent or not for the time  $t$ , i.e. the time the query was made.

However, there are some important challenges that we need to address during the crowdsourcing of our dataset to ensure that the resulting labels are of a high enough quality for the dataset to be useful. Firstly, we identify the possibility of workers choosing labels in a random or malicious manner, which we propose to address in two ways, namely through having multiple workers label each query in addition to worker validation. Secondly, we also note that our workers’ lack of information about the main news stories from the time of the query may hinder labelling quality. We propose to mitigate this by integrating news content from the time of the query into the job interface. Indeed, we empirically examine the integration of both news article content, i.e. headlines and news summaries, and Web search engine result rankings into the job interface, to determine which best overcomes the workers’ lack of information and so provides the highest quality labels.

In order to investigate these challenges, and indeed our proposed solutions to them, we follow an iterative design methodology [2].



In particular, we begin by creating a small set of queries of approximately  $\frac{1}{10}$ th of the final desired dataset size for testing purposes. This query set is referred to as the *testset*. This testset is advantageous as it allows us to much more cheaply and quickly investigate the challenges described earlier. Indeed, it is important to note that prototyping, while a valuable tool when crowdsourcing, can dramatically increase the total cost of the task one wishes to address. Using the testset, we empirically evaluate the effect of validation using a baseline interface and subsequently test our proposed alternative interfaces integrating news content in an iterative manner.

Having determined the most effective job design, we then create a full size query sample, denoted the *fullset*, and crowdsource labels for it, hence creating our final dataset. We lastly evaluate the quality of this dataset in terms of inter-worker labelling agreement, accuracy with regard to labels created manually by the primary author of this paper and also in the form of a meta-agreement study performed using crowdsourcing. In the remainder of this section, we describe our query sampling and worker validation approaches, in addition to the interfaces used in later experiments.

## 4.1 Query Log Sampling

Recall that we propose to use two sets of queries during our experimentation: a small query set, which we refer to as the testset, as well as a larger set of queries to be included in our final dataset, denoted the fullset. Importantly, the MSN Search engine query log [7] contains almost 15 million real user queries spread over the course of May 2006, which is many times the size of either of our desired query sets. Indeed, such a large query set could not be exhaustively labelled within a reasonable time-frame, even with crowdsourcing. Instead, we propose to sample the MSN query log to create our two query sets.

Notably, there are two desirable properties that we wish our sampled query-sets to hold. Firstly, we wish our samples to be *representative* [18]. This means that the sampling method chosen should maintain the statistics of the query log as a whole. Secondly, our samples need to be *unbiased* [17], i.e. every query within the query log should have an equal chance to be selected. Should the resulting sample lack either of these properties, our final dataset will be of limited use, as the dataset would not represent the querying behaviour of real users.

A well-known and straightforward sampling strategy is random sampling [22]. In random sampling, the query log is considered a ‘bag’ of queries. Queries are then iteratively selected at random from the query log without replacement. Notably, random sampling is unbiased. Indeed, each query has an equal chance of being selected. However, in practice, random sampling often produces an unrepresentative sample, as there is no guarantee that the selected queries will be spread over the entire log [17].

An alternative sampling strategy that has proved popular is known as systematic sampling [17]. Here, the query log is considered as a time-ordered stream, where queries are iteratively selected based upon a time interval. For example, given a time interval of three minutes, one query will be selected for every three minutes of log. A systematic sampling approach is advantageous, in that a fairly representative sample of the query log will be produced, with sampled queries being spaced evenly across the time-range of the log. On the other hand, systematic sampling is not unbiased, as the probability of a query being selected is independent from the density of queries within each time interval [18]. Hence, a query within a very dense time interval has a much lower probability of being selected than one from a sparser time interval.

In light of the drawbacks of these two prior approaches, the Poisson sampling strategy has been proposed as a means to create both an unbiased and representative sample [18]. In particular, Poisson sampling also treats the query log as a time-ordered stream.

---

### Poisson Sampling - Pseudo-Code

---

```

1: Input
   query log: a temporally ordered stream of queries
    $\alpha$ : a parameter to control global the sampling rate
   querylogsize: the number of queries in query log
    $\mu$ ,  $\varpi$  and  $\varsigma$ : parameters to control the sampling rate over time
2: Output
   query-set: a set of sampled queries
3: Integer numToSkip = 1000
4: Integer pos = 0
5: Integer skipped = 0
6: for each query q in query log
7:   if skipped == numToSkip
8:     add q to query-set
9:     Integer newNumToSkip = 0
10:    double[] values
11:    while values[newNumToSkip]  $\leq \exp(\text{numToSkip})$  loop
12:      newNumToSkip = newNumToSkip + 1
13:      values[newNumToSkip] = rand(0,1)*values[newNumToSkip-1]
14:    end loop
15:    Double  $\beta = \cos(((\text{pos}/\text{querylogsize}) * \mu) + \varpi) + \varsigma$ 
16:    numToSkip = numToSkip *  $\alpha * \beta$ 
17:    skipped = 0
18:  end if
19:  skipped = skipped + 1
20:  pos = pos + 1
17: end loop

```

Figure 1: Pseudo-code interpretation of Poisson sampling.

Queries are then sampled probabilistically in an iterative manner based upon the density of queries within the current time interval. The idea is that, for a fixed time-interval, the number of queries sampled should reflect the query density, such that each query has an equal chance to be selected. For example, during a dense section of the query log, the probability of being sampled will be higher, such that the probability of sampling any query remains constant over time. Hence, we choose Poisson sampling to sample both query-sets used in this paper. The pseudo-code of our implementation of this sampling strategy is shown in Figure 1.

Notably, to control the overall rate at which we sample the log under this approach, and hence determine the final sample size, we introduced a parameter  $\alpha$  on the distance between sampled queries, referred to as *numToSkip* in Figure 1. Furthermore, we note that queries near the start and end of the query log might be less useful when using the final dataset to evaluate approaches to news query classification. In particular, approaches that leverage evidence from other temporally close queries from the log [8] may be unfairly penalised due to a lack of available surrounding queries. As such, we favour our sampling towards the centre of the query log using a second parameter  $\beta$  on *numToSkip*. Specifically, the  $\beta$  value is dependent of the current position in the query log. *numToSkip* will be increased for queries near the start and end of the query log, hence the number of sampled queries will be less, while *numToSkip* is decreased for queries near the centre of the query log, thereby increasing the number of queries sampled. The exact distribution of  $\beta$  values is defined in terms of three parameters,  $\mu$ ,  $\varpi$  and  $\varsigma$ , which were set experimentally to 3, 1.6 and 1.5 respectively, such that a  $\beta$  distribution with the above properties was observed.

Using this sampling method, we created the two aforementioned query-sets, i.e the testset and the fullset, from the MSN query log. In particular, through experimentation we selected an  $\alpha$  value of 3 to create the fullset, resulting in a sample of just over 1000 of the 15 million original queries. However, we also noted that when creating the testset, the resulting sample was very sparse, i.e. only around 3 queries per day were sampled on average in our tests. Moreover, we later evaluate some interfaces that include news-related content for the day of a set of queries. To ensure that jobs have enough queries for a given day, we instead limit our Poisson sampling approach

	MSN query log	fullset	testset
Time Range	01/05 to 31/05	01/05 to 31/05	15/05
Number of Queries	14,921,286	1206	91
Mean Queries per Day	481,331	38.9	91
Mean Query Length	2.29	2.39	2.49

**Table 2: Statistics for the MSN query log from 2006, as well as the sampled testset and fullset.**

	fullset	testset
Time Range	01/05 to 31/05	15/05
Number of Queries	61	9
Mean Queries per Day	1.97	9
% of Target Query-set	5%	10%
News-Related Queries	47	5
Non-News-Related Queries	14	4

**Table 3: Statistics for the validation sets created for the testset and fullset.**

to only those queries from a single day. In particular, we use May 15th, which is the middle of the query log. In this way, we create a testset which is representative of a single day only, but where each job contains enough queries for it to be realistic. The assumption we make is that worker labelling performance will be similar for different days. With this restriction in place, we create our testset using an  $\alpha$  value of 2, resulting in a query-set of approximately  $\frac{1}{10}$ th the size of the fullset. Statistics for both the query-sets created are shown in Table 2.

## 4.2 Validation of Worker Labelling

Earlier, we raised the possibility that workers might label our queries in a random or malicious manner [14]. Indeed, it is logical for a worker to try to maximise their profit while minimising the effort required if there is no penalty in doing so [10]. This is especially acute in our case, as the binary labelling jobs that we ask our workers to complete can be easily and quickly accomplished by selecting labels at random. Hence, to address this, we propose to perform online validation of our worker labels against a set of ground-truth query-label pairs. The idea is that should a worker try and ‘game’ our system by randomly labelling queries, their accuracy on the ground-truth queries will be low. As such, we can identify and then eject those workers from our jobs. This is advantageous not only because ‘bad’ worker labels can be ignored, thereby improving the quality of the resulting dataset, but workers ejected on such grounds are left un-paid.

To perform this type of validation, we create one validation set comprised of ground-truth queries and associated labels for each of the two query-sets previously described. As recommended by CrowdFlower, which supports this form of validation, we create validation sets of between 5% to 10% of the target query-set size<sup>1</sup>. Notably, when selecting validation queries, it is important to consider the background probability of queries belonging to each class. For example, in our case, at best only 10% of queries might be news-related [5]. Therefore, if we choose a representative distribution for our validation set, then just by labeling each query as non-news-related a worker would achieve 90% accuracy. Instead, we favoured our validation sets toward the news-related class, thereby forcing our workers to pick out the validation queries from the predominantly non-news-related background queries. Statistics for our two validation sets are shown in Table 3.

## 4.3 Query Labelling Interfaces

Recall that the job that we want our workers to complete is the labelling of the queries in our aforementioned query sets as news-

<sup>1</sup><http://crowdfLOWER.com/docs/gold>

## News Query Classification [Test][B1]

### Instructions hide

The aim of this job is to classify a set of Web search queries as being news-related or not for a specific day. This job will display real user queries for the month of May 2006.

You will be shown a set of real user queries and the date when they were made and you need to classify them as being news-related or not, i.e. if this query had been entered into a Web search engine should the engine have included news-related content, e.g. news articles, into the ranking?

Note: This is one of many test jobs where we are examining various types of interface to see which perform the best.

**Figure 2: The basic instructions shown to our workers for each job.**

Query : Chris Daughtry

Date : 15/ 05/ 2006 06: 10: 26

**Is the query news related?** (required)

Yes

No

Should news content be displayed for this query?

**Figure 3: The basic interface with which our workers label each query.**

related or not. However, due to the temporal nature of news, the time at which each query was made must also be considered. In particular, a query should only be labelled as news-related if there was also a relevant noteworthy story in the news at the time the query was made. Indeed, even though two identical queries from different times may be judged, there is no guarantee that they should be assigned the same label. To this end, we need to design an interface comprised of two distinct components, namely: a set of instructions explaining the job the worker is to complete; and a labelling interface with which the user labels each query.

With this in mind, we designed our basic job interface as shown in Figures 2 and 3. In particular, the instructions we provide to each worker are presented by Figure 2. Notably, the first paragraph is a summary of the job and is displayed to workers searching for available jobs to complete. Hence, it is important that this succinctly describes what the worker will be asked to do. The second paragraph further clarifies the meaning of news-relatedness, as news-relatedness can be subjective in nature. For example, a worker might label the query ‘ipad sales’ as being news-related at the time this paper was written, as there was a news story indicating that the Apple iPad had sold over 2 million units. However, a worker uninterested in technology or holding anti-Apple views might label it otherwise.

Figure 3 shows the labelling interface with which our workers will interact during a job. Importantly, this labelling interface is replicated for each query. In particular, the workers are shown a query in addition to the date and time it was made, and asked to judge that query as being news-related or not. This interface combination is referred to as *Basic*.

However, earlier we identified our workers’ lack of information about the news stories from the time of the queries as a factor which might hinder labelling quality. Indeed, we hypothesise that our *Basic* interface, as described above, will be insufficient to garner high-quality judgements, as our workers lack the information needed to accurately judge queries as news-related or not. Furthermore, the query log that we use dates from 2006, hence, our workers will likely remember little from that far back. Moreover, the nature of crowdsourcing, in particular the lack of worker motivation, makes it unlikely that workers will independently attempt to address this, e.g. by searching news archives.

To address this issue, we propose to incorporate news-related content from the time of the queries into our interface design. Our intuition is that by providing the workers with information about the news stories from around the time of the query, the workers will be able to make better informed judgements, hence increasing

Headline:  
32 Dead in Iraq Attacks

Summary:  
Two suicide car bombers tore into a checkpoint for Baghdad's airport, killing at least 14 people and wounding 16. It was the first bomb attack in nearly a year aimed at the airport, and the worst in a spree of violent assaults that left at least 32 people dead on Iraq's deadliest day in weeks.

Figure 4: Example headline and news summary extracted from a New York Times article.

Query : Chris Daughtry

Date : 15/ 05/ 2006 06: 10: 26

News Headlines

- 1) Diverse Sri Lankan City Mired in Ethnic Violence
- 2) China Reacts to Scientist Fraud
- 3) 32 Dead in Iraq Attacks
- 4) Pyramid Fever in Bosnia
- 5) Prime Minister Tony Blair says he is contemplating changes in Britain's human rights laws
- 6) Israel Kills 6 Palestinians
- 7) Nepal Moves to Limit Monarchy
- 8) Somali Militias Observe Truce
- 9) U.S. Plan to Deploy Guard At Border Worries Mexico
- 10) Old Allies Criticize Republicans
- 11) Top Aide Defends N.S.A. Spying
- 12) Webb Wins Michelob Open

Is the query news related? (required)

Yes

No

Should news content be displayed for this query?

Figure 5: Examples of additional content added to the labelling component of the *Headline Inline* interface.

labelling quality. In particular, we identify two sources of news-related content that our workers might find informative, i.e. news articles (headlines and/or news summaries) and Web search results.

Initially, to supplement our interface with news article content, we use a collection of news headlines and article summaries from the time of the query log, as provided by the New York Times. An example headline and associated news story summary is shown in Figure 4. It is likely that the more relevant news-related information our workers have available the better they can judge each query. For instance, providing the summary in addition to the headline might enable our workers to identify queries that do not contain terms in the headline but were related to the news story. Consider, for example, the news summary shown in Figure 4. A worker might be able to label the query 'baghdad airport' as news-related having read the news summary but would be unlikely do so from the headline alone.

However, we suspect that the amount of useful information we can provide our workers with is limited, as overloading the interface with surplus irrelevant content will cause our workers to either ignore it or reject the job in the first place. To test this, we experiment using three alternative interfaces, varying the level of supporting information provided.

In the first interface, we include 12 news headlines, like the one shown in Figure 4 within the job instructions. This provides the worker with some news story context from the time of the queries being labelled. The resulting interface is denoted *Headline*.

For our second interface, we test the effect that the level of detail of information provided to our worker has on labelling quality. In particular, in addition to the headlines used in the above interface, for each of those headlines, we also included its associated news story summary. This interface we denote *Headline+Summary*.

However, we are concerned that our workers might not read the headlines provided in the instructions. So, for our third interface, instead of including the headlines within the instructions, we moved them into the labelling interface for each query. An example of this interface is shown in Figure 5. Notably, the headlines will always be onscreen at the time the worker makes each judgement, hence making it more likely that our workers will refer to these headlines when judging. We denote this interface *Headline Inline*.

As an alternative to providing news headlines, we examine the usefulness of incorporating Web search results. In particular, we

Query : Chris Daughtry

Date : 15/ 05/ 2006 06: 10: 26

Link 1 : [Click here](#)

Link 2 : [Click here](#)

Link 3 : [Click here](#)

Is the query news related? (required)

Yes

No

Should news content be displayed for this query?

If Yes, supply a link to a search result that supports your decision

for example a news article from a known newswire provider like CNN, BBC, NYTimes etc.

Figure 6: Examples of additional content added to the labelling component of the *Link-Supported* interface.

propose to provide automatically generated search links to the three top search engines, i.e. Bing, Google and Yahoo!, for each query and time. Each link initiates a search by the associated search engine. The query for each search constitutes the original user query and the date that the query was submitted, e.g. '4th May 2006 Moussaoui Verdict'. In this way, we hope that the Web search results will be similar to those that might have been returned for the original query at the time it was made. The modified labelling interface is shown in Figure 6. Note that we do not make the target of the search link immediately clear by obscuring it behind a 'click here' anchor. We do this on the suspicion that, should we show our workers the search engines, then only the worker's favoured engine would be clicked. This interface is referred to as *Link-Supported*.

Interestingly, the *Link-Supported* interface also provided us with the opportunity to gain additional feedback from our workers. In particular, as we are requesting our workers to inform their decision based on Web search results, we can also gain useful feedback from the search result that the worker based their decision on (if any). Hence, for this interface we also ask our worker to provide the URL of the appropriate Web result. We hypothesise that this can later be used to validate the judgements through examination of the supporting URLs. Indeed, we investigate this later in Section 6.5.

## 5. EXPERIMENTAL SETUP

In this section, we detail the experimental setup for our crowdsourcing experiments. In particular, Section 5.1 defines the specific research questions that we address in our experiments, while in Section 5.2, we describe the settings we used for our crowdsourcing jobs. Lastly, Section 5.3 describes the evaluation measures used.

### 5.1 Research Questions

In Section 6, we investigate the following research questions:

1. How well do users agree on news query classification labels using our basic interface? (Section 6.1)
2. What effect does online validation of the worker labels have on overall label quality? (Section 6.2)
3. Are any of the proposed alternative interfaces effective at countering our workers' lack of information with regard to the main news-stories of the time? (Section 6.3)

Interface	Time per query	Total cost
Basic	5s	\$1.30
Headline	10s	\$4.39
HeadlineInline	10s	\$4.59
Summary	15s	\$5.56
Link-Supported	22s	\$8.78

**Table 4: Mturk estimated worker time to spend per query and the cost incurred over the 100 query testset.**

4. Is the resulting news query classification dataset of good quality? (Section 6.4)
5. Is crowdsourcing useful for post labelling quality assurance? (Section 6.5)

## 5.2 Crowdsourcing Settings

When submitting a job, there are multiple variables which determine how it is handled by MTurk. Firstly, the crowdsourcer must specify the amount paid to each worker on a per job basis. In our case, we paid workers based on the estimated amount of time it would take to judge each query at a rate of \$2 per hour. The estimated time per query and the resulting cost paid on a per-interface basis for the testset queries are shown in Table 4. Secondly, the crowdsourcer has the option to limit the worker pool based on geographical location. Since the MSN query log is predominantly American, we limited our worker pool to the USA only. To control validation, we set the validation cutoff, i.e. the level at which a worker is ejected from the evaluation, to 70% (should they get more than 30% of the validation queries wrong they are ejected from the evaluation without remuneration). Lastly, we set the level of redundancy for judging our queries at three, whereby each query will be judged by three unique workers. Note that we do not use the four redundant judgements recommended by Snow *et al.* [21], as we wish to take a majority vote to determine the final label.

## 5.3 Measures

In this work, we measure the quality of our crowdsourced labels in two distinct ways. Firstly, we measure our worker agreement, i.e. how often our workers assigned the same label to each individual query. Our intuition is that should our workers often agree about which queries are news-related, then our confidence in the labels produced increases. In particular, we employ two well-known measures for evaluating agreement in user evaluations: Free-Marginal Multirater Kappa [20], denoted  $\kappa_{free}$ ; and Fleiss Multirater Kappa [11], denoted  $\kappa_{fleiss}$ . Notably, the two Kappa agreement measures differ in the way each calculates the probability of agreement occurring by chance. Free-Marginal Multirater Kappa assumes that the chance of selecting a class is equal to one over the number of classes, i.e. 50%, while Fleiss Multirater Kappa takes into account the relative size of the classes, i.e. in our case that queries are more likely to belong to the non-news-related class. Indeed, recall that at best only 10% of queries might be news-related [5].

The second manner in which we evaluate the quality of our crowdsourced labels is against labels manually generated by the primary author of this paper. Our intuition is that the labels we generate will have a higher probability of being correct due to a longer time spent, in addition to news article access from the time of the queries. In short, we use the labels generated as an ‘expert’ ground-truth. From this, we report standard classification measures, precision and recall. To provide a combined measure, we also report overall classification accuracy.

It is worth noting that labelling was based upon knowledge of the mainstream news stories of the time. However, there exist queries which refer to news events not reported in mainstream news, e.g.

a local football match, these are sometimes known as to as ‘tail events’ [19]. However, we believe that such tail events may be more difficult for assessors, leading to disagreement between our ground truth and the workers.

## 6. EXPERIMENTS AND RESULTS

In this section, we study the suitability of crowdsourcing as a means to generate labels for our news query classification dataset. In particular, Section 6.1 investigates the initial level of agreement observed between workers when labelling queries as news-related or not. Section 6.2 examines the effect of online validation of worker labelling quality. In Section 6.3, we investigate the alternate interfaces previously described in terms of labelling quality. In Section 6.4, we evaluate the quality in terms of agreement and accuracy on our final news query classification dataset, whilst we perform an additional meta agreement study into our dataset’s quality in Section 6.5.

### 6.1 Worker Agreement

Following our methodology described earlier in Section 4, we begin by examining the workers’ labelling agreement on the smaller testset. In particular, we wish to establish that crowdsourcing is indeed suitable for labelling queries as news-related or not, before committing to a labelling job over the fullset comprised of over 1000 queries. To this end, we evaluate worker agreement when labelling the queries of the testset using the *Basic* interface described in Section 4.3. A high level of agreement indicates that the resulting labels are of good quality, hence the labelling method is suitable. Importantly, there is no de facto standard for defining acceptable or significant levels of agreement. However, Landis and Koch [15] state that Kappa values over 0.61 indicate substantial levels of agreement, while values over 0.81 represent almost perfect agreement.

Table 5 reports two Kappa agreement measures,  $\kappa_{free}$  and  $\kappa_{fleiss}$  for labelling the testset queries with our *Basic* interface. As can be observed from the table, labelling using our basic interface provides a low level of agreement -  $\kappa_{fleiss} = 0.2647$  - when worker pooling is restricted to the USA only<sup>2</sup>. This level of agreement is markedly lower than that which we would judge acceptable, hence in the following sections we examine methods to improve labelling quality in terms of agreement.

### 6.2 Importance of Validation

Previous work into crowdsourcing with MTurk has highlighted the importance of result validation [14], i.e. the checking of worker input against a ground truth to prevent random or malicious results. Hence, we examine the importance of validation for our news query labelling job. In particular, the 9 validation queries are interspersed with the 91 queries of the testset. Workers are examined in terms of the percentage of the validation queries that they correctly label.

The first two rows of Table 5 report labelling agreement for the testset queries using our basic interface. We observe that both agreement and accuracy markedly improved over our earlier baseline run which did not validate worker input. Not only does this confirm the need to validate worker input, but it highlights the scale of the issue. In particular, 32% of queries judged during this job were rejected based on our validation. Such a high level of rejected judgements might be attributed to a lack of information provided to our workers, as no additional news content is provided at this stage. However, a large proportion of these judgements were also made

<sup>2</sup>Note that we also examined pooling workers from all countries, however, performance was markedly lower, i.e.  $\kappa_{fleiss} = 0.0395$ . This likely results from the predominantly American centric nature of the MSN query log, in addition to possible malicious worker spamming from other countries.

Interface	Query Set	Validation	Precision	Recall	Accuracy	$\kappa_{free}$	$\kappa_{fleiss}$
<i>Basic</i>	testset	✗	0.5	0.5714	0.9263	0.5833	0.2647
<i>Basic</i>	testset	✓	<b>1.0</b>	<b>0.5714</b>	<b>0.9681</b>	0.7373	0.3525
<i>Headline</i>	testset	✓	0.5	0.5714	0.9263	0.8	0.5148
<i>Headline_Inline</i>	testset	✓	<b>1.0</b>	0.2857	0.9474	0.7866	0.3018
<i>Headline+Summary</i>	testset	✓	<b>1.0</b>	<b>0.5714</b>	<b>0.9684</b>	0.83	0.5327
<i>Link-Supported</i>	testset	✓	<b>1.0</b>	<b>0.5714</b>	<b>0.9684</b>	<b>0.8367</b>	<b>0.5341</b>
<i>Link-Supported</i>	<b>fullset</b>	✓	<b>0.6761</b>	<b>1.0</b>	<b>0.9748</b>	<b>0.8358</b>	<b>0.7677</b>

**Table 5: Quality measures for news query classification on the approximately 100 query testset with varying interfaces, in addition to the over 1000 query fullset using the *Link-Supported* interface.**

within the first minute of the job, at a rate far exceeding a human’s labelling ability. This indicates that there are auto-completing bots attempting jobs on MTurk, which need to be identified and filtered.

### 6.3 Supplementing Worker Information with News Content

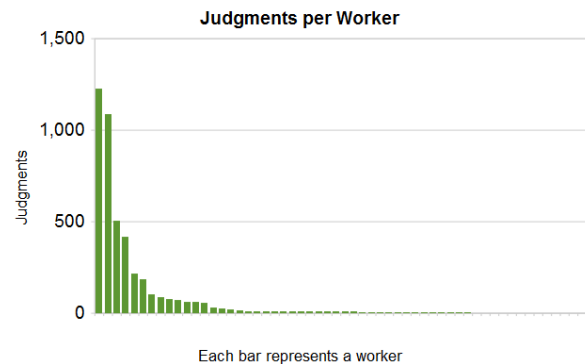
Recall that we identified the worker’s lack of knowledge about the major newsworthy stories of the time around which the queries were submitted as an important limitation on our workers. Moreover, we proposed four alternative interfaces, which provide workers with addition news-related information in an attempt to address this. In this section, we evaluate the quality of the labels produced using these alternative interfaces. Our intuition is that the use of this additional evidence will allow our workers to make informed decisions about the queries, thus improving the final label quality measured in terms of worker agreement.

Once again, Table 5 presents our workers’ labelling performance in terms of agreement for the testset queries using the four interfaces previously described in Section 4.3. As can be observed from the table, providing our workers with either news headlines (*Headline*), news summaries (*Headline+Summary*) or a ranking of search results (*Link-Supported*) from the time of the query, we can markedly increase worker agreement. However, we also noted that for the *Headline\_Inline* interface, placing news headlines prominently with each query instead of within the instructions decreased agreement between our workers. We suspect that this results from some workers only matching the query against the provided headlines, causing them to miss other newsworthy queries for which a headline was not provided. Indeed, the low recall against our manually judged query set confirms this.

Overall, we observe that our *Link-Supported* interface obtained the highest worker agreement, i.e.  $\kappa_{free} = 0.8367$  and  $\kappa_{fleiss} = 0.5341$ . Furthermore, the high overall labelling accuracy of 0.9684 obtained for this interface, in addition to the strong level of agreement shown, attests the suitability of crowdsourcing for labelling queries as news-related or not. [21] noted that in crowdsourcing there is a trade-off between the number of workers assigned to each job and the resulting quality. We note that the *Link-Supported* job cost over 6 times that of *Basic*. However, although we could conceivably have 6 times as many workers perform labelling using *Basic* for the same cost as *Link-Supported*, we would not expect accuracy with *Basic* to markedly increase as the information available to each worker remains constant.

### 6.4 Evaluating our News Query Classification Dataset

Having observed that crowdsourcing appears to be suitable for generating news query classification labels upon the 100 testset queries, we now build the our full news query classification dataset comprised of the 1206 queries from the fullset in addition to the 61 queries in its associated validation set and evaluate the quality of the resulting labels. The last row in Table 5 shows label quality for the fullset using our *Link-Supported* interface with validation. We report precision, recall and accuracy over our manually judged labels, in addition to the agreement measures  $\kappa_{free}$  and  $\kappa_{fleiss}$ .



**Figure 7: Number of judgements made by each worker when labelling the fullset queries using the *Link-Supported* interface.**

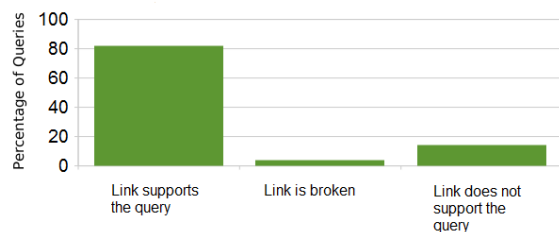
As can be observed from Table 5, over the larger query set, the accuracy of our crowdsourced labels is high, i.e. over 90%. Indeed, all of the queries judged by the author as being news-related were also labelled as such by the workers (100% recall). In comparison to our results on the testset, this recall increase indicates that the testset queries were more difficult. Reported precision on the other hand is markedly lower than on the testset. This is to be expected, as was noted earlier, there likely exist news-related queries that refer to tail events. For these queries, the workers may be more likely to disagree with our ground truth labels. Additionally, agreement between our workers is also high, indeed higher than shown on the smaller testset. This may have resulted from workers becoming more proficient at the job as they judge more queries and moreover pass a larger number of validation queries. Indeed, Figure 7 shows the number of judgements per worker. We observe that the majority (over 70%) of our judgements were completed by 3 workers, completing between 500 and 1200 queries each.

### 6.5 Crowdsourcing Additional Agreement

In the previous experiments, we used crowdsourcing as a means to label queries as news-related or not for a specific time. However, we also hypothesised that crowdsourcing could also be used as a quality assurance tool. In this section, we further evaluate the quality of our news query classification dataset by crowdsourcing additional agreement labels.

Intuitively, we could return the labels produced for our news query classification dataset to MTurk, asking a second group of workers to validate the quality of those labels under the same conditions as the original job. However, this is similar in effect to increasing the redundancy for the original job. Instead, we propose to leverage the URLs that each worker was asked to provide under the *Link-Supported* interface when labelling a query as news-related.

In particular, we ask our workers to validate each of the queries in the fullset which were judged news-related by our original workers, based upon the content of the linked Web page by that URL. Should the URL support the original worker’s label, then this increases confidence that the label is correct. Hence, the labelling quality of the query subset judged news-related can be determined by the proportion of queries within that set that are supported by their linked Web page. In particular, we ask each worker to label



**Figure 8: Percentage of URL's that were labelled as either broken, supporting the news-relatedness of the query or not supporting the news-relatedness of the query during the quality assurance job.**

the URL as either being broken, supporting the news-relatedness of the query or not supporting the news-relatedness of the query. Notably, this can be seen as a form of meta agreement, as we measure how well the new workers agree with the original workers based on evidence provided by those original workers.

Figure 8 reports the percentage of URLs that were placed in each class. We observe that the vast majority (82%) of URLs were judged as supporting the query as being news-related. This is very promising, as it shows that not only were our original workers effective at finding news-related queries, but they were making informed decisions based upon the Web search results provided. Moreover, these results further support the claim that our resulting news query classification dataset is of good quality.

Overall, we conclude that crowdsourcing is indeed suitable for labelling queries as news-related or not, as attested by the high levels of agreement between our workers, the high labelling accuracy upon manually judged labels and in terms of meta-agreement with a second set of workers. Moreover, from the experience that we gained from the creation of the resulting dataset, we suggest the following best practices when crowdsourcing.

1. **Be aware of geographical differences:** Worker performance varies by location to location. Consider from where your workers will be best able to complete your task.
2. **Online worker validation is paramount:** You need to evaluate worker performance to detect bots and poor quality workers early within the evaluation.
3. **Provide workers with as much information as possible:** Workers are not experts at most jobs. Overcome this by providing additional relevant information or external resources that workers can quickly and easily refer to.
4. **Workers can learn:** Workers are real people and can learn to become better at a job over time. This is true not only over a single large job, but equally over all the jobs submitted. Indeed, we observed that there was a notable worker overlap between our runs using the testset.
5. **Consider meta-agreement for additional validation:** Try to collect additional feedback from the workers. There may be too much data to be evaluated by hand, but crowdsourcing can be used for evaluation as well as data creation.

## 7. CONCLUSION

In this paper, we investigated building a dataset for the news query classification problem. In particular, we proposed the crowdsourcing of labels from Amazon's Mechanical Turk as a faster and cheaper method than relying on specialist annotators. Using queries sampled from a real search engine query log we experimented to determine the effect of worker validation, in addition to methods for mitigating the lack of information about the news stories from the time of the queries on the part of our workers.

We have shown that online validation of workers is paramount, and that by supplying workers with Web search rankings or related news article content for the query, we could dramatically increase labelling quality. Moreover, we have shown the suitability

of crowdsourcing for the news query classification problem as well as its application in practice. Indeed, we created a new dataset of sufficient quality, both in terms of inter-worker agreement and also against a set of manually judged queries. Furthermore, we have also examined the resulting dataset using a novel application of crowdsourcing as a quality assurance tool, showing that crowdsourcing can be useful as a means to calculate addition agreement between users and also confirming the high level of quality shown by our dataset. Lastly, based upon the experience we have gained from this study, we have provided a set of best practices to help future researchers design robust crowdsourcing experiments.

## 8. REFERENCES

- [1] O. Alonso, D. E. Rose, and B. Stewart. Crowdsourcing for relevance evaluation. *SIGIR Forum*, 42(2):9–15, 2008.
- [2] O. Alonso. Crowdsourcing for relevance evaluation. *Tutorial at ECIR'10*, Milton Keynes, UK.
- [3] J. Arguello, F. Diaz, J. Callan, and J. F. Crespo. Sources of evidence for vertical selection. In *Proceedings of SIGIR'09*, Boston, MA, USA.
- [4] J. Atwood. Is Amazon's Mechanical Turk a failure?, 2007. <http://www.codinghorror.com/blog/2007/04/is-amazons-mechanical-turk-a-failure.html>, accessed on 02/06/2010.
- [5] J. Bar-Ilan, Z. Zhu, and M. Levene. Topic-specific analysis of search queries. In *Proceedings of WSCD'09*, Barcelona, Spain.
- [6] C. Callison-Burch. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of EMNLP'09*, Singapore.
- [7] N. Craswell, R. Jones, G. Dupret and E. Viegas. In *Proceedings of WSCD'09*, Barcelona, Spain.
- [8] F. Diaz. Integration of news content into Web results. In *Proceedings of WSDM'09*, Barcelona, Spain.
- [9] R. Jones and F. Diaz. Temporal profiles of queries. *ACM Trans. Inf. Syst.*, 25(3), 2007.
- [10] J. Downs, M. Holbrook, S. Sheng, and L. Cranor. Are your participants gaming the system? Screening Mechanical Turk workers. In *Proceedings of CHI'10*, Atlanta, GA, USA.
- [11] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.
- [12] S. Hansell. Google keeps tweaking its search engine, 2008. <http://www.nytimes.com/2007/06/03/business/yourmoney/03google.html>, accessed on 08/06/2010.
- [13] J. Howe. The rise of Crowdsourcing, 2006. <http://www.wired.com/wired/archive/14.06/crowds.html>, accessed on 02/06/2010.
- [14] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with Mechanical Turk. In *Proceeding of CHI'08*, Florence, Italy.
- [15] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.
- [16] R. McCreddie, C. Macdonald, and I. Ounis. Insights on the horizons of news search. In *Proceedings of SSM'10*, New York, NY, USA.
- [17] H. C. Ozmutlu, A. Spink, and S. Ozmutlu. Analysis of large data logs: an application of Poisson sampling on Excite Web queries. *Information Processing & Management*, 38(4):473–490, 2002.
- [18] S. Ozmutlu, A. Spink, and H. C. Ozmutlu. A day in the life of web searching: an exploratory study. *Information Processing & Management*, 40(2):319–345, 2004.
- [19] J. Pedersen. Keynote speech. In *Proceedings of SSM'10*, New York, NY, USA.
- [20] J. J. Randolph. Free-marginal Multirater Kappa (multirater K free): an alternative to Fleiss' Fixed-marginal Multirater Kappa. In *Proceedings of JULIS'05*, Joensuu, Finland.
- [21] 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*, Honolulu, HI, USA.
- [22] J. S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, 1985.
- [23] E. M. Voorhees. TREC: Continuing information retrieval's tradition of experimentation. *Commun. ACM*, 50(11):51–54, 2007.
- [24] E. M. Voorhees and D. M. Tice. Building a question answering test collection. In *Proceedings of SIGIR'00*, Athens, Greece.

# Detecting Uninteresting Content in Text Streams

Omar Alonso, Chad Carson, David Gerster, Xiang Ji, Shubha U. Nabar  
Microsoft Corp.

1065 La Avenida, Mountain View, CA 94043

{omalonso, ccar, dgerster, xiangji, shubhan}@microsoft.com

## ABSTRACT

We study the problem of identifying uninteresting content in text streams from micro-blogging services such as Twitter. Our premise is that truly mundane content is not interesting in any context, and thus can be quickly filtered using simple query-independent features. Such a filter could be used for tiering indexes in a micro-blog search engine, with the filtered uninteresting content relegated to the less frequently accessed tiers.

We believe that, due to the nature of textual streams, it should be interesting to leverage the wisdom of the crowds in this particular scenario. We use crowdsourcing to estimate the fraction of the Twitter stream that is categorically not interesting, and derive a single, highly effective feature that separates “uninteresting” from “possibly interesting” tweets.

## Categories and Subject Descriptors

H.3.4 [Information Storage and Retrieval]: Systems and software — performance evaluation

## General Terms

Experimentation, classification, relevance.

## Keywords

Twitter, user study, crowdsourcing.

## 1. INTRODUCTION

Micro-blogging platforms such as Twitter and Jaiku have recently gained popularity as publishing mechanisms. Millions of users post opinions, observations, ideas and links to articles of interest in the form of status updates. Due to the decentralized and instantaneous nature of publishing on such platforms, these posts contain valuable real-time information. For the same reasons, however, we face the difficult problem of separating the wheat from the chaff. Much of what is published is trivial, of interest to only the publisher and a handful of others. How do we quickly filter out such content so that what remains is of potential interest to a wide audience?

Our motivation for studying this problem arose while building a “real-time” search engine that searches micro-blog updates for

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGIR '10, July 19-23, 2010, Geneva, Switzerland.

Copyright 2010 ACM 1-58113-000-0/00/0010...\$10.00.

real-time information on hot topics. On platforms such as Twitter, users typically generate 50 million updates (tweets) a day on average. The sheer volume of these updates necessitates a tiered index approach wherein potentially interesting updates are indexed in a smaller, more frequently accessed tier, with low-quality, categorically uninteresting updates indexed in the larger tiers. The question that arises then is: how do we identify, on the fly, which tier an update belongs to?

From a content perspective, we would like to explore if the updates appear to be relevant and if there is a class that we can call interesting or appealing based on user judgments.

In this paper, we use crowdsourcing for this exploration. We assign workers on the Amazon Mechanical Turk (AMT)<sup>1</sup> platform the task of labeling posts as “only interesting to author and friends” or “possibly interesting to others”, with the premise that no further context is needed for identifying the truly mundane. We chose a crowdsourcing approach because it is cheap and extremely fast for running these types of experiments.

Our studies bring to light certain interesting facts: 57% of the Twitter stream is categorically not interesting, and of these 89% do not contain hyperlinks. Moreover, we find that the simple presence of a link correctly classifies a tweet as “not interesting” or “possibly interesting” more than 80% of the time. This simple rule comes at a price, however, since it incorrectly classifies many tweets as not interesting simply because they do not contain a link.

## 2. RELATED WORK

Amazon Mechanical Turk has emerged as a viable platform for conducting relevance experiments. Most of the research has been on evaluating relevance assessments and comparing the performance of Mechanical Turk workers versus experts. Examples of this type of research are evaluating a subset of TREC [1] and annotator performance in four different NLP tasks [10].

There have been several recent studies on micro-blogging services. Much of the research has been focused on questions related to the structure and nature of the Twitter community. For example, the geographical and topological properties of the Twitter network are studied in [5] and [6]. In [4] and [11], the authors study motivations for using Twitter and argue that activities on Twitter can be thought of as information seeking or information sharing.

There has also been some work on semantic analysis of the textual content of Twitter updates: The authors in [8] use a partially supervised learning model to map tweets to dimensions that correspond roughly to substance, style, status and social

---

<sup>1</sup> [www.mturk.com](http://www.mturk.com)

characteristics of the updates. In [3], the authors use Twitter to track consumer sentiment towards certain brands.

The real-time nature of Twitter updates is studied and harnessed in [2] and [9]. Dong et al. [2] uses Twitter signals for url discovery and to improve the ranking of these newly discovered urls in web search results. The authors in [9] used Twitter to build an earthquake reporting system in Japan that outperforms the Japan Meteorological Agency in speed of notification.

The work that comes closest to ours is [7], wherein the authors propose several features for identifying interesting tweets; the features are not, however, experimentally validated.

### 3. EXPERIMENTS

We performed two experiments using two sets of tweets. For the first experiment, using the Twitter public timeline API<sup>2</sup> we downloaded 100 tweets in the morning and another 100 in the afternoon for five consecutive days (Monday through Friday). After each batch of tweets was downloaded, it was immediately uploaded to AMT, where workers were presented with a set of tweets and asked if the content was “interesting” or “not interesting.” Initially, we instructed workers to label tweets “interesting” if the content mentioned “specific information that people might care about” (e.g. “Another earthquake hits Haiti”). We defined “not interesting” content to include “advertisements, opinions, and trivial updates about daily life” (e.g. “Going for lunch with a friend”). Each worker was asked to label multiple tweets, and we collected five distinct judgments for each tweet. No qualification test was used, although we selected only workers having an approval rate (a reputation measure) of at least 97%. The cost of generating labels for each 100-tweet batch was less than \$3.

While analyzing the data, we realized that our instructions were unclear. We modified the instructions and defined the labels as “only interesting to author and friends” and “possibly interesting to others.” We resubmitted the batches of tweets and found that the quality of the labels improved. Figure 1 shows a large increase in scores of 0/5 or 5/5, which signify unanimous agreement among workers, and a large decrease in scores of 2/5 or 3/5, which signify disagreement.

Score	Initial Labels	Revised Labels	Change
Unanimous agreement: 0/5 or 5/5	30%	53%	+23%
Near-agreement: 1/5 or 4/5	32%	27%	-5%
Disagreement: 2/5 or 3/5	38%	20%	-19%
Total	100%	100%	

**Figure 1. Clearer Instructions Yield More Agreement Among Workers (Experiment 1)**

<sup>2</sup> [twitter.com/statuses/public\\_timeline.xml](https://twitter.com/statuses/public_timeline.xml)

For the second experiment, we sampled 1,791 tweets from our internal data system, into which we had loaded a week’s worth of status updates from the Twitter “firehose”. The agreement among workers in this experiment (Figure 2) showed a similar distribution to the first experiment (Figure 1).

Score	# of Tweets	% of Tweets
Unanimous agreement: 0/5 or 5/5	997	56%
Near-agreement: 1/5 or 4/5	483	27%
Disagreement: 2/5 or 3/5	311	17%
Total	1,791	100%

**Figure 2. Agreement among Workers (Experiment 2)**

### 4. DATA ANALYSIS

For each tweet we created a single “interestingness” score, calculated as the number of “possibly interesting to others” AMT labels divided by the total number of labels for that tweet. Each tweet received five labels from five different workers. We observed that 1) 57% of tweets scored 0/5 or 1/5 and 2) within each score band, there was a strong correlation between the fraction of tweets containing a hyperlink and the score (Table 1).

**Table 1. Distribution of Interestingness Scores and % of Tweets with Links for each Score (Experiment 1)**

Score	# of Tweets	% of Tweets	# of Tweets with Links	% of Tweets with Links
5/5	127	13%	120	94%
4/5	105	11%	82	78%
3/5	79	8%	45	57%
2/5	112	11%	50	45%
1/5	163	17%	41	25%
0/5	394	40%	17	4%
Total	980	100%	355	36%

*Read: “78% of tweets having a score of 4/5 contained a link.”*

Next we defined a class of “uninteresting” tweets having a score of 0/5 or 1/5 (shaded grey in Table 1), with the remainder classified “possibly interesting.”

We created multiple textual features, including 1) presence of a hyperlink, 2) average word length, 3) maximum word length, 4) presence of first person parts of speech, 5) largest number of consecutive words in capital letters, 6) whether the tweet is a retweet, 7) number of topics as indicated by the “#” sign, 8) number of usernames as indicated by the “@” sign, 9) whether the link points to a social media domain (e.g twitpic.com), 10) presence of emoticons and other sentiment indicators, 11) presence of exclamation points, 12) percentage of words not found in a dictionary, 13) presence of proper names as indicated



by words with a single initial capital letter and 14) percentage of letters in the tweet that do not spell out words.

We attempted to train a decision tree classifier using the above classes and features, but repeatedly found that the “has hyperlink” feature dominated. We then created a simple classifier with a single rule: if a tweet contains a hyperlink, classify it “possibly interesting”; if not, classify it “not interesting.” We were surprised to find that this single rule classified tweets with 81% accuracy (Table 2).

**Table 2. Confusion Matrix and Accuracy using Single "Has Hyperlink" Rule (Experiment 1)**

**Confusion Matrix**

Classified as →	a	b
<b>a = Not Interesting</b>	499	58
<b>b = Possibly Interesting</b>	126	297

*Read: "126 tweets whose actual class was Possibly Interesting were classified as Not Interesting."*

Accuracy	#	%
Tweets correctly classified	796	81%
Tweets misclassified	184	19%
Total	980	100%

As the confusion matrix shows, most classification errors (126 out of 184) were due to “possibly interesting” tweets being labeled “not interesting” simply because they did not contain a link. This raises the question of what features might be useful to correctly classify such tweets. Visual inspection of these misclassified tweets shows that many contain named entities (“State of the Union”, “China”) and quantities (“\$499”, “100K”).

We performed the same analyses on the second set of 1,791 tweets labeled using AMT. We were pleased to find that the distribution of interestingness scores was similar to the first experiment, demonstrating that the quality of judgments by AMT workers is high enough to create reproducible results. We noted that the accuracy of the single “has hyperlink” rule increased to 85%.

As with the first experiment, most misclassifications (149 out of 269) were due to tweets with no link being misclassified as not interesting.

To reduce misclassified tweets, we began experimenting with new textual features including the presence of named entities. We saw two ways to generate such features: 1) algorithmic entity extraction and 2) submitting tweets to AMT with instructions on the entities we seek to identify (e.g. “Does this tweet contain the name of a person, organization, or product?”).

**Table 3. Distribution of Interestingness Scores and % of Tweets with Links for each Score (Experiment 2)**

Score	# of Tweets	% of Tweets	# of Tweets with Links	% of Tweets with Links
5/5	103	6%	99	96%
4/5	140	8%	122	87%
3/5	126	7%	87	69%
2/5	185	10%	97	53%
1/5	343	19%	86	25%
0/5	894	50%	34	4%
Total	980	100%	525	29%

*Read: "87% of tweets having a score of 4/5 contained a link."*

**Table 4. Confusion Matrix and Accuracy using Single "Has Hyperlink" Rule (Experiment 2)**

**Confusion Matrix**

Classified as →	a	b
<b>a = Not Interesting</b>	1117	120
<b>b = Possibly Interesting</b>	149	405

*Read: "149 tweets whose actual class was Possibly Interesting were classified as Not Interesting."*

Accuracy	#	%
Tweets correctly classified	1522	85%
Tweets misclassified	269	15%
Total	1791	100%

Focusing on the second approach, we submitted the 126 misclassified tweets from the first experiment back to AMT and asked workers to judge what types of named entities each tweet contained. Workers were presented with one tweet and asked to judge named entities using the following categories:

- People (John Doe, Mary Smith, joedoe, etc.)
- Places (San Francisco, Germany, UK, etc.)
- Brands or products (Windows 7, Python, iPhone, etc.)
- Organizations (US Congress, Microsoft, etc.)
- Other (State of the Union, US Patent #123456, etc.)
- No. I don't see name(s).

To improve the quality of judgments, we intentionally included the “No” category to avoid workers feeling compelled to find a named entity even when one was not present. Each worker was asked to recognize entities from a single tweet, and we collected five distinct judgments for each tweet. No qualification test was used and the approval rate was 97%. We paid two cents per task for this experiment.

Table 5 shows the distribution of named entity types across these 126 tweets. (For simplicity, only the dominant entity type is

counted for each tweet; in reality, some tweets contained multiple entity types.) . 76% of the tweets had a named entity, highlighting the potential of the named entity feature.

**Table 5. Named Entity Types for 126 "Interesting" Tweets with no Links (Experiment 1)**

Entity Type	# of Tweets	% of Tweets
Person	40	32%
No entity	20	24%
Place	21	17%
Technology	21	17%
Other	10	8%
Organization	4	3%
Total	126	100%

*Read: "For 40 tweets, 'Person' was the entity type that received the most judgments."*

## 5. CONCLUSIONS AND FUTURE WORK

Using labels gathered from AMT, we learned that the presence of a hyperlink in a tweet strongly correlates to that tweet's "interestingness" score. This single "has hyperlink" feature classifies tweets with more than 80% accuracy, with most errors due to tweets without hyperlinks being misclassified as "not interesting."

The results are promising, especially given the low cost of the labels. At \$3 per 100 tweets, our 980-tweet sample from the first experiment cost less than \$30 to label, but still yielded enough information to classify tweets with high accuracy. Because the "has hyperlink" feature is so dominant, however, results may not be representative.

In addition to providing consistent high-quality labels, AMT also shows promise for creating named-entity features that are challenging to compute algorithmically. Such crowdsourced "faux features" could be useful for supervised learning experiments with a small number of labels and therefore a small number of instances. This approach could also be used to

evaluate features that are not computationally feasible today, with the goal of quantifying the value of such features if they did become available in the future.

## 6. REFERENCES

- [1] Alonso, O., and Mizzaro, S. 2009. Can we get rid of TREC Assessors? Using Mechanical Turk for Relevance Assessment. In *Proceedings of SIGIR Workshop on the Future of IR Evaluation*.
- [2] Dong, A. et al. 2010. "Time is of the Essence: Improving Recency Ranking Using Twitter Data". In *Proceedings of WWW*.
- [3] Jansen, B. J. et al. 2009. Twitter Power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*.
- [4] Java, A. et al. 2007. Why We Twitter: Understanding Microblogging Usage and Communities. In *Proceedings of SNA-KDD Workshop*.
- [5] Krishnamurthy, B., Gill, P., and Arlitt, M. 2008. A few chirps about Twitter. In *Proceedings of WOSP*.
- [6] Kwak, H. et al. 2010. What is Twitter, a Social Network or a News Media? In *Proceedings of WWW*.
- [7] Lauw, H., Ntoulas, A., and Kenthapadi, K. 2010. Estimating the Quality of Postings in the Real-time Web. In *Proceedings of SSM*.
- [8] Ramage, D., Dumais, S., and Liebling, D. 2010. Characterizing Microblogs with Topic Models. In *Proceedings of ICWSM*.
- [9] Sakaki, T. et al. 2010. Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In *Proceedings of WWW*.
- [10] Snow, R. et al. 2008. Cheap and Fast - But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of EMNLP*.
- [11] Zhao, D., and Rosson, M. 2009. How and Why People Twitter: The Role that Micro-blogging Plays in Informal Communication at Work. In *Proceedings of GROUP*.