Why Is That Relevant? Collecting Annotator Rationales for Relevance Judgments

Tyler McDonnell
Dept. of Computer Science
University of Texas at Austin
tyler@cs.utexas.edu

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

Mucahid Kutlu **Tamer Elsayed**
Dept. of Computer Science and Engineering
Qatar University
{mucahidkutlu,telsayed}@qu.edu.qa

Abstract

When collecting subjective human ratings of items, it can be difficult to measure and enforce data quality due to task subjectivity and lack of insight into how judges' arrive at each rating decision. To address this, we propose requiring judges to provide a specific type of *rationale* underlying each rating decision. We evaluate this approach in the domain of Information Retrieval, where human judges rate the relevance of Webpages to search queries. Cost-benefit analysis over 10,000 judgments collected on Mechanical Turk suggests a win-win: *experienced* crowd workers provide rationales with almost no increase in task completion time while providing a multitude of further benefits, including more reliable judgments and greater transparency for evaluating both human raters and their judgments. Further benefits include reduced need for expert gold, the opportunity for *dual-supervision* from ratings and rationales, and added value from the rationales themselves.

1 Introduction

Ensuring data quality remains a significant challenge in crowdsourcing (Kittur et al. 2013), especially with paid microtask platforms such as Mechanical Turk (MTurk) in which inexpert, remote, unknown annotators are provided only rudimentary communication channels and training. The annotation process is largely opaque, with only the final labels being observable. Such factors do little to inspire trust between parties and faith in the overall paradigm. Risks may be seen to outweigh potential benefits, limiting the scale and complexity of tasks for which crowdsourcing is considered viable, and thereby the number of jobs made available to workers. When the accepted practice to ensure data quality requires posting a task redundantly to multiple workers, the cost of data collection increases and worker wages suffer.

We propose that *annotator rationales* (Zaidan, Eisner, and Piatko 2007) offer a new opportunity for traction on the above problems. The key idea of rationales is to ask human annotators to provide justifications for their labeling decisions in a particular, constrained form. As with Zaidan, Eisner, and Piatko (2007), we emphasize that the idea of rationales generalizes beyond the particular annotation task or form of rationale used (e.g., Donahue and Grauman (2011) investigate rationales for imagery tasks). However, while rationales were originally conceived merely to support a specific machine learning goal (and pursued with trusted annotators), we hypothesize that rationales offer far broader applicability and benefits to be realized (Section 2).

We ground our investigation of annotator rationales in the specific Information Retrieval (IR) task of *relevance assessment*, which calls on human judges to rate the relevance of *documents* (e.g., Webpages) to search queries. Unlike simple labeling tasks, describing relevance criteria precisely is difficult. Consequently, annotator agreement is typically low, even with trusted judges (Voorhees 2000; Bailey et al. 2008). While crowdsourcing's potential for more efficient relevance judging has sparked great interest (Alonso, Rose, and Stewart 2008), its use has tended to only further exacerbate issues of low annotator agreement.

In this work, we ask *assessors* to provide a rationale for each judgment by copy-and-pasting a short document excerpt (2-3 sentences) supporting their judgment. **Table 4** shows examples. To collect relevance judgments, we created three task designs, iteratively refined through pilot experiments (Section 4). Our Standard Task collects relevance judgments without rationales. While intended as a baseline, it slightly outperforms careful task design of prior work (Hosseini et al. 2012), without any use of Honey-Pot questions or platform-specific worker filtering mechanisms. Our Rationale Task achieves further improvement, and remarkably, does so entirely from asking judges to provide rationales; the submitted rationales themselves are completely ignored. Moreover, we find that *experienced* workers (completing 20 or more tasks) are able to complete the Rationale Task with almost no increase in average task completion time (29 vs. 27 seconds). Finally, our Two-Stage Task asks one judge to complete the Rationale Task, then a second *reviewer* to verify or fix that judgment. With the same number of workers and task cost, the Two-Stage Task yields further improvement in quality over the Rationale Task.

Whereas Zaidan, Eisner, and Piatko (2007) motivate annotator rationales solely to support *dual-supervision* over collected rationales and labels, so far we have only discussed data quality improvement we achieve while ignoring the collected rationales. To derive further benefit from the rationales themselves, we hypothesize that our task design will

encourage multiple annotators judging the same document to select similar excerpts as rationales (i.e., that more accurate judges will tend to select rationales exhibiting greater textual *overlap* with one another). We exploit such correlation when aggregating judgments by excluding judgments whose rationales exhibit low overlap. Section 5 presents two heuristic algorithms for such filtering which yield further improvement over the Rationale Task without filtering.

In sum, we believe adoption of rationales stands to significantly promote greater transparency and trust in crowdsourcing. Our cost-benefit analysis conducted over 10,000 MTurk judgments[1] shows the practical effectiveness of our approach in improving data quality. In addition, we believe rationales offer a myriad of further benefits (Section 2), including better quality assurance for subjective tasks, reduced need for expert gold (e.g., our eschewing of Honey-Pots discussed above), the opportunity for *dual-supervision* for both domain task (Zaidan, Eisner, and Piatko 2007) and label aggregation, and added value of the rationales themselves.

2 Motivations for Annotator Rationales

Enhancing Transparency. As discussed earlier, annotator rationales offer a simple, concise, and light-weight form of communication to explain a given answer and demonstrate it represents a thoughtful decision. When a worker disagrees with "expert" opinion or accepted gold for objective tasks, a rationale can help establish the validity of an alternative answer or reveal errors in the gold standard. For subjective tasks in which answer quality can be difficult to directly evaluate or verify, rationales provide a focused context to interpret a given answer and assess whether it is plausible. Once acquired, rationales stand to benefit all future users of a dataset, not only those who originally collected it. When data from a given study is published, rationales would help others to inspect and assess data quality for themselves.

`CrowdFlower.com` already employs rationales in the other direction. Because requesters' gold *honey-pot* questions used to evaluate workers are not always clear, correct, or complete, requesters are encouraged to provide a textual "reason" for each correct answer in order to justify it to workers. Workers, in turn, are provided an avenue to appeal if they disagree. In this spirit, collecting rationales also provides a new means of scalable crowd-based creation of honey-pot questions for worker testing: even if the "gold" label is sometimes wrong, the rationale provides a basis on which workers who disagree can appeal (and thereby simultaneously checking correctness of both workers).

Enhancing Quality. Collecting rationales may also help to encourage more thoughtful decision making and discourage any temptation to cheat. When one need only provide a label, it is rather easy to click and be done without giving the task much thought. However, when one is forced to provide a rationale for one's decisions, greater care and reflection is needed. In addition, when one is paid per-task (rather than hourly) and any answer seems acceptable (e.g., subjective rating tasks), it can be tempting to answer quickly to increase one's effective pay rate. An established practice to discourage such behavior is to design tasks such that it is no easier to create "believable invalid responses" than to undertake the given task in good faith (Kittur, Chi, and Suh 2008). *We hypothesize that creating a plausible rationale for a randomly selected answer would be at least as effortful as simply undertaking the task in good faith*. We test this hypothesis indirectly by evaluating the quality of judgments obtained in the presence or absence of rationales.

Moreover, because rationales can be checked relatively easily (even for subjective tasks), *we hypothesize this will reduce the temptation to cheat* (due to greater perceived risk of getting caught). As above, we test this hypothesis indirectly. Significantly, we would expect that higher quality data might be obtained by simply requiring rationales, *even if they are discarded without inspection*. Because subjective tasks make it difficult to verify answers, Kittur, Chi, and Suh (2008) recommend creating additional, non-task verifiable questions to include (e.g., "What is the third word on this page?"). However, such questions are easily distinguished from real task questions, so they can be easily passed without undertaking the real task questions in good faith (Marshall and Shipman 2013). In contrast, because rationales are tied to the real task questions of interest, they support more robust measurement and verification of data quality.

Enabling Crowd Verification. Rationales also create a new opportunity for utilizing iterative task design in the spirit of Find-Fix-Verify (Bernstein et al. 2010). While labels alone do not provide sufficient information for such iterative refinement, rationales could enable one worker's label and/or rationale to be further revised or refined by a subsequent worker (Section 4.4). Rationales could thus help extend the generality of sequential task design to a broader range of common data labeling tasks. Moreover, because rationales make it easier to verify worker answers, there is increased opportunity for effectively delegating such verification tasks to the crowd to reduce "expert" workload.

Improving Aggregation. As in Zaidan, Eisner, and Piatko (2007), collecting rationales generally enables *dual-supervision* of a learner over rationales and labels. In the context of crowdsourcing, while there has been plentiful work on label aggregation (Sheshadri and Lease 2013), we are not familiar with any prior work proposing dual-supervision for aggregation. In this paper, we present two heuristic algorithms to filter judgments based on rationale overlap, then aggregate remaining labels (Section 5).

Additional Domain Value. Finally, rationales themselves may provide additional value in the task domain. For example, whereas traditional document retrieval simply returns entire documents, *passage retrieval* and *question answering* seek to provide searchers with more focused search results than entire documents (Trotman, Pharo, and Jenkinson 2007). By requiring assessors to provide a document excerpt supporting a judgment of document relevance, judges effectively annotate relevant passages. While our task design encourages judges to converge on similar rationales (rather than find all relevant passages in a document), future work to support relevance judging for passage retrieval could relax this aspect of our task design and still realize many of the other benefits provided by collecting rationales.

[1] `http://github.com/tylermcdonnell/WhyIsThatRelevant`

3 Related Work

3.1 Effective Task Design

Alonso (2009) recommends collecting optional, free-form, task-level feedback from workers. While the authors found that some workers did provide example-specific feedback, the free-form nature of their feedback request elicited a variety of response types, difficult to check. They also found that requiring open-ended feedback led many workers to submit unhelpful text that was difficult to automatically cull.

In contrast, we assume rationales are required, constrained, and example-specific. Because rationales are strictly-defined, it is possible to provide clear instructions about what is expected (e.g., in our work, a document extract of specified length). We can automatically check (strictly or approximately) if a submitted rationale is found in the document, and whether length requirements are met. While task-level feedback is certainly useful, rationales serve to explain specific labeling decisions. Moreover, because rationales are document extracts, they enable dual-supervision, as in Zaidan, Eisner, and Piatko (2007)'s work, and can provide additional domain-specific value (e.g., in our task, implicitly marking relevant document passages).

3.2 Relevance Judging & Agreement

While the concept of search relevance has been investigated for over 80 years, it remains a thorny phenomenon with many complex and interacting underlying factors (Saracevic 2007). To create a useful gold-standard to train and evaluate IR systems, relevance judges are typically instructed to assess a simplified form of *topical relevance* which ignores various factors, such as redundancy in search results, the searcher's prior knowledge about the topic, etc. (Voorhees 2001). For 25 years, NIST TREC (`trec.nist.gov`) has organized shared task evaluations and collected and shared relevance judgment datasets to support IR evaluation (Voorhees, Harman, and others 2005). Trusted relevance judgments are a cornerstone of TREC, and we adopt TREC judgments as our gold standard (Section 6).

In perhaps the closest work to our own, Hosseini et al. (2012) compare a sophisticated *full* task design to a baseline *simple* task design for collecting relevance judgments. For each task, 3 MTurk judgments were collected per document, followed by aggregating judgments using EM (Dawid and Skene 1979). They achieve 80-82% accuracy, including when aggregating all 6 judgments per document across task designs. In other work, Blanco et al. (2011) use MTurk relevance judging and achieve Fleiss' κ of 0.3 ± 0.18 for ternary judgments and 0.45 ± 0.23 for binary judgments.

Historically, relevance judgments are not typically collected from multiple trusted judges for the same document, precluding measurement of inter-annotator agreement. When such data has been collected, relatively low agreement is typical, even with trusted judges (Voorhees 2000; Bailey et al. 2008). In short, describing precise relevance criteria is difficult, even when assuming simplified *topical relevance*. Precise quantitative comparisons among prior work are difficult to make due to a wide variety of judging scales and agreement measures which have been reported, as well as lack of crowd judging. That said, Bailey et al. (2008)'s Table 1 suggests Jaccard accuracy of 31-55%.

Little work has investigated multi-assessor agreement in making passage-level relevance assessments (Trotman, Pharo, and Jenkinson 2007), though agreement appears low here as well and merits further investigation.

4 Task Design

This section describes three different task designs developed and evaluated in this work: our Standard Task (Section 4.2, which does not collect rationales, our Rationale Task (Section 4.3), and a Two-Stage Rationale Task (Section 4.4). We begin by describing our initial pilot studies (Section 4.1) which iteratively experimented with different designs.

4.1 Pilot Studies

Our iterative design process involved deploying many small-scale relevance judgment tasks which varied key design features, such as the specificity of instructions for the crowd worker, the format of the grading scale, and most importantly, the definition of a rationale. In each iteration, we relied on manual inspection of work to evolve our design. We also included a free-form text box in which workers were encouraged to provide constructive feedback for the task with a possibility of bonus compensation (Alonso 2009).

Before launching our many studies, we conducted a medium-scale pilot study judging 70 Webpages from ClueWeb09 (see Section 6.1) for 25 search topics drawn from the 2009 TREC Million Query Track (Carterette et al. 2010). For each document, we collected 8 judgments each for Standard and Rationale Tasks. One of the authors blindly judged each document, both as a benchmark for what level of agreement we might expect from the crowd, and to account for potential changes in content, since crowd workers were directed to judge live web pages rather than originally crawled versions of the Webpages judged by TREC (see Section 6.1). Essentially all of the same trends and findings reported in our main study evaluation (Section 6) were observed in this earlier pilot study. In this sense, our second, main study implicitly shows that our findings are reproducible, similar to Blanco et al. (2011)'s work.

However, besides the scale of this pilot study being relatively small and some parameters for filtering (Section 5) being tuned on pilot study data, the real problem we encountered was an inexplicable problem with the MQ Track gold judgments. While the crowd was internally consistent and consistent with the author's judgments, neither were consistent with the TREC judgments for reasons we could never explain. Consequently, for our main study, we abandoned the MQ Track data in favor of the Web Track's data (see Section 6.1). Of relevance to our broader motivation for collecting rationales, despite close analysis of the MQ gold standard by one of the authors, we were unable to explain the gold judgments we found there. Having only the judgments and topic narrative provided little insight, whereas if the gold data had included the sort of rationales we motivate here, we imagine this problem might have been resolvable.

4.2 Standard Task

No Qualifications or Honey-Pots. An important design decision made from the outset was to avoid reliance on any platform-specific worker filtering. Crowdsourcing research too closely-tied to a particular commercial platform's capabilities (or addressing its peculiar limitations) risks reducing the generality and impact of its findings (Adar 2011). Some platform features offer unknown black-box behavior (e.g., MTurk's *Master Workers*), while others (e.g., MTurk's *Approval Rating*) have known and significant flaws (Ipeirotis 2010). Neither seems like a solid foundation to build upon.

In addition, while some Requesters impose geographic restrictions to exclude certain regions, presuming lower quality work, such geographic filtering can represent a biased crutch compensating for lazy design. We embrace crowdsourcing's ideal of providing anyone interested an equal opportunity to work and demonstrate their ability. Our responsibility in task design is to enable this vision.

Gold honey-pots require "expert" effort to create and typically assume objective tasks in which each question has a single, correct answer. Such tests are inherently less applicable for subjective rating tasks. Ideal quality assurance methods would be effective and applicable to both task types.

Relevance Scale. At what granularity of relevance should judgments be collected? Binary judgments are simplest but least informative for system evaluation, and all-or-nothing relevance permits judges no way to indicate borderline relevance, which searchers often encounter in practice. On the other hand, a finer granularity scale such as {*Perfect, Excellent, Good, Fair, Bad*} (Sanderson 2010) is more informative for system evaluation and more flexible for judges, but requires making more subtle distinctions and borderline decisions between categories. Prior literature appears to offer little guidance regarding how choice of relevance scale interacts with judges' efficiency and effectiveness in executing their work (Tang, Shaw Jr, and Vevea 1999).

Simply adopting the same relevance scale found in our TREC gold standard would have been simplest. However, as mentioned above, we found scant past work justifying *why* any particular scale was better than any other; we know of no evidence that the TREC scale is optimal or any set of factors for which it was optimized. Moreover, TREC has traditionally assumed trained judges rather than the crowd, and prior work has suggested a mismatch (Alonso 2009). Even if the gold standard had k categories, judges might still find it easier to judge on a $k + 1$ scale, and judgments could be post-processed to conflate categories for comparison. Finally, we did not want to tie our task design to an arbitrary gold standard we happened to be evaluating against here.

After iterative experimentation with MTurk judges using a variety of options for scale granularity and relevance category names, we ultimately selected a balanced, quaternary (4-point) scale with the following named categories: {*Definitely Not Relevant, Probably Not Relevant, Probably Relevant, Definitely Relevant*}. Having four degrees of relevance, uniformly spaced across the spectrum of relevance, appears to offer flexibility to judges without overwhelming them, satisfying the so-called "Goldilocks" criterion (offering neither too many nor too few options). Consistent with best practices (Alonso 2009; Blanco et al. 2011), it was found best to name relevance categories colloquially and avoid technical jargon (e.g., *marginally relevant*) familiar to trained judges but not intuitive or meaningful to laymen. Final category names used are, by design, both consistent with one another and symmetric with regard to adjectival descriptors and colloquiality. This helped us to simplify task instructions (see below) and appeared to improve judges' comprehension and distinction between relevance categories. Finally, by excluding a neutral option, there is no "easy-out" for judges when they are unsure, forcing them to actively lean in one direction or the other.

Instructions. We ultimately converged on very concise and self-explanatory task instructions which were less specific. Our early designs experimented with different relevance category names and accompanying specific instructions which showed examples of pages that should fall into each category and why. However, we received feedback suggesting workers were frustrated by this level of instruction, not only because of the extent of reading required, but also because it made them feel unsure of whether a given document would fit under the strict, but ultimately ambiguous, definitions we provided. (As a point of comparison, Google (2016)'s own extremely detailed judging guidelines still fall back on the phrase "use your judgment" 22 times). While we initially tried to address these concerns by further clarifying corner cases, the instructions quickly became unwieldy, while a long-tail of new corner cases continued to be found with each new batch of experiments. By adopting the colloquial and self-explanatory relevance scale above, we were able to provide very concise task instructions.

Payment. We expected the Rationale Task to take more time than the Standard Task; Zaidan, Eisner, and Piatko (2007) found it typically required two trusted annotators twice as long to collect rationales in addition to labels. Consequently, our pilot study paid \$0.05 for the Standard Task vs. \$0.10 for the Rationale Task. However, we were surprised to observe that *experienced* workers (who completed 20 or more tasks) converged to the same average completion time for both tasks, reproduced in our main study in (**Figure 2**). Consequently, our main study paid the same for both tasks, also rendering payment a fixed control variable in explaining any difference observed in work time or quality. Our pilot study also gave us a good estimate for task completion times, leading us to set task payment at \$0.05 (roughly \$6.00 hourly wage for experienced workers) for all task types: Standard, Rationale, and Two-Stage review.

4.3 Rationale Task

A major goal of our early pilot studies was to experiment with different definitions of rationale in order to determine what worked best. Regarding rationale length, for Zaidan, Eisner, and Piatko (2007), annotating a few keywords sufficed for the learner; for us, keywords do not provide meaningful insight into the thought process of the annotator. Moreover, keyword rationales might bias judges toward simple *keyword-spotting* (i.e., judging any document containing a query term as relevant). In contrast, we wanted assessors to reflect and provide more complete justifications.

On the other hand, extremely long excerpts would not provide focused insight into a judge's thought process and key elements of their decision-making process. Moreover, if wish to support multi-stage, sequential tasks in which one judge verifies or fixes another's work (Section 4.4), an overly long rationale might save only minimal time for second judge vs. simply re-reading the entire document.

Giving no guidance on expected length provided little task clarity for judges and tended to result in overly terse responses, insufficient for either understanding judges' thought processes or automatically analyzing their rationales for overlap-based filtering (Section 5). Consequently, we found that requesting rationales of roughly 2-3 sentences in length frequently provided clear, focused insight into the worker thought process and supported post-processing.

Requiring document excerpts rather than free-form feedback enables one to automatically check (strictly or approximately) if a submitted excerpt is actually found in the document. Moreover, excerpts permit dual-supervision (Zaidan, Eisner, and Piatko 2007) and can provide additional domain-specific value (e.g., implicitly marking relevant passages).

However, this form of rationale is not appropriate for all situations. Because resource-type searches and Webpages dominated by imagery provide few useful textual extracts for explaining relevance, we created special instructions for these cases: workers were asked to manually type a fixed string if the document's text did not support their judgment, which was then treated as their rationale. Workers provided this string in lieu of a rationale in roughly 10% of cases.

4.4 Two-Stage Rationale Task

In the spirit of Find-Fix-Verify (Bernstein et al. 2010), we also designed a two-stage, sequential task which first collected a relevance judgment and rationale from a single judge, and then asked four subsequent *reviewers* to either confirm or modify the initial judgment. Second-stage reviewers were presented with the following scenario:

> Alice was looking for some information. She typed the following search query into a popular search engine. Tom looked at the page seen below and decided whether he thought the page was *Definitely Not Relevant*, *Probably Not Relevant*, *Probably Relevant*, or *Definitely Relevant* to Alice's search. He also provided the following quotation from the web page to support his decision. Please answer the following questions:
>
> 1. Do you agree with Tom's assessment of the page for Alice's search? If not, how would you rate the page? (*multiple-choice selection of relevance judgment*)
> 2. Please describe in words why you agree or disagree with Tom's decision. (*free-form feedback*)

To better understand how reviewers conducted the review process, our second question above requested open-ended justifications (rather than revised rationales). Because the second-stage task did not request a revised rationale, it was not possible to apply our overlap-based filtering methods TOP-N and THRESHOLD in conjunction with our two-stage design. As with other task designs, however, we could collect multiple judgments and then aggregate judgments to induce a combined consensus judgment.

Algorithm 1 Threshold Filtering

```
1:  procedure FILTER-BY-THRESHOLD(J_d)
2:      T ← SELECT-THRESHOLD(J_d)
3:      selected ← ∅
4:      for each (j_1, j_2) ∈ COMBINATIONS(J_d, 2) do
5:          if SIMILARITY(j_1, j_2) ≥ T then
6:              selected ← selected ∪ j_1 ∪ j_2
7:      return selected
8:  procedure SELECT-THRESHOLD(J_d)
9:      T ← 0
10:     for each (j_1, j_2) ∈ COMBINATIONS(J_d, 2) do
11:         T ← max(T, SIMILARITY(j_1, j_2))
12:     return ROUND-DOWN(T, 10)
```

Algorithm 2 Top-N Filtering

```
1:  procedure FILTER-BY-TOP-N(J_d, N)
2:      pairs = COMBINATIONS(J_d, 2)
3:      for each pair ∈ pairs do
4:          pair.sim ← SIMILARITY(pair.j_1, pair.j_2)
5:      Sort(pairs) by descending similarity
6:      return GETTOPJUDGMENTS(pairs, N)
```

5 Filtering Judgments by Rationale Overlap

Assuming our task design motivates workers to quickly find clear rationales for their judgments, maximizing per-task compensation, we hypothesize that judges will tend to converge on selecting similar document extracts as rationales: one of the first plausible rationales found in a document. While we do not investigate the document position of selected rationales in this study (left for future work), we do attempt to exploit any such correlation between *overlap* in annotator rationales and judging accuracy. Specifically, we seek to filter out (pre-aggregation) any judgments whose corresponding rationales show low overlap with others'. We describe two heuristic algorithms for this below.

THRESHOLD FILTERING (Algorithm 1) computes the similarity between each pair of rationales provided by annotators for a particular document, defines a similarity threshold T (determined automatically, as discussed below), and selects all judgments whose rationales have a similarity score with at least one other rationale that is greater than or equal to T. Though a single threshold value may be selected for all documents, in practice this might require a large number of worker rationales to ensure that at least one pair per document meets the threshold in order to provide judgments for each document in the test collection. Instead, our algorithm dynamically chooses a threshold for a particular document by rounding down to the nearest 10 from the highest observed similarity among the rationales collected for that document. The intuition is to capture the opinions of people with rationale similarities in the vicinity of the maximum observed, rather than unnecessarily excluding all but the highest N, which our Top-N Filtering algorithm below does. We

arbitrarily chose this rounded-down value and noticed that it worked well in our pilot study, though we did not optimize it for either the pilot or main study.

TOP-N FILTERING (Algorithm 2) computes the similarity between every pair of rationales provided by annotators for a particular document and selects the N judgments with the highest similarity to some other rationale in the batch.

In either case, the selected judgments are then aggregated (see Section 6 for aggregation methods). These heuristics rely on a metric for computing similarity between strings (rationales), but they are general enough such that any similarity metric which is monotonic may be applied. For our evaluation, we use the Ratcliff-Obershelp similarity metric (Ratcliff and Metzener 1988), which computes the similarity of two strings as the number of matching characters divided by the total number of characters, where matching characters are taken from the longest common subsequence (LCS) and then recursively from the regions on either side of the LCS. Other string similarity measures, such as Jaccard, yielded similar results (not shown or discussed further).

6 Evaluation

We evaluate the potential benefits of annotator rationales for the specific IR task of collecting document relevance judgments, a task which prior work has shown to have low annotator agreement (Section 3.2). While relevance in general is quite subjective (Saracevic 2007), *topical relevance* is intended to be impersonal and objective (Voorhees 2001). We thus hypothesize that: 1) *high agreement is possible, provided one is willing to invest enough annotation effort to achieve it*; and that 2) *rationales require relatively little additional effort to achieve higher annotator agreement*.

We first investigate whether collecting rationales during crowdsourced relevance judging can improve the quality of judgments, even if the submitted rationales themselves are completely ignored. To evaluate this, we perform A/B testing of our Standard Task (Section 4.2) vs. our Rationale Task (Section 4.3). We measure annotator agreement (Section 6.2) to test whether the crowd is internally consistent, regardless of their agreement with our gold standard.

Next, we measure the accuracy of crowd judgments (individually and aggregated) vs. the TREC gold standard (Section 6.3). While aggregation using either majority vote (MV) or EM ala Dawid and Skene (1979) yielded largely similar results, EM results were always at least as good (and sometimes better), so we largely omit MV results due to space. Following this, Section 6.4 evaluates whether accurate judges select similar document extracts as rationales. Specifically. we evaluate the two methods described in Section 5 for filtering out judgments with low rationale overlap prior to performing aggregation. Section 6.5 then reports our cost-benefit analysis of Standard vs. Rationale Tasks. Finally, Section 6.6 evaluates the Two-Stage Rationale Task in which one assessor's judgment and rationale are reviewed by a second judge for verification or correction.

6.1 Experimental Setup

We collect *ad hoc* Web search relevance judgments for the ClueWeb09 dataset (`lemurproject.org/clueweb09`). Search *topics* and judgments are drawn from the 2009 TREC Web Track (Clarke, Craswell, and Soboroff 2010). Each topic includes a narrative for the user's *information need* which we provide to judges (See Table 4 for an example). We utilize TREC gold relevance judgments using a 3-point scale: *not relevant, relevant* and *highly relevant*.

We select 700 documents to judge from different topics covering 43 of the 50 topics in the Web Track. TREC gold judgments for our 700 documents are distributed as follows: 46% *not relevant*, 24% *relevant*, and 30% *highly relevant*. We evaluate collected crowd judgments against both this ternary gold standard and a binarized version in which we collapse *relevant* and *highly relevant* distinctions (yielding 46% *not relevant* and 54% *relevant* documents).

While we had planned to judge ClueWeb09's crawled Webpages, images and style sheets associated with each page were often missing or rendered incorrectly, making the rendered pages difficult to assess. Consequently, we decided to judge the live web pages associated with each crawled URL. The 700 URLs we judge exclude all URLs yielding a `Page Not Found` error. Also, because live web pages today may differ from the versions crawled and judged in 2009, one of the authors blindly judged 200 of these URLs. Results closely mirrored the gold standard (95% binary accuracy, 88% ternary accuracy), suggesting that the gold judgments are reasonably accurate for the live web pages.

We collect 5 crowd responses per Webpage (700x5=3500 judgments) for each task design: Standard, Rationale, and Two-Stage. We set $N = 3$ for TOP-N judgment filtering and rounding down to the nearest 10 for THRESHOLD filtering based on pilot experiments (Section 4.1).

6.2 Annotator Agreement

We measure agreement using Fleiss' Kappa $\kappa_F = \frac{\bar{P}-\bar{P}_e}{1-\bar{P}_e}$, where $1 - \bar{P}$ is the agreement attainable by chance and $\bar{P} - \bar{P}_e$ is the degree of agreement achieved above chance. Blanco et al. (2011) question use of Fleiss' Kappa with crowd annotations since it assumes a consistent set of annotators, while the set of crowd annotators per example is rarely consistent since workers come and go. However, Blanco et al. (2011) still report mean κ_F, and after exploring a variety of measures for measuring inter-annotator agreement, we found that each told a similar story and none more clearly or simply than Fleiss' Kappa, which we adopt.

Table 1 shows agreement between crowd judges. We observed much higher inter-annotator agreement among judgments collected with Rationale (binary $\kappa_F = 0.79$) and Two-stage (binary $\kappa_F = 0.85$) vs. Standard (binary $\kappa_F = 0.61$). The ternary agreement shows similar trends. Intended to serve as a strong baseline vs. prior work, Standard's ternary agreement $\kappa = 0.36$ slightly exceeds Blanco et al. (2011)'s κ_F of 0.3 ± 0.18, while Standard's binary $\kappa_F = 0.61$ far exceeds their own κ_F of 0.45 ± 0.23.

Near-perfect binary agreement of second stage annotators in Two-Stage is particularly notable, suggesting its design emphasizes critical thinking elements central to making reasonable and consistent judgments. While common practice of aggregating results from 3-5 workers may be necessary

with the Standard task design to remedy its relatively low agreement seen here (a moderate binary κ_F of 0.61 and a fair ternary κ_F of 0.36), higher agreement seen between workers on Rationale and Two-Stage tasks suggest that sufficient data quality might be achievable using only 1-2 workers, and without requiring any further aggregation stage.

Judging Task	Binary Agreement	Ternary Agreement
Standard	0.61 (*moderate*)	0.36 (*fair*)
Rationale	0.79 (*substantial*)	0.59 (*moderate*)
Two-Stage	0.85 (*near-perfect*)	0.71 (*substantial*)

Table 1: Annotator Agreement using Fleiss' Kappa κ_F, whose values are typically interpreted assuming 5 equisized bins: *slight, fair, moderate, substantial*, and *near-perfect*.

6.3 Individual and Consensus Accuracy

In addition to measuring simple accuracy to evaluate the quality of crowd judgments vs. TREC gold, we also adopt Cohen's Kappa κ_C (Carletta 1996; Artstein and Poesio 2008; Bailey et al. 2008), which accounts for chance in measuring agreement between two raters. We treat TREC gold as one rater and either a single crowd judge or aggregated crowd consensus as the other. Cohen's Weighted κ incorporates weights for treating disagreements differently, e.g., assigning partial credit for almost-correct answers in our ordinal judging scale. We adopt a squared weighting function without tuning. While Weighted Kappa seems most appropriate to us with ordinal judging, we note that regular Kappa (not shown) yielded similar results. κ_C agreement can be interpreted similarly to Fleiss' Kappa κ_F above: *slight* $[0-0.2]$, *fair* $[0.2-0.4]$, *moderate* $[0.4-0.6]$, *substantial* $[0.6-0.8]$, and *near-perfect* $[0.8-1]$.

Table 2 shows binary and ternary quality of crowd judgments, as measured by both simple accuracy and Cohen's κ_C, reported for individual judgments and consensus induced from aggregating 5 judgments. Our Standard Task is intended to serve as a strong baseline vs. prior work, and its binary accuracy of 86% actually outperforms the 80-82% binary accuracy achieved by Hosseini et al. (2012)'s careful task design. Moreover, unlike them, we do not rely on any Honey-Pot questions or platform-specific worker filtering.

In comparing the Rationale Task vs. the Standard Task, we observed notable improvement for both conditions of individual judging (Row 2 vs. 1) and aggregate consensus (Row 5 vs. 4), as well as binary vs. ternary evaluation. With individual judging (Row 2 vs. 1), Rationale outperforms Standard for both binary judging (80% accuracy & $\kappa_C = 0.51$ vs. 65% accuracy & $\kappa_C = 0.36$) and ternary judging (64% accuracy & $\kappa_C = 0.5$ vs. 47% accuracy & $\kappa_C = 0.34$). For consensus, we see aggregated judgment quality for Rationale beats Standard (Row 5 vs. 4) for both binary judging (92% accuracy & $\kappa_C = 0.8$ vs. 86% accuracy & $\kappa_C = 0.59$) and ternary judging (84% accuracy & $\kappa_C = 0.8$ vs. 75% accuracy & $\kappa_C = 0.46$), though we do see shrinking gains as we aggregate judgments and collapse to binary relevance.

6.4 Filtering Judgments via Rationale Overlap

We next investigate whether accurate judges select similar document extracts as rationales, evaluating the two methods from Section 5 for filtering out judgments having rationale *overlap* prior to performing aggregation.

Using TOP-N (Algorithm 2) or THRESHOLD (Algorithm 1) methods (with parameters set as discussed earlier), we observe accuracy gains across the board. Consensus results using THRESHOLD filtering (Row 7) vs. no filtering (Row 5) show binary judging of 96% accuracy & $\kappa_C = 0.85$ vs. 92% accuracy & $\kappa_C = 0.80$ and ternary judging of 91% accuracy & $\kappa_C = 0.84$ vs. 84% accuracy & $\kappa_C = 0.80$. This indicates that accurate assessors do select similar document extracts as rationales, indicating a correlation between *overlap* in annotator rationales and judging accuracy.

Comparing methods, THRESHOLD (Row 7) consistently outperforms TOP-N (Row 6). Whereas TOP-N always uses a fixed number of judgments, THRESHOLD tunes the number of judgments kept per document according to the level of observed overlap. This suggests that adapting to rationale nuances between different Webpage types is important.

6.5 Cost-Benefit Analysis of Rationales

While **Table 2** shows simple accuracy for the binary relevance of Standard vs. Rationale tasks using either 1 judgment (individual judging) or 5 judgments (aggregate consensus), **Figure 1** shows the full range of how accuracy varies across the full range of [1:5] judgments. We randomly sample n judgments (x-axis) and apply MV consensus (EM results were similar), averaging over 20 random trials for each judgment count. Binary accuracy of Standard judging exhibits fairly consistent gains as judgments increase, achieving 86% accuracy with 5 judgments. In contrast, Rationale Judging approaches 90% accuracy with only three judgments, then shows rather modest gains thereafter.

As discussed earlier (Section 4.2), workers were paid the same amount ($0.05, roughly $6.00 hourly) for both tasks based on task completion times observed in our pilot study, where *experienced* workers (who completed 20 or more tasks) were remarkably seen to converge to nearly the same average completion time for both tasks (27 seconds for Standard and 29 for Rationale; Two-Stage's reviewer task took 26 seconds, not shown). In contrast, Zaidan, Eisner, and Piatko (2007) found it typically required two trusted annotators twice as long to collect rationales in addition to labels. We offer a possible explanation for this difference below.

We originally plotted the average time *all* workers spent completing each of their first 20 tasks (up to the total number of tasks completed per worker). However, while the plot was essentially identical to **Figure 2**, it was unclear whether completion time decreased with experience or if experienced workers were always faster, and increasing the number of tasks simply filtered out slower workers. To remedy this, **Figure 2** instead plots the average time the subset of *experienced* workers spent completing each of their first 20 tasks, clearly showing a decrease in task time with more experience. We received an average of 415 unique workers for each task type, with 8% of workers completing 20

				Binary		Ternary	
Row	**Task**	**Filter**	**Judgments**	**Accuracy**	**Cohen's κ_C**	**Accuracy**	**Cohen's κ_C**
1	Standard	-	Single Judge	0.65	0.36	0.47	0.34
2	Rationale	-	Single Judge	0.80	0.51	0.64	0.50
3	Two-Stage	-	Judge + Reviewer	0.85	0.58	0.75	0.60
4	Standard	-	5 Judges (EM)	0.86	0.59	0.75	0.46
5	Rationale	-	5 Judges (EM)	0.92	0.80	0.84	0.80
6	Rationale	TOP-3	5 Judges (EM)	0.93	0.81	0.91	0.82
7	Rationale	THRESHOLD	5 Judges (EM)	0.96	0.85	0.91	0.84
8	Two-Stage	-	Judge + 4 Reviewers (EM)	0.96	0.85	0.91	0.85

Table 2: Quality of judgments obtained vs. TREC gold using different task designs (Standard, Rationale, and Two-Stage) and individual vs. aggregate judging, measuring simple accuracy vs. Cohen's Weighted Kappa κ_C for binary vs. ternary relevance.

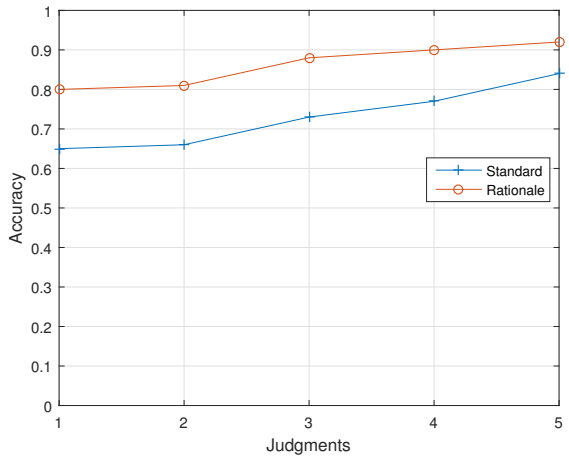

Figure 1: Judging accuracy vs. number of judgments, with MV for aggregation in the case of multiple judgments.

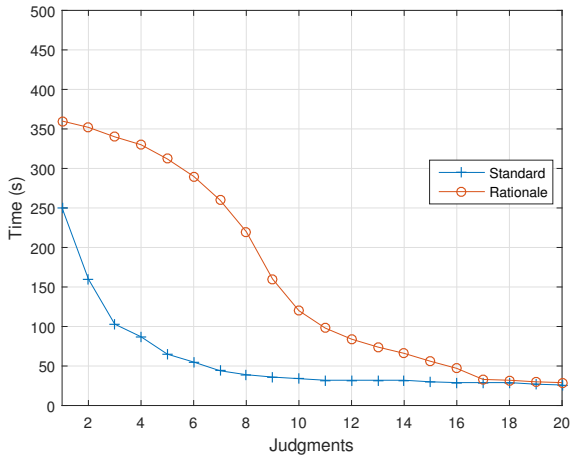

Figure 2: Average completion time vs. completed task count on Standard vs. Rationale tasks for experienced workers.

or more tasks. These experienced workers accounted for approximately 50% of all the judgments in our study.

Intuitively, both Standard and Rationale tasks involve overhead for reading instructions and task familiarization. For Standard, we see task time rapidly fall off after this early phase, whereas Rationale task time drops more slowly. We speculate this is due to relevance judgment tasks being more common and familiar on MTurk. However, task time critically converges in both cases for *experienced* workers. We hypothesize that once familiarized with the task, both tasks effectively require the same mental processes and effort: reviewing document text in order to formulate a relevance decision; the Rationale task simply makes this explicit.

6.6 Two-Stage Task Results

Our Two-Stage Task (Section 4.4) collects a judgment and rationale from a single assessor, then asks 4 subsequent *reviewers* to either confirm or modify the initial judgment.

Table 3 shows that the second-stage reviewer never introduced new judgment errors and fixed an error made by the initial judge 82% of the time. Further, recall the near-perfect binary agreement of second stage annotators in Two-Stage (Section 6.2). These results suggest that Two-Stage may provide high-quality data with only one judge and reviewer.

		Stage 2 Reviewer	
		Incorrect	Correct
Stage 1 Judge	Incorrect	4%	18%
	Correct	0%	78%

Table 3: Binary correctness contingency table of relevance judgments from Two-Stage Task. $\frac{18}{4+18} = 82\%$ of first stage errors are corrected without introduction of any new errors.

Table 2 shows that Two-Stage with 2 judgments (Row 3) achieves much higher accuracy (85% binary, 75% ternary) vs. either Rationale (Row 2) or Standard (Row 1). While comparing a 2-judgment design to a 1-judgment design may

Query (Alice)	dogs for adoption
Narrative (Alice)	I want to find information on adopting a dog. This includes names and locations of rescue organizations or vehicles (e.g. classifieds) as well as documents with info on qualifications, fees (if any), what to expect, resources, etc. Organizations may be rescue organizations, pounds, shelters, etc. but not breeders or pet shops, unless the pet shop runs adoption fairs. A site providing general information on dog adoption is also relevant.

	Document 1	**Document 2**	**Document 3**
Worker 1 Judgment (Tom)	Probably Not Relevant	Definitely Relevant	Definitely Not Relevant
Worker 1 Rationale (Tom)	*Rooterville Sanctuary. For adoption: pets, pig, pigs, piggy, piggies, pork.*	*View our rescue dogs - visit our organization or contact us directly to see what is available.*	*The dogs listed here all require a new home. These dogs all deserve that second chance and you may be that special person to give it to them. View Rescue Dogs adoption fees. Contact us for more info.*
Worker 2 Judgment	Probably Not Relevant	Probably Relevant	Definitely Relevant
Worker 2 Reasoning	I agree that this organization is probably not likely to be one where Alice will find the animal she is looking for, since they seem to focus on pigs, although they mention dogs	It is a site that lists dog rescue organizations, which is what Alice is searching for. But it is an Australian website. I suspect Alice was looking for an organization in the US.	Tom provided a lot of information that shows why this page should be useful for Alice.
Gold Standard	Probably Not Relevant	Probably Relevant	Definitely Relevant

Table 4: Examples of the Two-Stage Task with worker responses for three different documents.

not seem fair, **Figure 1** shows binary accuracy of Rationale and Standard Tasks for $x = 2$ is nearly unchanged, suggesting the above comparison is fair after all. Most significantly, **Table 2** shows that Two-Stage with 2 judgments matches Standard's performance with 5 judges (Row 4) using 3 fewer judgments and with higher ternary κ_C: 0.60 vs. 0.46.

Next, we consider the case of consensus with 5 judgments. We aggregate judgments from 4 second-stage reviewers so that the 1 judge + 4 reviewers = 5 judgments matches the 5 judgment count (and cost) of Rationale with consensus shown in **Table 2**. Two-Stage (Row 8) is seen to match the accuracy and κ_C of Rationale with THRESHOLD filtering (Row 7) while incurring the exact same cost.

Finally, note that while THRESHOLD filtering consensus achieves notable improvement over Rationale when using 5 judgments, with only 2 judgments it simply selects both, and is, therefore, equivalent to the unfiltered Rationale Task. As noted above, **Figure 1** shows that binary accuracy of Rationale for $x = 1$ and $x = 2$ are nearly identical, making Two-Stage the clear winner when using few judgments.

Qualitative Analysis **Table 4** presents a subset of judgments on three documents judged for the same search topic about pet adoption. The Table shows the judgment and rationale provided by the initial annotator, as well as the subsequent reviewer's judgment and reasoning. The gold standard is taken from one of the authors' blind judgments on the same 4-point scale used by the workers. Recall that the judge is always referred to as *Tom* in the second-stage task completed by the reviewer (Section 4.4).

Document 1. The judge rated the document to be *Probably Not Relevant*, citing a rationale which suggested that though the website was for a pet adoption sanctuary, they appeared to specialize in pigs. The reviewer affirmed this judgment and specifically cited agreement with Tom's rationale that the sanctuary focused on pigs, not dogs.

Document 2. The judge indicated *Definitely Relevant* because the website explicitly advertises dog adoptions. However, the reviewer tweaked the judgment to *Probably Relevant*, understanding Tom's justification but noting that the rescue organization is based in Australia and that, "I suspect Alice was looking... in the US." Such transparency of thought is invaluable since there is nothing explicit in the Narrative supporting the reviewer's supposition, though American vocabulary or spelling may be the culprit.

Document 3. The judge selected *Definitely Not Relevant*, but gave a rationale suggesting the website was quite relevant. The reviewer caught this, mentioning Tom's rationale, and suggested the submitted judgment was an accident.

Each example highlights the utility of rationales as a source of transparency and verifiability not possible with traditional relevance judging. In each case, the judge's rationale enabled the reviewer to weigh the judge's reasoning against their own. In all cases, the reviewer was empowered to take a different, confidently informed action: affirming, tweaking, or correcting the original judgment, respectively.

Reviewer justifications further suggest that the Two-Stage Task design requires a more involved critical thinking process in which reviewers understand that their duty is not only to form a strong justification for their subjective judgment, but also grounding their decision-making process in tandem with reasoning about the original judge's opinion.

7 Conclusion

We believe that forming a rationale is critical to forming a coherent judgment, whether or not task instructions explicitly require it. Our results show that requiring annotators to provide rationales incurs almost no additional time for *experienced* annotators (who complete 20 or more tasks), suggesting that annotators might be already doing so implicitly. By choosing to capture this critical reasoning process, a variety of benefits can be realized to improve transparency of work and quality of data from crowdsourcing, especially for subjective tasks in which multiple answers may be valid.

In contrast with most prior work, we invite anyone interested to work on our tasks (we perform no worker filtering), and we require no labeled data to test workers (i.e., on questions with known answers). Despite this, our baseline Standard Task for collecting relevance judgments without rationales still slightly outperforms prior work's careful task design (Blanco et al. 2011; Hosseini et al. 2012). Our Rationale Task design further improves data quality while entirely ignoring the collected rationales. With only two workers, our sequential, Two-Stage Task design achieves 85% binary accuracy. Aggregating judgments from 5 workers provides further improvement, and by exploiting degree of overlap in judges' rationales, we can achieve 96% binary accuracy.

In future work, we plan to further investigate sequential task iteration beyond two-stages, dynamic collection of judgments based on rationale overlap, dual-supervision of aggregation with rationales, and the validity of using crowdsourcing labels for conducting repeatable, reliable, and rigorous A/B system testing evaluations (Blanco et al. 2011).

Acknowledgments. We thank our many talented crowd contributors. This work was made possible by NPRP grant NPRP 7-1313-1-245 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

Adar, E. 2011. Why i hate mechanical turk research (and workshops). In *CHI Workshop on Crowdsourcing and Human Comp.*

Alonso, O.; Rose, D. E.; and Stewart, B. 2008. Crowdsourcing for relevance evaluation. In *ACM SigIR Forum*, volume 42, 9–15.

Alonso, O. 2009. Guidelines for designing crowdsourcing-based relevance experiments. CiteSeerX DOI 10.1.1.149.6649.

Artstein, R., and Poesio, M. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics* 34(4):555–596.

Bailey, P.; Craswell, N.; Soboroff, I.; Thomas, P.; de Vries, A. P.; and Yilmaz, E. 2008. Relevance assessment: are judges exchangeable and does it matter. In *SIGIR*, 667–674. ACM.

Bernstein, M. S.; Little, G.; Miller, R. C.; Hartmann, B.; Ackerman, M. S.; Karger, D. R.; Crowell, D.; and Panovich, K. 2010. Soylent: a word processor with a crowd inside. In *UIST*, 313–322. ACM.

Blanco, R.; Halpin, H.; Herzig, D. M.; Mika, P.; Pound, J.; Thompson, H. S.; and Tran Duc, T. 2011. Repeatable and reliable search system evaluation using crowdsourcing. In *SIGIR*, 923–932. ACM.

Carletta, J. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational linguistics* 22(2):249–254.

Carterette, B.; Pavlu, V.; Fang, H.; and Kanoulas, E. 2010. Million query track 2009 overview. In *Proceedings of NIST TREC*.

Clarke, C. L.; Craswell, N.; and Soboroff, I. 2010. Overview of the TREC 2009 Web Track. In *Proceedings of NIST TREC*.

Dawid, A. P., and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied statistics* 28(1):20–28.

Donahue, J., and Grauman, K. 2011. Annotator rationales for visual recognition. In *ICCV*, 1395–1402. IEEE.

Google. 2016. Search quality rating guidelines. *Inside Search: How Search Works*. `www.google.com/insidesearch/`.

Hosseini, M.; Cox, I. J.; Milić-Frayling, N.; Kazai, G.; and Vinay, V. 2012. On aggregating labels from multiple crowd workers to infer relevance of documents. In *ECIR*. Springer. 182–194.

Ipeirotis, P. 2010. A plea to amazon: Fix mechanical turk! *Blog: Behind Enemy Lines*. October 21, 2010. `www.behind-the-enemy-lines.com`.

Kittur, A.; Nickerson, J. V.; Bernstein, M.; Gerber, E.; Shaw, A.; Zimmerman, J.; Lease, M.; and Horton, J. 2013. The Future of Crowd Work. In *CSCW*, 1301–1318. ACM.

Kittur, A.; Chi, E. H.; and Suh, B. 2008. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, 453–456. ACM.

Marshall, C. C., and Shipman, F. M. 2013. Experiences surveying the crowd: Reflections on methods, participation, and reliability. In *5th Annual Web Science Conference*, 234–243. ACM.

Ratcliff, J. W., and Metzener, D. E. 1988. Pattern-matching-the gestalt approach. *Dr Dobbs Journal* 13(7):46.

Sanderson, M. 2010. *Test collection based evaluation of information retrieval systems*. Now Publishers Inc.

Saracevic, T. 2007. Relevance: A review of the literature and a framework for thinking on the notion in information science. part iii: Behavior and effects of relevance. *Journal of the American Society for Information Science and Technology* 58(13):2126–2144.

Sheshadri, A., and Lease, M. 2013. SQUARE: A Benchmark for Research on Computing Crowd Consensus. In *Proceedings of the AAAI Conference on Human Computation (HCOMP)*, 156–164.

Tang, R.; Shaw Jr, W. M.; and Vevea, J. L. 1999. Towards the identification of the optimal number of relevance categories. *Journal of the Association for Information Science and Technology* 50(3):254.

Trotman, A.; Pharo, N.; and Jenkinson, D. 2007. Can we at least agree on something. In *Proceedings of the SIGIR 2007 Workshop on Focused Retrieval*, 49–56.

Voorhees, E. M.; Harman, D. K.; et al. 2005. *TREC: Experiment and evaluation in information retrieval*. The MIT Press.

Voorhees, E. M. 2000. Variations in relevance judgments and the measurement of retrieval effectiveness. *Information processing & management* 36(5):697–716.

Voorhees, E. M. 2001. The philosophy of information retrieval evaluation. In *Evaluation of cross-language information retrieval systems*, 355–370. Springer.

Zaidan, O. F.; Eisner, J.; and Piatko, C. D. 2007. Using annotator rationales to improve machine learning for text categorization. In *HLT-NAACL*, 260–267.