# Crowdsourcing for Relevance Evaluation

**Omar Alonso  Daniel E. Rose  Benjamin Stewart**

A9.com
Palo Alto, CA
*{oalonso, danrose, bstewart}@a9.com*

**Abstract**

Relevance evaluation is an essential part of the development and maintenance of information retrieval systems. Yet traditional evaluation approaches have several limitations; in particular, conducting new editorial evaluations of a search system can be very expensive. We describe a new approach to evaluation called TERC, based on the crowdsourcing paradigm, in which many online users, drawn from a large community, each performs a small evaluation task.

## 1 Introduction

Relevance evaluation for information retrieval is a notoriously difficult and expensive task. In the early years of the field, a set of volunteer editors – often graduate students – would painstakingly read through every document in a corpus to determine its relevance to a set of test queries. This process was sufficiently difficult that only a few small test collections (Cranfield, CACM, etc.) were created.

With the advent of TREC in 1992 [9], researchers had access to test collections with millions of full-text documents. However, the scale of TREC was only possible by eliminating the notion that every document would be read and evaluated. Instead, the pooling approach was developed, in which only the top N documents retrieved by at least one of the participating systems were examined. The other factor that made TREC possible was the availability of a large number of professional assessors – retired intelligence analysts – who were paid for their work with funds from the sponsoring agencies.

While the TREC collections – and as importantly, the query sets and evaluations – have been invaluable in furthering IR research over the past 15 years, they still have some limitations. The most obvious of these is that researchers are limited to the types of IR tasks that TREC has studied. For example, if a researcher wishes to study search in a particular vertical domain area (for example, a yellow-pages-style search for local businesses) or experiment with a new search interaction paradigm (for example, collaborative search), then existing TREC collections may not help. Furthermore, despite the presence of a Web track, evaluating general Web search has unique challenges [7], which often require another approach.

For these reasons, many researchers in both industry and academia now rely on the strategy of using editorial resources to create their own new relevance assessments from scratch, specific to the needs of

the system they are testing. Many web search engines reportedly use large editorial staffs, either in-house or under contract, to judge the relevance of web pages for queries in an evaluation set. Academic researchers, without access to such editors, often rely instead on small groups of student volunteers. Because of the students' limited time and availability, test sets are often smaller than desired, making it harder to detect statistically significant differences in performance by the experimental systems being tested.

A recent article by Saracevic presents an in-depth discussion on how people behave around relevance and how it was studied [8]. Looking at the summary of the studies, the article shows that most of them include a handful of individuals. This is not a surprise, because setting up an experiment takes time and resources, people being the most important factor.

As an alternative, Joachims [6] and others have proposed exploiting user behavior as an evaluation signal. This approach allows relevance evaluation to be performed at a much larger scale and at much lower cost than the editorial method. While behavioral evaluation can be very effective in certain circumstances, it has limitations as well. It requires access to a large stream of actual behavioral data – something not always available to a researcher testing an experimental system. And, as with TREC, there are certain tasks for which it does not make sense. For example, Rose [7] points out that user click behavior cannot be used to assess the quality of a web search result snippet, since the lack of a click might indicate either a perfect snippet that satisfies the user's information need, or a poor one that fails to convey the relevance of the underlying page.

For many tasks, then, what is needed is a third approach, one that provides the customizability of the editorial approach, but on a larger scale. We propose the use of *crowdsourcing* for this purpose. Jeff Howe coined the word "crowdsourcing" in a *Wired* magazine article to describe tasks that were outsourced to a large group of people instead of performed by an employee [4]. Crowdsourcing is an open call to solve a problem or carry out a task and usually involves a monetary value in exchange for such service. Crowdsourcing has the "Web 2.0"-style attribute of increased interactive participation by large numbers of online users. But unlike user-generated content, social networks, and other popular trends, participants in a crowdsourcing ecosystem have little or no contact with each other. In particular, one worker cannot see the results of another's work.

In the remained of this document, we describe the paradigm, which we call Technique for Evaluating Relevance by Crowdsourcing (TERC) and discuss some of its advantages and disadvantages.

## 2 A Sample Relevance Experiment

Suppose that we would like to conduct an experiment to test the effectiveness of an experimental IR system. For example, our system might be designed to search for information about countries in the CIA World Factbook. For the purposes of this example, we use the well-known DCG approach for graded relevance evaluation [5].

Our first task will be to determine a set of queries and a set of results for each query. Next, we want to obtain relevance judgments for each query-result pair. As in the original DCG formulation, we'll be using a four-point scale for relevance assessment:

- Irrelevant document (0)
- Marginally relevant document (1)
- Fairly relevant document (2)
- Highly relevant document (3)

Let's say we have 50 queries and want to judge the top 50 results for each. This gives a total of 50 x 50 = 2500 query-result pairs that need to be judged. Following the crowdsourcing approach, we will distribute those judging tasks to a large group of potential assessors on the Internet. To do this, we use Amazon Mechanical Turk [1], a crowdsourcing service that is part of the Amazon Web Services platform.

## 3 Amazon Mechanical Turk

Amazon Mechanical Turk (MTurk) is an Internet service that gives developers the ability to include human intelligence as a core component of their applications [2]. MTurk is sometimes described as an "artificial artificial intelligence" system, in the sense that instead of a piece of software trying to accomplish a certain task by means of artificial intelligence, it can simply ask a human.[1]

Developers use a web services API to submit tasks to the MTurk web site, 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, searching for and completing tasks, and receiving payment for their work. In addition to the web services API, there is also the option to load work into MTurk with a set of command line tools.

Today, there are over 200,000 registered workers from over 100 countries, and millions of tasks have been completed. On a typical day there may be thousands of tasks available on a variety of topics. Recent tasks ranged from labeling images to entering "happy hour" times at bars in a resort area.

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* (although MTurk workers often refer to themselves as "Turkers" in online discussion forums). The unit of work to be performed is called a *Human Intelligence Task*, or *HIT*. Each HIT has an associated payment and an alloted completion time; workers can see sample hits, along with the payment and time information, before choosing whether to work on them. Details of the MTurk web services API and command line tools are available in the developer documentation [1]; our focus here is on how the system can be used for the relevance assessment task.

## 4 TERC: Relevance Evaluation with Mechanical Turk

Returning to our example from Section 2, we now have 2500 query-result pairs generated from our geographic IR system for which we wish to obtain relevance judgments. We will define the process of viewing one query-result pair and choosing a relevance level as a single HIT for which we will pay one cent. The task, as the workers see it, will appear as shown in Figure 1. In order to create a HIT that looks like this, we need to describe its structure in XML:

[1] The name "Mechanical Turk" comes from the 18th-century device that appeared to be a chess-playing automaton, but turned out to have a human hidden inside.

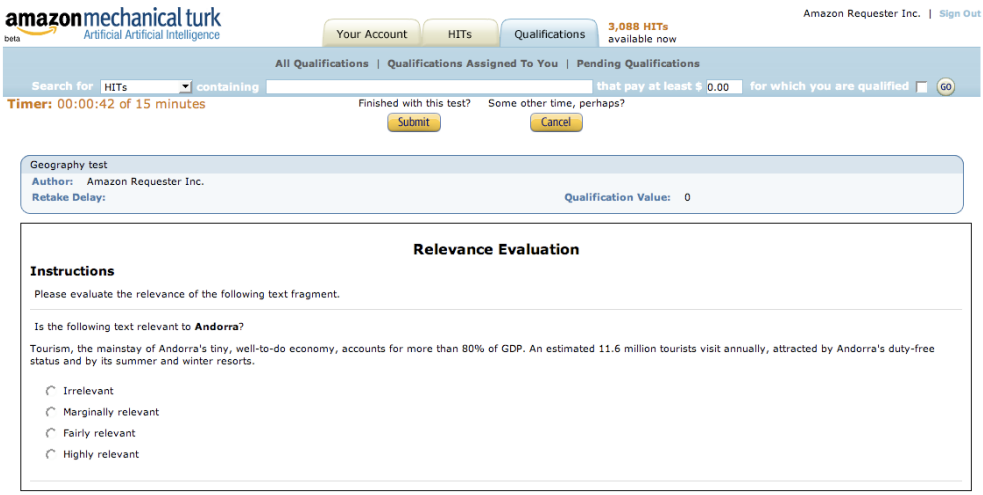


Figure 1. Task in MTurk

```
<Question>
    <QuestionIdentifier>question1</QuestionIdentifier>
    <DisplayName>Question 1:</DisplayName>
    <IsRequired>true</IsRequired>
    <QuestionContent>
      <FormattedContent><![CDATA[
       Is the following text relevant to Andorra?
       Tourism, the mainstay of Andorra's tiny, well-to-do economy, accounts for more than 80%
          of GDP. An estimated 11.6 million tourists visit annually, attracted by Andorra's duty-free
          status and by its summer and winter resorts.
      ]]></FormattedContent>
    </QuestionContent>
    <AnswerSpecification>
      <SelectionAnswer>
        <StyleSuggestion>radiobutton</StyleSuggestion>
        <Selections>
          <Selection>
            <SelectionIdentifier>ir</SelectionIdentifier>
            <Text>Irrelevant</Text>
          </Selection>
          <Selection>
            <SelectionIdentifier>mr</SelectionIdentifier>
            <Text>Marginally relevant</Text>
          </Selection>
          <Selection>
            <SelectionIdentifier>fr</SelectionIdentifier>
            <Text>Fairly relevant</Text>
          </Selection>
          <Selection>
            <SelectionIdentifier>hr</SelectionIdentifier>
            <Text>Highly relevant</Text>
          </Selection>
        </Selections>
      </SelectionAnswer>
    </AnswerSpecification>
  </Question>
```

Once we have uploaded our 2500 HITs into MTurk, workers who log into the system will see samples of these tasks and can accept the assignment to work on them. When each HIT is presented, the worker has the choice of accepting or skipping it. When each HIT is completed, the worker has the choice of stopping or continuing on to other available HITs in the set.

Requesters receive a daily report indicating how many of their HITs have been completed. Once the entire set is finished, the requesters can download the results – in our case, the relevance judgments. With this set of judgments, we can then perform our normal analysis to compute recall, precision, DCG, etc. Meanwhile, the workers are paid by having money transferred to their accounts.

## 5 Quality Control

There are two obvious concerns with using an apparently random collection of strangers to perform relevance evaluation, or any other task. First, how do we know that the workers performing the task will have the requisite skill or knowledge? Second, how do we know that they will actually make a good-faith effort to do the work, rather than clicking randomly on the responses?

The first issue is addressed by a qualification procedure built in to the MTurk framework. Requesters can create qualification tests that workers must complete before they are eligible to work on HITs. The nature of the qualification is, of course, domain dependent. If your HITs require the ability to write French, your qualification might require writing a brief essay in that language. If the task requires familiarity with rock music, a qualification test might include names of bands and songs to test the worker's knowledge. The qualification may take the form of a multiple-choice test, which can easily be scored automatically. The requester determines the value of each answer, as well as the score threshold needed to pass the qualification.

In our example, since the IR system involves articles about different countries around the world, we want our workers to know at least a little about geography. We'll use the multiple-choice approach. So, our qualification might have a few questions such as these:

- Which of these countries contains a major city called "Cairo"? (Brazil, Tunisia, Egypt, Turkey)
- Which of these is closest to the current population of India? (250 million, 500 million, 750 million, 1 billion)

On a question like the second of these, we might choose to give different numbers of points for different answers, e.g. 5 points for the correct answer and 3 points for an incorrect but plausible answer.

To address the second issue – whether workers will do the tasks correctly – the system is designed so that requesters much accept a worker's completed HITs before he or she is paid. So, if a worker completed 100 relevance assessment HITs and we found that he gave the identical answer for all of them, we could decide not to accept the work for payment.

Because relevance judgments are subjective, we will want to have more than one person judge each query-result pair. We can then use a variety of methods to aggregate the scores. For example, we might use a voting scheme, where the relevance level that receives the most votes wins. Alternatively, we might make the overall assessment be the weighted sum of the individual assessments, or require that consensus be achieved. Depending on the number of redundant judgments we obtain and the strictness of our aggregation scheme, we can insure higher levels of quality control.

It is also possible to have a hierarchy of qualification tests, in which only workers who pass one level are permitted to participate in the next. In this fashion, a requester can identify an increasingly specialized or skilled group of workers who can be paid more to do higher-quality work.

## 6 Discussion and Conclusions

After conducting several relevance experiments using MTurk, we believe the TERC approach is very promising. We have observed other search technology companies starting to use this method as well.

For example, a recent study comparing the relevance of results from four commercial search engines was conducted using the crowdsourcing approach [3].

Based on our experience, we believe TERC has several strengths:

- *Fast Turnaround*. We have uploaded an experiment requiring thousands of judgments and found all the HITs completed in a couple of days. This is generally much faster than an experiment requiring student assessors; even creating and running an online survey can take longer.
- *Low Cost*. Many typical tasks, such as judging the relevance of a single query-result pair based on a short summary, are completed for payment of one cent. (Obviously, tasks that require more detailed work require higher payment.) In our example, we could have all our 2500 judgments completed by 5 separate workers for a total cost of $125.
- *High Quality*. Although individual performance of workers varies, low cost makes it possible to get several opinions and eliminate the noise. As described in Section 5, there are many ways to improve the quality of the work.
- *Flexibility*. The low cost makes it possible to obtain many judgments, and this in turn makes it possible to try many different methods for combining their assessments. (In addition, the general crowdsourcing framework can be used for a variety of other kinds of experiments – surveys, etc.)

Of course, there are also some limitations:

- *Artificiality of Task*. As with many explicit editorial assessment tasks, the assessors do not actually have the information need that motivated the query. They are being asked to put themselves in the position of a hypothetical user. While this is largely unavoidable, the qualification test and instructions can go a long way toward making the intent clear.
- *Unknown Population*. MTurk workers may come from anywhere in the world and have any kind of background. For example, if you want to test an Italian search engine, you may want workers who know Italian culture as well as language; a college student in California may not be an adequate judge, even if she is fluent in Italian. Again, good qualification tests may help alleviate this problem.

In summary, we have described TERC, a crowdsourcing-based alternative to traditional relevance evaluation. There will always be a role in IR research for standard test collections, click-through data analysis, query logs, and user studies. We believe that the TERC approach is complementary to these methods and provides a flexible and inexpensive method for large-scale editorial relevance judgments.

## 7 References

[1] Amazon Mechanical Turk, http://www.mturk.com

[2] Jeff Barr and Luis Felipe Cabrera. “AI Gets a Brain”, *ACM Queue*, May 2006.

[3] Brendan O’Connor, “Search Engine Relevance: An Empirical Test”, http://blog.doloreslabs.com/2008/04/search-engine-relevance-an-empirical-test/#more-35, accessed April 13, 2008.

[4] Jeff Howe. “The Rise of Crowdsourcing”. *Wired*, June 2006. http://www.wired.com/wired/archive/14.06/crowds.html

[5] Peter Ingwersen and Kalervo Järvelin. *The Turn: Integration of Information Seeking and Retrieval in Context*, Springer, 2005.

[6] Thorsten Joachims and Filip Radlinski, "Search Engines that Learn from Implicit Feedback", *IEEE Computer*, Vol. 40, No. 8, August 2007.

[7] Daniel E. Rose, "Why Is Web Search So Hard… to Evaluate?" *Journal of Web Engineering*, Vol. 3, Nos. 3 & 4, pp. 171-181, December 2004.

[8] Tefko Saracevic. "Relevance: A Review of the Literature and a Framework for Thinking on the Notion in Information Science. Part III: Behavior and Effects on Relevance". *Journal of the American Society for Information Science and Technology*, 58(13):212-2144, 2007.

[9] Ellen Voorhees. "TREC: Continuing Information Retrieval's Tradition of Experimentation". *Comm. Of the ACM*, Vol. 50, No. 11, November 2007.