

# Guidelines for Designing Crowdsourcing-based Relevance Experiments

Omar Alonso  
oralonso@gmail.com

## ABSTRACT

Amazon Mechanical Turk has gained a lot of attention as a viable platform for conducting information retrieval evaluations. The service is easy to use and has useful features for setting up experiments and collecting results. However, it is important to pay attention to the design of the experiment and its execution to gather useful results. In this short paper I present practical aspects to have in mind when preparing such experiments based on my own experience.

## Categories and Subject Descriptors

H.3.4 [Information Storage and Retrieval]: System and software – performance evaluation

## General Terms

Performance, experimentation

## Keywords

IR evaluation, relevance, crowdsourcing, design of experiments

## 1. INTRODUCTION

Amazon Mechanical Turk (MTurk, [www.mturk.com](http://www.mturk.com)) is an Internet service that gives developers the ability to include human intelligence as a core component of their applications. Developers use a web services API to submit tasks, approve completed tasks, and incorporate the answers into their software applications. To the application, the transaction looks very much like any remote procedure call. The application sends the request, and the service returns the results. People come to the web site looking for tasks and receive payment for their completed work. In addition to the API, there is also the option to interact using a dashboard that includes several useful features for prototyping experiments.

The individual or organization who has work to be performed is known as the *requester*. A person who wants to sign up to perform work is described in the system as a *worker*. The unit of work to be performed is called a *HIT* (Human Intelligence Task). Each HIT has an associated payment and an allotted completion time; workers can see sample hits, along with the payment and time information, before choosing whether to work on them. It is possible to control the quality of the work by using qualification tests. A *qualification test* is a set of questions (like a HIT) that the worker must answer to qualify and therefore work on the assignments. Details on the API and user interface are available in

the developer documentation [3].

Relevance evaluation is one area that MTurk is being used by companies, researches, and start-ups to test and evaluate different techniques. An example of that is the TERC crowdsourcing approach that uses an editorial approach on a larger scale [1]. The main benefits of this approach are fast turnaround, low cost, high quality, and flexibility.

MTurk provides examples of the kind of experiments that one may be interested in running; however there is little information about the practicalities of starting and conducting such studies. In this paper I describe the basic evaluation framework and propose design guidelines and operational aspects that can be used when preparing experiments for MTurk. Much of this is based on personal experience of interacting with the service as requester and worker in a wide range of tasks.

## 2. DESIGN AND OPERATIONAL ASPECTS

Designing, conducting, and analyzing experiments is an area of statistics called Design of Experiments [5]. Not surprisingly the guidelines for designing experiments can be easily adapted to our needs in the following framework:

- Design what to test
- Sample data
- Design the experiment
- Run experiment
- Collect results and analyze data

In the case of relevance evaluation, one has a good idea of what to test (e.g., new ranking technique, summarization, etc.). There are extensive sampling techniques in the literature so I won't cover this topic. Instead, I'll concentrate in the last three items of the framework: the survey design from a content perspective, an incremental sequence of steps to launch experiments, and issues that one can encounter during data analysis.

### 2.1 Survey design

This is the most important part of the MTurk experiment design: how to ask the right questions in the best way possible. The first step is to follow standard guidelines for survey and questionnaire design. A good introduction is [2] that outlines good techniques and also points to other resources. Workers interact with MTurk via a user interface so it is important to use well-known usability techniques for presenting information [3].

Having said all that, coming up with a good survey for MTurk is part art and part science so here are some items to consider when designing one.

1. Experiment should be self-contained. Avoid asking people to go to a different URL to do something and come back to perform the task. Everything that the worker needs to know should be visible.
2. Keep it short and simple. Clear instructions with examples of what you'd expect. Avoid long and dreadful questionnaires. It should be brief, clear, and concise.
3. Be very clear with the relevance task. Workers may not be IR experts so don't assume the same understanding in terms of terminology (e.g., "binary relevance", "graded relevance", etc.) and what they have to do.
4. Engage with the worker. Select interesting content that people would enjoy working on it. Avoid boring stuff.
5. Always ask for feedback (open-ended question) in an input box. It is really amazing the quality of personal feedback that you can get when you ask open questions.
6. Internationalization. Design the content so it can be easily translated to other locales if needed.

## 2.2 Execution sequence

In the following I describe a sequence of actions that can be followed as an incremental series of steps for uploading experiments. MTurk offers access via API and dashboard. For beginners, it is better to start experimenting with the dashboard first and then use the API for automation.

1. Build an experimental test case and run it in house with your teammates.
2. Incorporate feedback from step 1 and run the test case in MTurk with a small number of HITs and workers (10 queries and 3 workers). Make sure to time the experiment and gather data to answer the following questions: how long does it take to complete? Do workers understand the instructions? How much to pay is also important. You can start with \$0.01 and then increase if the task requires more effort.
3. A qualification test at this stage is probably not desirable. After a first experiment you can try to replicate the same results with a different data set.
4. A qualification test is a much better quality filter but also involves more development cycles. Same as the HIT design, you need to create a qualification test and test it with your teammates. It is important to time the experiment with and without qualification test.
5. Assuming you are using a qualification test, adjust your expiration dates and increase one order of magnitude

the objects to test (e.g., queries, documents, etc.) with a minor increase in workers.

6. Download the results and compute your desired metrics. Examine the results by hand and look for outliers. Identify cases where the answers are wrong. Maybe the instructions are not clear or the examples not representative. Adjust qualification test and/or instructions accordingly.
7. Run modified experiment with previous data set and see if the results improve. If so, then there is indication that you can now increase the data volume. If not, keep polishing instructions and qualification test.
8. Scale on data: once the experiment is working, increase the number of objects that you are evaluating on (e.g., queries, documents, images, etc.).
9. Scale on workers: increase the number of workers. Before doing this, the expiration time should be increased because it would be difficult to get more and more workers. Also make sure to check Ipeirotis's post on estimated waiting time [4].

## 2.3 Data analysis

Data analysis is pretty straightforward and is usually domain specific. One may find surprises analyzing the results, in particular when the results are too noisy or just bad quality.

In MTurk one can accept or reject work. It is important to distinguish between rejecting an assignment from the analysis phase than annoying a worker. It is better to pay the worker even when the work is not good as one expected. You can always use a new qualification test to filter out sloppy answers or pay bonuses to good workers. In the case of relevance evaluation, people don't necessary agree if a particular document is relevant or not to a topic, so the design of experiment and data analysis are crucial.

## 3. CONCLUSIONS

Crowdsourcing-based relevance evaluation using MTurk is a feasible alternative to conduct relevance experiments. I presented tips for running experiments that should be useful for practitioners trying to leverage the MTurk infrastructure.

## 4. REFERENCES

- [1] O. Alonso, D. Rose, and B. Stewart. "Crowdsourcing for relevance evaluation", *SIGIR Forum*, Vol. 42, No. 2 2008.
- [2] F. Scheuren. "What is a Survey" (<http://www.whatisasurvey.info>) 2004.
- [3] J. Nielsen. *Usability Engineering*. Morgan-Kaufmann, 1993.
- [4] P. Ipeirotis blog ([behind-the-enemy-lines.blogspot.com](http://behind-the-enemy-lines.blogspot.com)).
- [5] D. Montgomery. *Design and Analysis of Experiments*, John Wiley, 2005