University of Massachusetts - Amherst
ScholarWorks@UMass Amherst

Computer Science Department Faculty Publication Series

Computer Science

2008

Here or There: Preference Judgments for Relevance

Ben Carterette
University of Massachusetts - Amherst

Paul N. Bennett

David Maxwell Chickering

Susan T. Dumais

Follow this and additional works at: http://scholarworks.umass.edu/cs_faculty_pubs

Part of the Computer Sciences Commons

Carterette, Ben; Bennett, Paul N.; Chickering, David Maxwell; and Dumais, Susan T., "Here or There: Preference Judgments for Relevance" (2008). *Computer Science Department Faculty Publication Series.* Paper 46.
http://scholarworks.umass.edu/cs_faculty_pubs/46

This Article is brought to you for free and open access by the Computer Science at ScholarWorks@UMass Amherst. It has been accepted for inclusion in Computer Science Department Faculty Publication Series by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

Here or There
Preference Judgments for Relevance

Ben Carterette[1], Paul N. Bennett[2], David Maxwell Chickering[3], and Susan T. Dumais[2]

[1] University of Massachusetts Amherst
[2] Microsoft Research
[3] Microsoft Live Labs

Abstract. Information retrieval systems have traditionally been evaluated over absolute judgments of relevance: each document is judged for relevance on its own, independent of other documents that may be on topic. We hypothesize that preference judgments of the form "document A is more relevant than document B" are easier for assessors to make than absolute judgments, and provide evidence for our hypothesis through a study with assessors. We then investigate methods to evaluate search engines using preference judgments. Furthermore, we show that by using inferences and clever selection of pairs to judge, we need not compare all pairs of documents in order to apply evaluation methods.

1 Introduction

Relevance judgments for information retrieval evaluation have traditionally been made on a binary scale: a document is either relevant to a query or it is not. This definition of relevance is largely motivated by the importance of *topicality* in tasks studied in IR research [1].

The notion of relevance can be generalized to a graded scale of absolute judgments. Järvelin and Kekäläinen [2] proposed doing so to identify very relevant documents in addition to relevant and non-relevant documents. They developed the *discounted cumulative gain* (DCG) measure to summarize performance taking into account both graded relevance and greater importance for items retrieved at the top ranks. DCG has been used to evaluate web search applications where the first few results are especially important. In web search applications, factors other than topical relevance, such as quality of information, quality of display, or important of the site, are often included in assessing relevance.

Although evaluations over graded relevance allow for finer distinctions among documents, adopting graded relevance has two significant drawbacks. First, the specifics of the gradations (i.e. how many levels to use and what those levels mean) must be defined, and it is not clear how these choices will affect relative performance measurements. Second, the burden on assessors increases with the complexity of the relevance gradations; when there are more factors or finer distinctions to consider, the choice of label is less clear. High measured levels of disagreement on binary judgments [3] suggests the difficulty of the problem.

When measurement is difficult in practice or not completely objective, judgments of *preference* may be a good alternative [4]. Instead of assigning a relevance label to a document, an assessor looks at two pages and expresses a preference for one over the other. This is a binary decision, so there is no need to determine a set of labels and no need to map judgments to a numeric scale.

Of course, using preference judgments poses a new set of questions: how do we use preference judgments to evaluate a search engine? The number of pairs of documents is polynomial in the number of documents; will it be feasible to ask for judgments on every pair? If not, which pairs do we choose? But these questions are more amenable to empirical investigation.

There is another advantage to direct preference judgments: algorithms such as RankNet [5] and ranking SVMs [6] are trained over preferences. Sometimes preferences are obtained by inference from absolute judgments [5]. By collecting preferences directly, some of the noise associated with difficulty in distinguishing between different levels of relevance may be reduced. Additionally, absolute judgments result in ties in inferred preferences; direct preferences may allow more data to be used for training.

In this work we follow three successive lines of investigation. First, we compare assessor agreement and time spent per judgment for preference and absolute judgments. Next, we consider the evaluation of search engines when judgments are preferences. Finally, we look at focusing assessor effort to collect sufficient preferences to be able to compare search engines accurately.

2 Previous Work

The idea of pairwise preference judgments has not been explored much in the IR literature. When the idea of preference judgments has arisen, the practice has typically been to infer preferences from existing absolute judgments (e.g. [7, 8]), sidestepping questions about collecting preferences directly.

The most closely related previous work is that of Joachims, who first hypothesized that a click could be treated as a preference judgment (the document clicked being preferred to all ranked above it) [6], then used an eye-tracking study to verify that hypothesis [9]. Neither work touched on questions of evaluation.

Buckley and Voorhees's *bpref* evaluation measure [10] is calculated by summing the number of relevant documents ranked above nonrelevant documents. It suggests the idea of preferences, but it is defined over absolute judgments. The calculation of bpref entails inferring that each relevant document is preferred to every nonrelevant document, but all the relevant documents are "tied": none is preferred over any other.

Mizzaro has proposed measures of assessor agreement for both absolute and preference judgments [11], but we could find no work that empirically evaluated whether assessors tend to agree more on one or the other as we do here.

In a study that lends support to this work, Rorvig made the case for preference-based test collections using an idea from mathematical psychology known as

"simple scalability" [12]. He argued that, despite their high cost, preference judgments are an imperative for tasks for which the goal is to find highly-relevant documents. Rorvig showed that necessary conditions for the application of simple scalability held in practice, but we were unable to find any follow-up studies on preferences versus absolute judgments.

Thus to the best of our knowledge this is the first comparison of absolute judgments versus preference judgments in terms of assessor performance. It is also the first investigation into making preference judgments cost-effective by reducing the total number needed for evaluation of search engines.

3 Assessor Study

Our study investigated whether preferences are "easier" to make than absolute judgments by measuring inter-assessor consistency and time spent on each judgment. All judgments will be made on web pages retrieved by the Yahoo!, Google, and Microsoft Live search engines.

We compared three types of judgments: (1) absolute judgments on a five-point scale (Bad, Fair, Good, Excellent, Perfect); (2) binary preference judgments as described above; and (3) a stronger version of preference judgment in which the assessor can additionally say that he or she *definitely* preferred one page over another. To mitigate against assessors abstaining from hard decisions, neither preference type allowed an "equal" or "same" judgment. [4]

3.1 Experimental Design

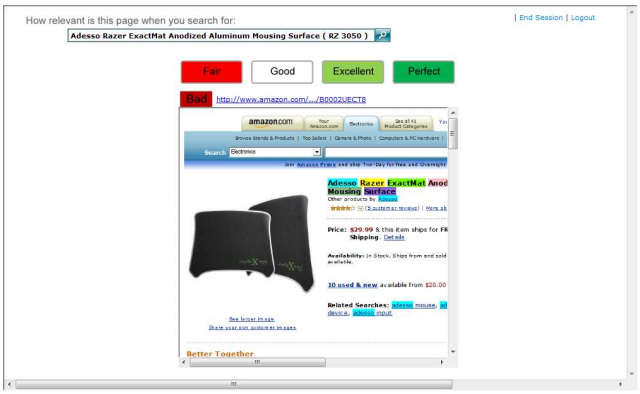
Measuring agreement requires that each query be seen by at least two different assessors for each of the three judgment types. Since an assessor cannot see the same query twice, we needed at least six assessors. Requiring that each assessor see every query imposed the constraint that assessors could not enter their own queries; the implications of this will be discussed in the next section.

Judging Interfaces We designed interfaces for each of the three judgments types. Screenshots for two of them are shown in Figure 1; the binary preference interface is identical to Figure 1(b) but excludes the "Definitely Here" buttons.

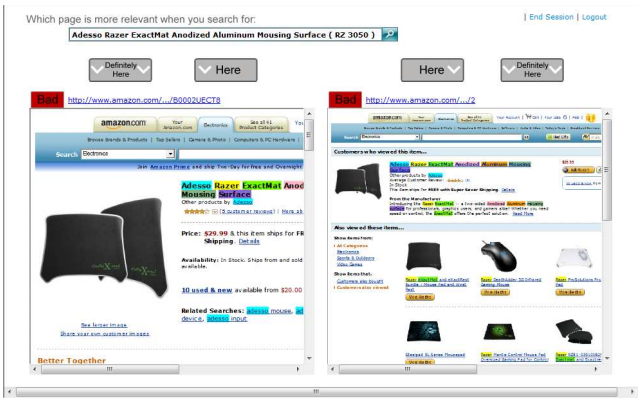
The query was shown at the top of the screen. If the assessor did not understand the query, he or she could obtain context by clicking on the magnifying glass button to see snippets from the top 20 web pages retrieved by a search engine. The order of these pages was randomized so as not to influence the assessor's judgments.

We allocated the same area to each web page in all three interfaces, regardless of whether one or two pages were shown. We highlighted the query terms that we found in a simple parse of the web page to make it easier for the judge to find relevant content.

[4] A followup study that included a "duplicates" judgment on pairs showed results consistent with those described in the next section.



(a) Absolute judgments.



(b) Preference judgments.

Fig. 1. Screenshots of the judgment interfaces.

Queries We sampled 51 queries from Microsoft's Live Search query logs. We chose queries that had previously been judged for the purpose of assessing search engine quality; in particular, we selected a biased sample that had some diversity in existing judgments, but was in other respects random. Some of the queries had clear intent, but most were vague, underspecified, or had myriad possible intents. The queries can generally be considered "informational"; examples include "andie mcdowell", "binghamton", "soda pop and oral hygiene".

Assessors The six assessors were Microsoft employees. All assessors had backgrounds in information retrieval or related fields and had experience in judging web pages for relevance.

Web Pages For each query, we took the top five web pages retrieved by three large search engines. The number of unique pages for a query depended on

the diversity of results retrieved by the engines: if all three retrieved the same documents, there would only be 5 pages to judge, but if they all retrieved different documents, there would be 15 pages. There were on average 11.9 unique pages per query, indicating a high degree of diversity among the top ranked results.

We did not remove web pages that had duplicate content but different URLs. We made this decision because there were some cases in which the URL provided some valuable additional information that helped judge relevance. For example, one query specified a product number. Two identical pages about the product were retrieved. Neither page contained the product number, but it was part of one of the URLs. It is important to note, however, that this will be the source of some disagreements in the preferences judgments.

In order to avoid time delays or temporary internet outages, we pre-captured web pages by saving images of them to disk. This also guaranteed that the pages were saved at a fixed point in time and the experiment could be reproduced.

Judgments As shown in Figure 1, the three interfaces had buttons along the top for judging documents, as well as a "Bad" button at the top left of the displayed web pages (in the same location relative to the web page for consistency). In the preference interface, a "Bad" judgment could be used for pages that were clearly not relevant, had not been properly loaded, or were spam. A page labeled "Bad" would not be seen again for any subsequent preference judgment.

We gave assessors guidelines explaining differences between relevance labels. The guidelines included the topicality of the page as well as the ease of finding relevant content on the page, trust in the domain, the likelihood that the page reflects the intent of the query, and so on. Assessors used the same guidelines for both absolute and preference judgments.

We fixed pages in a random order prior to any judging. Assessors made absolute judgments in that order. For preference judgments, the first two pages in the fixed order were presented first. The next judgment retained the preferred page and asked for its preference over the next page in the fixed order. When all comparisons involving the preferred page were exhausted, judgments restarted with the next two pages in the fixed order.

3.2 Results and Analysis

Agreement There are two types of agreement: agreement between two assessors over all judgments, or agreement about each judgment over all assessors. We chose to look at the latter in an attempt to average out differences in expertise, prior knowledge, or interpretation of the query.

Agreement for absolute judgments is shown in Table 1. Each cell (J_1, J_2) is the probability that one assessor would say J_2 (column) given that another said J_1 (row). They are normalized by row, which is why columns do not add to 1. The percentage of pages with each label is $20\%, 28\%, 25\%, 25\%, 2\%$, for Bad, Fair, Good, Excellent, and Perfect, respectively.

Agreement for preference judgments is shown in Table 2(a). For comparison, we inferred preferences from the absolute judgments: if the judgment on page A

	Bad	Fair	Good	Excellent	Perfect	Total
Bad	0.579	0.290	0.118	0.014	0.000	221
Fair	0.208	0.332	0.309	0.147	0.003	307
Good	0.095	0.348	0.286	0.260	0.011	273
Excellent	0.011	0.167	0.264	0.535	0.022	269
Perfect	0.000	0.042	0.125	0.250	0.583	24

Table 1. Assessor agreement for absolute judgments.

	$A < B$	A, B bad	$A > B$	Total
$A < B$	0.752	0.033	0.215	2580
A, B bad	0.208	0.567	0.225	413
$A > B$	0.201	0.034	0.765	2757

(a) Preferences.

	$A < B$	A, B bad	$A > B$	Total
$A < B$	0.657	0.051	0.292	2530
A, B bad	0.297	0.380	0.323	437
$A > B$	0.278	0.053	0.669	2654

(b) Inferred preferences.

Table 2. Assessor agreement for actual (a) and inferred (b) preference judgments.

	$A \ll B$	$A < B$	A, B bad	$A > B$	$A \gg B$	Total
$A \ll B$	0.247	0.621	0.000	0.132	0.000	219
$A < B$	0.059	0.661	0.043	0.221	0.015	2288
A, B bad	0.000	0.244	0.453	0.300	0.002	406
$A > B$	0.012	0.212	0.051	0.670	0.055	2389
$A \gg B$	0.000	0.180	0.005	0.680	0.134	194

Table 3. Assessor agreement for definite preference judgments.

was greater than the judgment on page B, we inferred that A was preferred to B. To compare to true preferences, we had to assign some preference to pairs of pages that were given the same label ("ties"). Table 2(b) gives results when the assigned preference is random (i.e. the expected value of a coin flip), simulating an assessor that makes a random judgment about which of two similar pages is preferred.

Statistical significance between Tables 2(a) and 2(b) can be measured by a χ^2 test comparing the ratio of the number of pairs agreed on to the number disagreed on for both preferences and inferred preferences. The difference is significant ($\chi^2 = 143, df = 1, p \approx 0$).

We can also explore redistributing ties at different rates to model different levels of agreement. Up to about 70% agreement on ties, true preference agreement is still significantly greater than inferred preference agreement. Above 80%, inferred preference agreement is significantly greater.

Agreement for the two-level "definite" preferences is shown in Table 3. Assessors do not appear to have been very consistent in their use of the "definitely" judgment. When the definite judgments are pooled together with the preference judgments (i.e. $A < B$ and $A \ll B$ treated as identical), the agreement is slightly less than in Table 2(a), but more than Table 2(b).

	Preference	Definite	Absolute	Overall
Assessor 1	3.50	3.41	7.96	3.70
Assessor 2	3.24	3.67	6.12	3.55
Assessor 3	2.35	2.82	5.56	2.82
Assessor 4	4.13	4.30	8.78	4.71
Assessor 5	2.72	3.30	8.20	3.17
Assessor 6	2.09	2.40	3.21	2.31
Overall	2.87	3.15	6.33	3.23

Table 4. Median seconds per judgment by each assessor in each interface.

Time Table 4 shows the median number of seconds spent on each judgment by each assessor for each interface, along with overall medians for each assessor and for each interface. [5] Absolute judgments took about twice as long to make as preferences. As the table shows, there was little variance among assessors.

Two main variables affect the time it takes to judge a page or a pair of pages: time spent reading the page(s) and time spent deciding on the correct judgment. One reason that preferences could be faster is that the assessor "memorizes" the page, or at least forms an impression of it, so that he or she does not have to re-read it each time it appears. If this were the case, judgments would get faster as each document had been seen.

To investigate this, we looked at the time each assessor spent making a judgment the first time the page was shown. For the preference, definite, and absolute judgments, the median time spent on a judgment when seeing a document for the first time was 3.89, 5.40, and 6.33 seconds, respectively. Thus it seems that making a preference judgment is faster than making an absolute judgment even after taking reading time into account.

Additional Analysis The "context search" button, which allowed assessors to see twenty search results, was used a total of 41 times, slightly under once per seven queries. There was no correlation between the judgment interface and use of context search.

After each query, assessors were presented with a feedback page to report their confidence in their judgments and their understanding of the query on a scale of 1 to 5, with 1 being least confident and 5 most. The median for both questions was 4, and despite not having "ownership" of queries there was no significant correlation between confidence and time spent judging.

4 Evaluating Engines

With preference judgments, standard evaluation measures like average precision and DCG can no longer be used. We must develop new evaluation measures.

[5] Median is reported instead of mean due to hours-long outlying inter-judgment times that skewed the means upward.

	NDCG	ppref	wpref
DCG	0.748	0.485	0.584
NDCG		0.662	0.738
ppref			0.950

(a) Correlation between evaluation measures.

	NDCG	ppref	wpref
DCG	1.000	0.873	0.866
NDCG		0.873	0.866
ppref			0.941

(b) Agreement on system differences.

Table 5. Comparisons between evaluation measures defined over absolute judgments and measures defined over preferences.

A simple but intuitive measure is the proportion of pairs that are correctly ordered by the engine. We call this "precision of preferences" or *ppref* for short. More formally, over all pairs of pages i, j such that i is preferred to j by an assessor, ppref is the proportion for which the engine ranked i above j. If neither i nor j is ranked, ppref ignores the pair. If i is ranked but j is not, ppref considers i to have been ranked above j.

The pairs in ppref can be weighted for a measure we call *wpref*. We use a rank-based weighting scheme: for pages at ranks i and j such that $j > i$, let the weight $w_{ij} = \frac{1}{\log_2(j+1)}$. wpref is then the sum of weights w_{ij} over pairs i, j such that i is preferred to j and the rank of i is less than the rank of j. The normalizing constant is the sum of all weights w_{ij}.

4.1 Results

We compared evaluations between four different measures: DCG, normalized DCG (NDCG), ppref, and wpref. A common formulation of DCG is $DCG@k = \sum_{i=1}^{k} \left(2^{rel_i} - 1\right) / \log_2(i+1)$ [5], where rel_i is the relevance of the document at rank i. NDCG@k is DCG@k divided by the DCG of the top k most relevant documents ranked in descending order of relevance.

DCG and NDCG were calculated over both sets of absolute judgments obtained for each query. Since assessor disagreement could be a source of variance in a comparison between absolute measures and preference measures, we calculated ppref and wpref over the preferences inferred from the absolute judgments.

Pearson correlations among the four measures calculated for each query are shown in Table 5(a). The absolute-based measures correlate well, and the preference-based measures correlate well. The correlation between wpref and NDCG is nearly as high as the correlation between DCG and NDCG.

We can also measure "agreement" among the measures in determining whether one system is better than another. We calculate each measure for each query and each system, then look at the sign of the difference between two measures on each query. If both measures say the difference is positive or negative, they agree; otherwise they disagree. As Table 5(b) shows, the measures agree at a fairly high rate, though preference measures agree more with each other than they do with absolute measures.

5 Efficient Judging

One of the biggest obstacles to the adaption of preference judgments, is that the number of document pairs increases polynomially with the number of documents. Although we had at most 15 documents for any query (105 preferences), in a large-scale evaluation there would likely be dozens or hundreds, as pages are drawn from different engines and different test algorithms. A polynomial increase in the number of judgments means much greater cost in assessor time, no matter how much faster assessors are at judging. In this section we look at ways to reduce the number of judgments required.

5.1 Transitivity

If assessors are consistently transitive, the full set of judgments is not necessary; this is the idea behind comparative sorting algorithms such as heapsort. The rate of growth in the number of comparisons needed by these algorithms is in $\mathcal{O}(n \lg n)$, much slower than the $\mathcal{O}(n^2)$ growth rate of all comparisons.

To evaluate transitivity, we iterated over all triplets of documents i, j, k in each set of preference judgments. We counted the number of times that, if i was preferred to j and j was preferred to k, the assessor also preferred i to k.

Transitivity holds for over 99% of triplets on average. Each individual assessor was consistently transitive at least 98% of the time. This suggests we can use a sorting algorithm with a minimum of information loss, and possibly improve assessor consistency at the same time. This agrees with Rorvig's finding that preference judgments of relevance are transitive [12]. Figure 2 shows the $\mathcal{O}(n \lg n)$ growth rate compared to the $\mathcal{O}(n^2)$ rate.

5.2 "Bad" Judgments

In Section 3 we discussed the use of "Bad" judgments in the preference interface. About 20% of absolute judgments were "Bad". Since we can reasonably assume that nothing will be preferred to these pages, we can additionally assume that every non-"Bad" page would be preferred to any "Bad" page. Therefore each "Bad" judgment gives us $\mathcal{O}(n)$ preferences.

The empirical reduction in judgments by inferring preferences in this way is shown in Figure 2. At $n = 15$, this has reduced the number of judgments to about 40 (averaged over all queries and all assessors that were assigned a preference interface for that query). The average decrease from $\mathcal{O}(n \lg n)$ over all values of n is 16 judgments.

The curve appears to be increasing at a rate of $n \lg n$, though it is not clear what it will do as n continues to increase beyond 15. Presumably increasing n results in a greater proportion of bad pages, so it may be that the curve asymptotically approaches a linear increase.

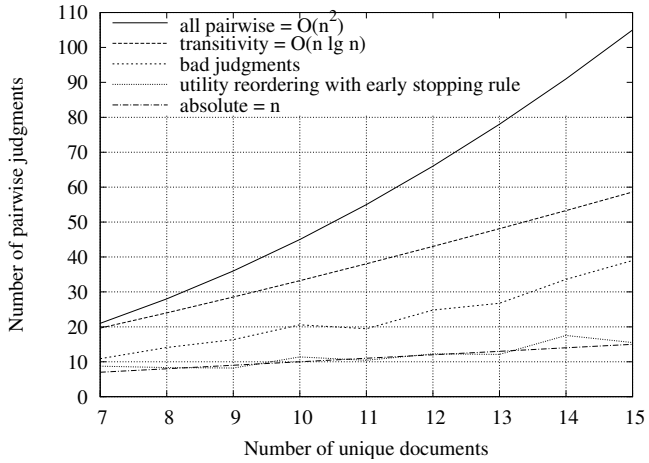


Fig. 2. Number of judgments made by assessors under different conditions.

5.3 Cost-Effective Judgments

Applying transitivity and "Bad" judgments still gives us the full set of preference judgments, though some are inferred rather than asked of the assessor directly. There are additional steps we can take to increase the utility of the assessors' time, and, even with some judgments unmade, still prove that differences between systems exist on a particular set of queries. This is based on the work of Carterette et al. [13], who showed that by estimating the utility of a possible judgment and bounding performance differences, the relative performance of systems could be determined with very little effort.

Estimating Utility Each judgment has a particular utility in helping us determine the sign of the difference in a measure over each engine. For example, if ppref is the measure of interest, and engines E_1 and E_2 both rank document A above document B, then whether A is preferred to B or not is of no consequence: the difference in ppref between the two engines will be the same regardless.

Furthermore, since transitivity holds, each judgment we make may bring additional transitive judgments along with it. For example, if we have already judged that A is preferred to B and we are debating whether to next judge (B, C) or (A, C), we should keep in mind that if we judge $B > C$, we can infer $A > C$ by transitivity; likewise, if we judge that $C > A$, we can infer $C > B$. As above, whether these transitive judgments are useful depends on how the documents are ranked by the systems.

The utility function for a preference measure is:

$$U(A, B) = p(A > B) \cdot gain(A > B) + p(B > A) \cdot gain(B > A), \text{ where}$$

$$
\begin{aligned}
gain(A > B) = & \left| w_1(A,B) sgn\left(r_1(A) - r_1(B)\right) - w_2(A,B) sgn\left(r_2(A) - r_2(B)\right) \right| \\
& + \sum_{i|B>i} \left| w_1(A,i) sgn\left(r_1(A) - r_1(i)\right) - w_2(A,i) sgn\left(r_2(A) - r_2(i)\right) \right| \\
& + \sum_{i|i>A} \left| w_1(i,B) sgn\left(r_1(i) - r_1(B)\right) - w_2(i,B) sgn\left(r_2(i) - r_2(B)\right) \right|
\end{aligned}
$$

The sums are over pairs (i, B) such that we had previously judged that $B > i$ and (i, A) where we had judged $i > A$. These capture the transitive judgments discussed in the paragraph above. The weights $w_n(i, j)$ are set for an evaluation measure: $w_n(i, j) = 1$ gives the utility function for ppref, while $w_n(i, j) = 1/\log_2\left(\min\{r_n(i), r_n(j)\} + 1\right)$ (where $r_n(i)$ is the rank of document i by system n) produces the utility function for wpref.

Note that the expected utility relies on an estimate of the probability that A is preferred to B. We assume a priori that this probability is $\frac{1}{2}$. After we have made some judgments involving A and some judgments involving B, we may have more information. We can use a simple logistic regression model such as [14] to estimate these probabilities with no features; the model can easily be adapted to incorporate any feature.

By judging pairs in decreasing order of utility, we can ensure that after k judgments we have the most possible confidence in the difference between two systems. The next question is how big k has to be before we can stop judging.

Early Stopping Rule Suppose after partially completing judgments, the ppref of E_1 is greater than that of E_2 (excluding unjudged pairs). If there is no possible set of judgments to the remaining pairs that would results in E_2 "catching up", we can safely stop judging.[6]

Although it is difficult to determine the exact point at which we are guaranteed that E_1 must be superior to E_2, we can easily compute bounds on $E_1 - E_2$ that allow us to stop judging before evaluating all pairs. A very simple bound iterates over all unjudged pairs and assigns them a judgment depending on how much they would "help" either engine. If we have a pair i, j such that E_1 ranked i above j but E_2 ranked j above i, then we want to know what happens if j is preferred to i, i.e. that pair helps E_2 and hurts E_1. We assign judgments in this way for all pairs, ignoring consistency of judgments. This gives us a loose bound.

The number of judgments that are required to differentiate between systems after applying dynamic reordering based on expected utility and the early stopping rule is shown in Figure 2. The number of judgments has effectively been reduced to n on average. Reordering and early stopping can be applied to absolute judgments as well, but the gain is not nearly as dramatic: on average it results in only 1–2 fewer judgments per query.

Although there is no guarantee our results would continue to hold as n increases, we can guarantee that using "Bad" judgments and transitivity will give

[6] If we need all of the judgments in order to train a ranking algorithm, on the other hand, we may not want to stop.

us a slower rate of increase than making all preference judgments, and that using dynamic reordering and the early stopping rule will give us an even slower rate of increase. Furthermore, utility-based reordering produces that a set of judgments is maximally useful no matter when the judging effort is stopped.

6 Conclusion

We have performed the first investigation into the direct acquisition of preference judgments for relevance and the first comparison of preference judgments to absolute judgments. We have also provided a suite of methods by which preference judgments can become practical to use for evaluation of search engines.

There are several clear directions for future work: choosing the correct evaluation measure for preferences, the robustness of these measures to missing preferences, and measuring the uncertainty in an evaluation when preferences are missing are three. Additionally, whether training ranking algorithms over preference judgments rather than inferred preferences results in more robust performance is an interesting open question.

References

1. Voorhees, E.M., Harman, D., eds.: TREC. The MIT Press (2005)
2. Järvelin, K., Kekäläinen, J.: IR evaluation methods for retrieving highly relevant documents. In: Proceedings of SIGIR. (2000) 41–48
3. Voorhees, E.: Variations in relevance judgments and the measurement of retrieval effectiveness. In: Proceedings of SIGIR. (1998) 315–323
4. Kendall, M.: Rank Correlation Methods. Fourth edn. Griffin, London, UK (1970)
5. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: Proceedings of ICML. (2005) 89–96
6. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of KDD. (2002) 133–142
7. Bartell, B., Cottrell, G., Belew, R.: Learning to retrieve information. In: Proceedings of the Swedish Conference on Connectionism. (1995)
8. Frei, H.P., Schäuble, P.: Determining the effectiveness of retrieval algorithms. Information Processing and Management **27**(2-3) (1991) 153–164
9. Joachims, T., Granka, L., Pang, B., Hembrooke, H., Gay, G.: Accurately interpreting clickthrough data as implicit feedback. In: Proceedings of SIGIR. (2005) 154–161
10. Buckley, C., Voorhees, E.M.: Retrieval evaluation with incomplete information. In: Proceedings of SIGIR. (2004) 25–32
11. Mizzaro, S.: Measuring the agreement among relevance judges. In: Proceedings of MIRA. (1999)
12. Rorvig, M.E.: The simple scalability of documents. JASIS **41**(8) (1990) 590–598
13. Carterette, B., Allan, J., Sitaraman, R.: Minimal test collections for retrieval evaluation. In: Proceedings of SIGIR. (2006) 268–275
14. Carterette, B., Petkova, D.: Learning a ranking from pairwise preferences. In: Proceedings of SIGIR. (2006) 629–630