Ranking From Pairs and Triplets: Information Quality, Evaluation Methods and Query Complexity

Kira Radinsky
Computer Science Department
Technion–Israel Institute of Technology
kirar@cs.technion.ac.il

Nir Ailon
Computer Science Department
Technion–Israel Institute of Technology
nailon@cs.technion.ac.il *

ABSTRACT

Obtaining judgments from human raters is a vital part in the design of search engines' evaluation. Today, a discrepancy exists between judgment acquisition from raters (training phase) and use of the responses for retrieval evaluation (evaluation phase). This discrepancy is due to the inconsistency between the representation of the information in both phases. During training, raters are requested to provide a relevance score for an individual result in the context of a query, whereas the evaluation is performed on ordered lists of search results, with the results' relative position (compared to other results) taken into account. As an alternative to the practice of learning to rank using relevance judgments for individual search results, more and more focus has recently been diverted to the theory and practice of learning from answers to combinatorial questions about sets of search results. That is, users, during training, are asked to rank small sets (typically pairs).

Human rater responses to questions about the relevance of individual results are first compared to their responses to questions about the relevance of pairs of results. We empirically show that neither type of response can be deduced from the other, and that the added context created when results are shown together changes the raters' evaluation process. Since pairwise judgments are directly related to ranking, we conclude they are more accurate for that purpose. We go beyond pairs to show that triplets do not contain significantly more information than pairs for the purpose of measuring statistical preference. These two results establish good stability properties of pairwise comparisons for the purpose of learning to rank. We further analyze different scenarios, in which results of varying quality are added as "decoys".

A recurring source of worry in papers focusing on pairwise comparison is the quadratic number of pairs in a set of results. Which preferences do we choose to solicit from paid raters? Can we provably eliminate a quadratic cost? We employ results from statistical learning theory to show that the quadratic cost can be provably eliminated in certain cases. More precisely, we show that in order to obtain a ranking in which each element is an average of $O(n/C)$ positions away from its position in the optimal ranking, one needs to sample $O(nC^2)$ pairs uniformly at random, for any $C > 0$. We also present an *active learning* algorithm which samples the pairs adaptively, and conjecture that it provides additional improvement.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Relevance feedback*

General Terms

Theory;Experimentation

Keywords

Ranking from pairs, Ranking Evaluation, Relevance feedback

1. INTRODUCTION

Evaluation and training are essential to information retrieval (IR). Training refers to the process of obtaining feedback in order to train and improve the system. This feedback is obtained either explicitly, by asking people to provide it, or implicitly, by analyzing search engine traffic logs. In this work we refer only to explicit training. Evaluation refers to the process of measuring the goodness of the system. Up until now, most of the IR literature has evolved around systems in which a systematic discrepancy exists between the training and the evaluation methods, which we explain here briefly. In training, raters are asked, given a query and a search result, to provide a graded relevance score for the result in the context of the query (see survey [32]). In evaluation, the quality of a set of ordered search results for a test query is scored using one of a family of metrics: AP (Average Precision), DCG [22] and NDCG (normalized DCG), RBP (Ranked Bias Precision) [28], MMR (Maximum Marginal Relevance) [9], ERR (Expected Reciprocal Rank) [12]). All of these metrics favor orderings which place higher relevance test results in preferable positions, compared to lower relevance test results. [1] Because the metrics take into account relative relevance of results in an ordered list, the problem has become known as the ranking problem for information retrieval.

The discrepancy implied above is due to the inconsistency between the representation of the information in the training system and in the evaluation phase. In the training system, the basic token of information is *relevance judgment* on individual search results, whereas the evaluation is on ordered lists of search results, where the results relative position is taken into account. The ability of these metrics to measure the true goodness of a retrieval system

*Work supported by Marie Curie Grant PIRG-GA-2010-268403.

[1] The lists of results are always with respect to the same query. Evaluations are summed up over the different queries.

WSDM'11, February 9–12, 2011, Hong Kong, China.
Copyright 2011 ACM 978-1-4503-0493-1/11/02 ...$10.00.

[12, 27] is debatable. From the theoretical aspect, directly applying optimization techniques to maximize these metrics is computationally difficult [36].

As an alternative to the practice of learning to rank using relevance judgments for individual search results, more and more focus has recently been diverted to the theory and practice of learning from answers to combinatorial questions about sets of search results. More precisely, given a set of n items to rank (search results for the same query), subsets of size k are chosen from the set of all subsets. In the typical case of *pairwise preference information*, k equals 2. In the case of $k = n$, this idea has become known as the *listwise* approach [35, 26, 8]. The raters are asked a combinatorial question related to the correct order of the subset. Possible questions are, *Which is the best result in the tuple?* and *What is the correct way to order the results in decreasing relevance order?* Roughly speaking, the system trains a model for predicting the human responses. In evaluation, it is no longer possible to use the IR measures mentioned before, because the raters did not provide us with graded relevance scores. Given a query and an unordered set of search results, the evaluation score should measure the extent to which the tuples match the predicted human response.

In this work, we discuss the following questions arising from the aforementioned combinatorial approach.

- **Information.** Does human response to comparative combinatorial questions on k-sets contain information that differs from that contained in relevance score responses? Does the information contained in responses to sets of size $k = 2$ (pairs) subsume information coming from higher k?

- **Query Complexity.**[2] Which subsets do we choose from the possible $\binom{n}{k}$ to send to raters in training? Do we need to send all $\Omega(n^k)$ possibilities?

- **Evaluation.** How do we evaluate an ordering of search results in testing?

- **Computational Complexity.** How do we find the best ordering with respect to this evaluation function?

This work mainly tackles the questions of *information, query complexity*, and *evaluation*:

- **Information.** We show in Section 3 that, for the purpose of ranking, pairwise preference judgments contain more accurate information than relevance scores on individual results. In fact, we show that relevance judgments do not satisfy *independence* and *consistency*, two natural requirements assumed in all Cranfield experiments [32]. We go beyond pairs to show that the marginal preference information for pairs within a triplet does not differ much from the information obtained from the pair when presented alone. These two results indicate the stability of pairwise information within higher order tuples. To the best of our knowledge, this is the first time this kind of study has been performed.

- **Query Complexity.** In Section 4 we analyze a standard evaluation function running over all $\binom{n}{2}$ pairs. The quadratic dependence of the score in the size of the input should not scare practitioners. We employ results from statistical learning theory to show that the quadratic cost can be provably eliminated in certain cases. More precisely, we show that in order to obtain a ranking in which each element is an average of $O(n/C)$ positions away from its position in the optimal ranking, one needs to sample $O(nC^2)$ pairs uniformly at random, for any $C > 0$. We also present an *active learning* algorithm which samples the pairs adaptively, and conjecture that it provides additional improvement in the query complexity. We also discuss a variant of the cost function that favors top results, in the spirit of, e.g., [31] and the wpref evaluation function from [7, 11].

- **Evaluation.** Our goal here is not to claim that one objective function is better than the other for the purpose of comparing IR systems - we leave this to other empirical studies. We do claim that there should be a match between the manner in which information is fed into the system (e.g. individual relevance scores, pairwise preference or 5-tuple full rankings) and the manner in which it is evaluated. This is not only intuitive, but also allows us to use tools from statistical learning theory in order to bound the regret incurred from subsampling the information, in an attempt to escape quadratic (in the pairwise case) or higher degree polynomial (for larger sets) complexity.[3] We choose a ranking evaluation function with a quadratic number of summands (as a function of results retrieved for a query) to show how the analysis can be done.

2. BACKGROUND AND MOTIVATION

2.1 Drawing from Econometric Theory

The question of relevance level of search results can be compared to the econometric approach of assigning *utility* to goods. Using responses to the question *which is the best result?* to build a retrieval system can be compared to the econometric study of *discrete choice* for predicting consumer behavior [34]. A typical approach in the latter theory is the *random utility model*, in which one posits that a discrete choice is equivalent to the process of selecting the maximum coordinate from a random vector of alternative utilities. Analogously, many papers in the IR ranking literature try to model ranking by inducing it from a machine learned utility function, but few papers use human stated choice as their source of information.[4] Instead, they (e.g., [35, 8, 20]) may induce choice or preference information from relevance responses to individual results, which are merely a form of a latent score or perceived utility (albeit quantized to fit some graded scale). The net result is a learning system using scores both for learning and for modeling. Why, in that case, go via ranking? We find this to be a questionable detour, which we refer to in what follows as the *IR ranking detour* (Figure 1). One of the results in this paper shows that relevance responses from raters contains information that differs from their stated preference. The IR detour hence adds noise. Additionally, a result by Ailon [1] illustrates that given relevance responses, a regression algorithm predicting the relevance directly does at least as well as ranking algorithms, including the aforementioned *listwise* algorithms.

[2] The term *query* is overloaded in this work. It means (1) a request for information from an IR system, and (2) soliciting a preference response from a human rater. The combination *query complexity* always refers to the latter interpretation.

[3] By regret we mean the additional cost we pay compared to the optimal solution to the utility function given all (quadratically much) pairwise preference information.

[4] A noteworthy exception is Joachim's work [23] in which click-through data is used as expression of discrete choice from a set, although this, in econometric language, is *revealed choice* and not *stated choice*, which we study here.

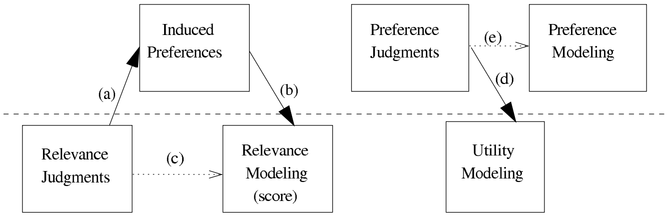


Figure 1: The IR detour is represented by (a)+(b) in the drawing, where (c) represents a direct approach (e.g., risk minimization in ordinal classification or regression). Econometric theory of discrete choice is represented in (d). Process (e) represents the first of a two stage approach: machine learning to predict preferences, followed by combinatorial optimization for ranking (not drawn).

We present in this work an empirical study quantifying some problematic aspects of the IR Detour.

2.2 Avoiding the Detour

There are two obvious direct schemes that avoid the detour, both of which call for new IR evaluation metrics. One scheme, illustrated in arrow (c) in Figure 1, is to treat the problem as an ordinal classification problem, and solve it using ordinal regression or structural risk minimization, as done in, e.g., [17, 1]. The second alternative (arrow (e) in the figure) is the combinatorial approach mentioned in the introduction, in which the system is trained and evaluated against rater responses to combinatorial questions involving sets of size k.

Li and Cao et al. [35, 8, 20] use the combinatorial listwise approach with $k = n$. In other words, raters provide a full ranking of search results for a query. This can be a daunting task for humans, especially if the number of results is large [29]. In their real data experiments, in fact, the authors take the IR detour. That is, users are requested to give relevance judgments to individual results for each query, and a ranking is induced from these results.

In this work we study an approach which avoids the IR detour. Pairwise preference judgments (corresponding to $k = 2$) are an alternative receiving much attention [2, 5, 15, 21]. Already for this value of k, the questions of *information, query complexity, evaluation* and *computational complexity* complicate matters. The question of query complexity was raised in, e.g., [17] as a reason to avoid the approach, while others [11, 18] offer interesting heuristics for dealing with it. In this study, we initiate a rigorous analysis to solve this problem, which to the best of our knowledge has not been done before. We prove a theoretical statistical learning bound, from which one can derive a sub-quadratic query complexity algorithm in certain cases. We conjecture that an active learning approach which we present as an adaptive sampling algorithm does even better. The underlying evaluation function optimized in our algorithm is chosen to be a simple cost function running over all pairwise inconsistencies of an output ranking of items (search results). This function is tightly connected with *bpref* and *wpref* defined in [7, 11], and a similar analysis could be done for those functions as well. In spite of its quadratic nature, we conjecture that an algorithm outlined in Figure 5 which adaptively samples only $O(n \operatorname{polylog} n)$ pairs returns a result with negligible regret compared to the optimum. This algorithm is based on a recent PTAS for a similar problem by Kenyon-Mathieu et. al [24].[5] From a computational point of view, a black box for solving the NP-hard problem of minimum feedback arc-set is required in the algorithm of Figure 5. We do not address this black box in the paper. We concentrate on query complexity only. We refer the readers to known fast heuristics (such as [16]) and to work on optimizing the problem in the stochastic settings [6].

Carterette et al. [11] empirically show that assessors tend to agree more and spend less time per judgment when asked preference judgments of the form "document A is more relevant than document B." In our work we also compare statistical differences between both type of responses. Additionally, we study cases in which assessors are asked to provide two relevance scores for a pair of results simultaneously (as opposed to soliciting the scores separately from distinct raters). We check whether the marginal distribution of the individual scores fits the distribution obtained by asking for scores separately. In order to simulate the first step in the IR detour (Figure 1), we also check whether the preference distribution *induced* by responses to separate questions fits the distribution obtained when directly asking for preference. We then go beyond the pairwise setting and study triplets of results, to see how the additional context affects the induced statistical preference for the two other results. We study different types of context, including pseudo-results obtained by obfuscating other results.

3. EXPERIMENTAL EVALUATION

We perform numerous experiments on human subjects to check whether for the purpose of ranking, information obtained from result pairs is substantially different from that obtained from single results for a query. Furthermore, we check whether information from triplets differs from that of pairs. In our experiments, we ask human raters to respond to both relevance questions (corresponding to individual search results) and comparative questions (corresponding to sets of 2 or 3 results for the same query). The sample we choose to present is extracted from a comprehensive variety of results retrieved from commercial search engines, from both high and low positions.

3.1 Evaluation Procedure

Experiments are performed using 50 queries obtained from the TREC Web Track [13]. For each query, we obtained results from Google's search engine. We refer to several types of results in our experiments:

1. Given a query q, a result r is defined as a *Low Result*, denoted $r \in Low$, if it is retrieved at position 200 or worse by Google's search engine given query q. For each query we sample 10 *Low* results.

2. Given a query q, a result r is considered to be a *High Result*, denoted $r \in High$, if it is retrieved at position 10 or better by Google. For each query we sample 10 *High* results.

3. A result $r' = Obf(r)$ is defined to be an *Obfuscated Version* of a result r if it is obtained from r by introducing grammar and syntax errors. Here we obfuscate by translating the snippet of result r to Italian and then back to English using the "Google translate service", taking advantage of imperfections in current state-of-the-art automatic translation technology.[6][7]

[5] A PTAS is shorthant for Polynomial-Time Approximation Scheme: An algorithm which returns a solution of cost at most $(1 + \varepsilon)$ that of an optimal solution.

[6] `http://translate.google.com`

[7] By snippet we mean a short summary of a retrieved page presented to the user before navigation to the actual page.

Experiments were conducted using Amazon Mechanical Turk, an emerging utility for performing user study evaluations [25].[8] Each question in our study is incarnated as 10-30 *tasks*. In our terminology, one *task* refers to routing a question to a (paid, random, online) user and obtaining an answer. Tasks are structured using one of 5 templates (see below). The templates share a common structure, consisting of a query q and a set of results, which is either a singleton, a pair, or a triplet. Templates are further subdivided into *relevance* and *preference* types. The relevance type template solicits a relevance score in a 5-grade scale for each one of the results in the corresponding set. [9] The preference type template (for pairs and triplets only) asks for identification of the best result. We summarize the 5 template types. *For a query q:*

1. (single-relevance) *Assign a relevance grade to* r.
2. (pairwise-relevance) *Assign a relevance grade to* r_1, r_2.
3. (pairwise-preference) *Mark the better result from* r_1, r_2.
4. (triplet-relevance) *Assign a relevance grade to* r_1, r_2, r_3.
5. (triplet-preference) *Mark the best result among* r_1, r_2, r_3.

"To avoid bias towards a certain preference due to the position of the results in the template, we created tasks that permute pairs and triplets in all possible ways.

3.2 Results and Analysis

Our main statistical tool is Cochran-Mantel-Haenszel's (CMH) [14] repeated tests of independence. The test computes a statistic on a list of 2×2 integer matrices, used for accepting or rejecting the null hypothesis of independence between the row and column variables in all matrices. If the i'th histogram is defined as

$$\left(\begin{array}{cc} a_i & b_i \\ c_i & d_i \end{array} \right)$$

and $n_i = a_i + b_i + c_i + d_i$, the statistic is given as

$$\chi^2 = \frac{\{|\sum_i a_i - \frac{(a_i+b_i)(a_i+c_i)}{n_i}| - 0.5\}^2}{\sum_i \frac{(a_i+b_i)(a_i+c_i)(b_i+d_i)(c_i+d_i)}{(n_i^3 - n_i^2)}} . \qquad (3.1)$$

The subtracted 0.5 is Yate's correction. The significance of rejecting the null hypothesis is given by a p-value corresponding to the the χ^2 distribution (with 1 degree of freedom).

In our statistical analysis, we apply CMH to two types of histograms: **Relevance Histograms** and **Preference Histograms**.

Relevance Histograms: Each histogram corresponds to a result r for a query q. The rows correspond to tasks from two different templates. The columns correspond to a binary partitions of the possible relevance scores $\{1, 2, 3, 4, 5\}$. For example, we may assign $\{1, 2, 3\}$ to the first column and $\{4, 5\}$ to the second. Each cell counts how many times raters gave a relevance score contained in the set corresponding to the column, in the template corresponding to the row. For example, if the top row corresponds to single-relevance and the left column corresponds to $\{1, 2, 3\}$, then the top-left cell in the histogram counts how many times raters judged result r with a relevance score of either 1, 2 or 3 within a task formulated in the single-relevance template. For pairwise-relevance or triple-relevance templates, the relevance scores to the other one or two results are ignored (we take the marginal for r).

Preference Histograms: Each histogram corresponds to a pair of results r_1 and r_2 for the same query q. The columns correspond to comparison counts (how many times r_1 was preferred over r_2 versus how many times r_2 was preferred over r_1). The rows correspond to different templates which were used for obtaining these preferences. For a template of type single-relevance, the count is obtained as follows. Let h_i denote the histogram of the 5-grade relevance score given to result r_i for $i = 1, 2$. More precisely, $h_i(j)$ is the number of raters assigning relevance j for result i, where $j = 1, \ldots, 5$. We use the histograms to infer preference counts as follows. For $i \in \{1, 2\}$ let $\bar{i}$ denote $3 - i$ (the alternative index). If N_i denotes the inferred number of times that r_i is preferred over the alternative $r_{\bar{i}}$, then N_i is given as

$$N_i = \sum_{j=2}^{5} \sum_{j'=1}^{j-1} h_i(j) h_{\bar{i}}(j') .$$

We call (N_1, N_2) the *tie-ignoring* count. We also define an alternative counting method, called *tie-splitting* which divides ties equally among both alternatives, as in [11]. We denote this inferred count by $\tilde{N}$ and compute it as follows:

$$\tilde{N}_i = N_i + \frac{1}{2} \sum_{j=1}^{5} h_i(j) h_{\bar{i}}(j) .$$

The counting method for single-relevance corresponds to the *random utility* model in discrete choice theory, in which preference is given by comparing two *independent* random utilities and choosing the best.

For pairwise-relevance templates, let $h(j_1, j_2)$ denote the number of tasks for which r_i received a relevance score of j_i for $i = 1, 2$. The definitions of the tie-ignoring and the tie-splitting counts are clear:

$$N_i = \sum_{j_1=2}^{5} \sum_{j_2=1}^{j_1-1} h(j_i, j_{\bar{i}}), \;\; \tilde{N}_i = N_i + \frac{1}{2} \sum_{j=1}^{5} h(j, j) . \qquad (3.2)$$

For pairwise-preference templates, the counts are simply read from the task responses (no ties appear in the templates). For triplet-relevance templates, the relevance of the additional result r_3 is ignored (it is used only to create additional context). For triplet-preference templates, we ignore the cases in which the context creator r_3 was chosen as best.

Each histogram contains one template in the first row and another in the second row. The first experiment in Section 3.3 compares single-relevance task responses to those of pair-relevance. The second, in Section 3.4, compares pairwise-preference to triplet-preference. The third, in Section 3.5, compares pairwise-relevance with pairwise-preference. The experiment in Section 3.6 compares two triplet-preference scenarios, one with the context result r_3 satisfying $r_3 \in High$, and the other with $r_3 \in Low$. We perform a similar test for triplet-relevance. Finally, the experiment in Section 3.7 compares pairwise-preference with triplet-preference, where the contexts result r_3 is given as $Obf(r_1)$. A similar test is conducted for pairwise-relevance vs. triplet-relevance.

3.3 Are Single-Relevance Responses Stable for Ranking?

In classic IR training design (e.g., TREC [13]), results are evaluated by raters without any context: A result is shown to the human evaluator as a singleton. This approach was simulated with our single-relevance template. In the IR ranking literature [11, 10] there is much debate around whether preference or relevance judgments should be requested from evaluators. Indeed, singleton relevance

[8] `https://www.mturk.com`

[9] The template instructions define the scale grades as highly irrelevant(1), irrelevant, marginally relevant, relevant and highly relevant(5).

responses are believed to be inaccurate, and subject to individual interpretation of the graded scale.

We first applied CMH over 100 relevance histograms, two histograms corresponding to two possible $High$ results r for each one of 50 queries q. The first row in each histogram counted relevance from single-relevance template task responses, and the second row counted relevance from pairwise-relevance template task responses. We used the partition $\{1,2,3\} \cup \{4,5\}$ for the columns. We used 20 tasks for each r, q and for each of the two templates. The resulting p-value was 0.05. We then ran the same test with two Low results instead. We used the partition $\{1,2\} \cup \{3,4,5\}$ for this case (this was done because Low results generate too few responses of $4, 5$). We computed a p-value of 0.007. We therefore reject the null hypothesis, and conclude that marginal perceived relevance distribution of results in pairwise-relevance responses differs from the perceived relevance distribution from single-relevance responses. This means that added context affects relevance perception.

One might argue that this still shouldn't discourage us from using single-relevance responses for IR ranking. Indeed, maybe the preference relation induced from relevance responses *is* stable across single-relevance and pairwise-relevance responses. We test this hypothesis using preference histograms. Consider the case in which we wish to determine the ordering of two results r_1, r_2 for query q. Comparing single-relevance template task responses with pairwise template responses will indicate whether the classic IR human judgment approach is equivalent, for the purpose of ranking these two results, to an approach which presents each one of the two results as context for the other.

For each one of 50 queries we chose 4 results, 2 $High$ results and 2 Low results. For each single result we had 10 single-relevance template task responses. The two high results were then paired , as were the two low results; high and low results were not mixed. Each pair was sent to 10 pairwise-relevance template tasks and to 10 pairwise-preference tasks.

We computed CMH for both $High - High$ and $Low - Low$ pairings over the corresponding 50 histograms, using tie-ignoring counts. For both cases, the results provide strong evidence rejecting the null hypothesis, with p-value of 0.02. This means that we reject the assumption of independence.

These results in fact imply that the validity of taking the IR detour is questionable. Indeed, the random ordering induced by separately and independently drawing relevance scores for the two results from a population of raters is not the same as the random ordering induced by drawing both relevance scores simultaneously in the same task.

3.4 How much context do we need?

In the previous section, we saw that for the purpose of ranking two results, when showing an evaluator a result without context to which the evaluator can compare the result to, the ranking of a pair of results changes substantially. We now explore whether adding a context result as a "decoy" affects the induced preference for the pair.

We computed CMH over histograms built from 50 queries, with 2 $High$ results r_1, r_2 chosen for each query. We ran the experiment four times, once for each possible combination of the following criteria:

- $r_3 \in High$ / $r_3 \in Low$
- Histogram from pairwise-relevance vs triplet-relevance tasks / Histogram from pairwise-preference vs triplet-preference tasks. [10]

We adopted tie-ignoring in all tests.

Table 1: From Pairs to Triplets

Results Type	pairwise-**relevance** vs. triplet-**relevance** (p-value)	pairwise-**preference** vs. triplet-**preference** (p-value)
$r_3 \in High$	0.20	0.70
$r_3 \in Low$	0.9	0.23

The results (Table 1) suggest that relevance and preference judgments of $High$ result pairs do not differ substantially from result triplets obtained by adding a context of either $High$ or Low type. However, the relevance and preference judgments for triplets with $High$ added context are different from those judgments for triplets with Low added context; see Section 3.6.

3.5 Is Relevance the same as Preference?

In this section we wish to check whether as a result of the conclusion from Section 3.3 we should completely abandon relevance scores. To that end, we run CMH on histograms comparing pairwise-relevance vs pairwise-preference on 50 queries. In one experiment, we pair $High$ results with $High$ results, and in another we pair Low with Low. The preference count is done in a tie-ignoring manner for the pairwise-relevance task responses (this is obvious, because pairwise-preference template does not allow tied responses). Both tests returned a p-value in the range $[0.45, 0.47]$ indicating that the null hypothesis should not be rejected. This means that relevance scores, when provided in mutual context, probably do not contradict binary preference responses.

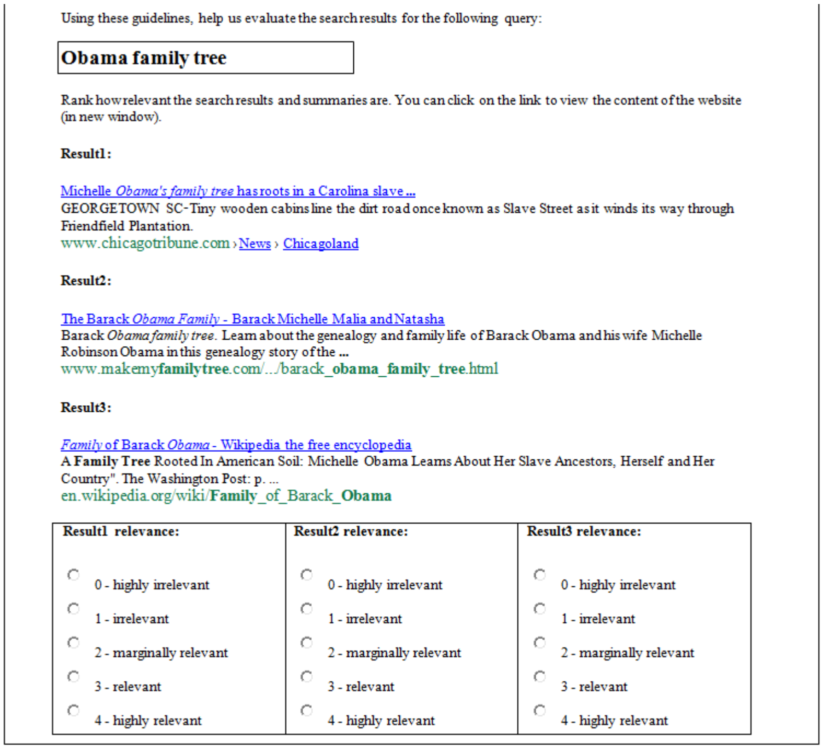
3.6 Different Context Types

The use of the term *context* in our work is analogous to the use of *anchoring* in a famous behavioral economics experiment by Ariely [4]. In his experiment, users were asked to choose an amount of money to gamble on. Before the game began, some users were exposed to big numbers, and others to small numbers. For example, subjects from the first group might be asked how many people live in China, whereas the second group might be asked for the number of Biblical commandments. The chosen gambling stake distribution for both groups significantly differed. Those exposed to bigger numbers tended to gamble on larger sums.

In ranking evaluation, we wish to explore the same phenomenon. Will the relevance of a pair result change in the context of an additional $High$ context result as opposed to a Low one?

We computed CMH over 50 queries, where for each query we chose a pair of $High$ results r_1, r_2, and for each pair we added different types of context. In one task collection we added $r_3 \in High$ (see Figure 2) and in another we added $r_3' \in Low$ (see Figure 3). We used the triplet-relevance template for the tasks. We routed each question to 30 raters, 5 for each possible permutation on 3 objects. Our 2×2 histogram for each query was obtained by placing in one row the responses with r_3, and placing in the other row the results with r_3'. When employing tie-splitting, we obtain a p-value of 0.04. When employing tie-ignoring, we obtain a p-value of 0.1. We also ran the same experiment with triplet-preference templates instead of triplet-relevance. We obtained a p-value of 0.1.

[10] Each pairwise question was routed to 10 tasks, 5 for each possible order, and each triplet question was routed to 30 tasks, 5 for each possible order.



Using these guidelines, help us evaluate the search results for the following query:

Obama family tree

Rank how relevant the search results and summaries are. You can click on the link to view the content of the website (in new window).

Result1:

Michelle *Obama's family tree* has roots in a Carolina slave ...
GEORGETOWN SC-Tiny wooden cabins line the dirt road once known as Slave Street as it winds its way through Friendfield Plantation.
www.chicagotribune.com › News › Chicagoland

Result2:

The Barack *Obama Family* - Barack Michelle Malia and Natasha
Barack *Obama family tree*. Learn about the genealogy and family life of Barack Obama and his wife Michelle Robinson Obama in this genealogy story of the ...
www.makemy**familytree**.com/.../barack_**obama_family_tree**.html

Result3:

Family of Barack *Obama* - Wikipedia the free encyclopedia
A **Family Tree** Rooted In American Soil: Michelle Obama Learns About Her Slave Ancestors, Herself and Her Country". The Washington Post: p. ...
en.wikipedia.org/wiki/**Family**_of_Barack_**Obama**

Result1 relevance:	**Result2 relevance:**	**Result3 relevance:**
○ 0 - highly irrelevant	○ 0 - highly irrelevant	○ 0 - highly irrelevant
○ 1 - irrelevant	○ 1 - irrelevant	○ 1 - irrelevant
○ 2 - marginally relevant	○ 2 - marginally relevant	○ 2 - marginally relevant
○ 3 - relevant	○ 3 - relevant	○ 3 - relevant
○ 4 - highly relevant	○ 4 - highly relevant	○ 4 - highly relevant

Figure 2: Results for the query “Obama family tree”. The first two results are a high results ($r_1, r_2, r_3 \in High$. The user is requested to submit three relevance judgements, but we ignore the response for r_3 because it is used as decoy here.

We consider these results as a weak rejection of the null hypothesis. In other words, the type of added context creates a noticeable but small difference for the purpose of comparing r_1 with r_2.

3.7 Using Obfuscated Results

Additional psychological experiments by Ariely [3] indicate that when 2 out of 3 alternatives are easily comparable to each other (but neither easily comparable to the third), people tend to go for the better of those two. For example, in one experiment, one group of people was presented with two options: vacation+breakfast in country A vs. vacation+breakfast in B, both for the same price. In the second group, people were faced, on top of those two alternatives, with an additional option of vacation without breakfast in A (for the same price). The results revealed a surprising stronger bias toward vacation+breakfast in A within the second group, as opposed to the bias being equally distributed between the two choices in the first group.

We tried to repeat this result by generating obfuscated results which are easily comparable to (and less appealing than) their original counterparts. An example can be seen in Figure 4.

As before, we computed CMH over 50 histograms corresponding to 50 queries, with two results $r_1, r_2 \in High$ chosen for each query. One row of the histogram was taken from pairwise-preference and the other from triplet-preference template tasks, with r_3 taken as $Obs(r_1)$.[11] The resulting p-value was 0.002. We thus reject the null hypothesis, which stipulates that r_3 does not affect the preference distribution between r_1 and r_2. A close look at the results revealed a counterintuitive phenomenon: Adding $r_3 = Obs(r_1)$, instead of creating a preference bias *in favor of* r_1 (as Ariely's result would suggest), led raters to shy away from r_1. We conjecture that this is due to an impression of redundancy or maybe even spamminess for r_1, causing r_2 to appear as a cleaner alternative. This phenomenon will be the subject of future research.

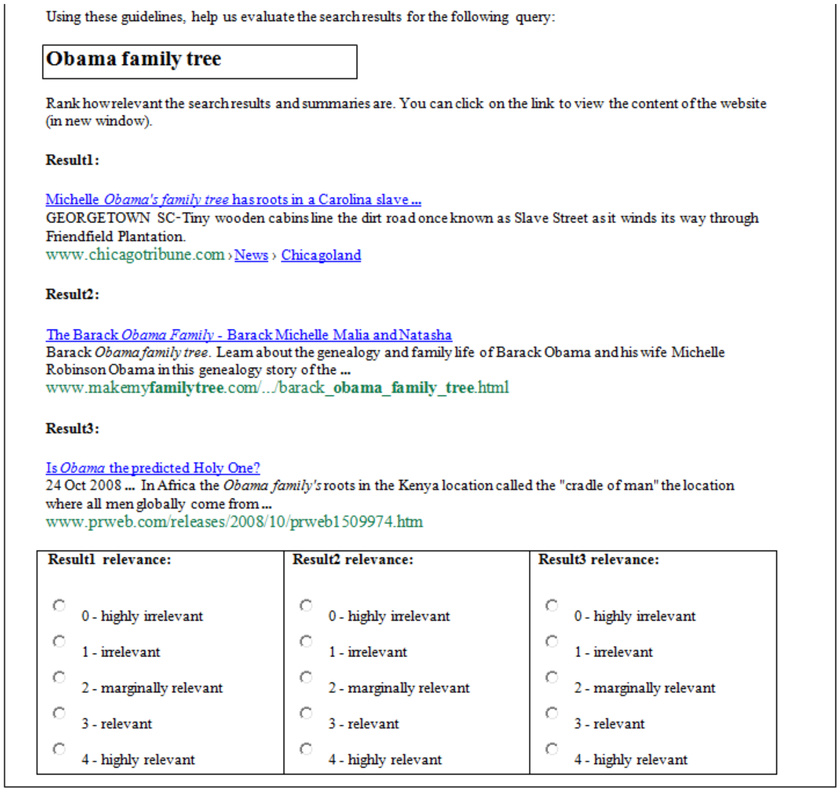
We also ran the same experiment, this time using pairwise-relevance with triplet-relevance for the histograms. Employing tie-ignoring resulted in p value of 0.9. Employing tie-splitting for the same test resulted in 0.4. From this we conclude that relevance responses, within sufficient context, are more stable than preference responses when we care about induced preference *only*.

4. HOW MANY PAIRS, AND WHICH ONES?

In this section we dispel the fear of quadratic cost that is often noted in papers on ranking, e.g., from [11]: *“Preferences have some disadvantages, most notably the lack of defined evaluation measures for preference judgments and the polynomial increase in the number of preferences needed in a test collection”*. The following quote from from [17] is another example: *“..the..approach is time consuming since it requires increasing the sample size from n to $O(n^2)$”*.

We show here that this problem can be dealt with using sampling tools from statistical learning theory. Consider a set of items V of cardinality n. Think of V as candidate search results for

[11] As usual, we routed 10 tasks per query for pairwise, 5 for each order, and 30 for the triplet, 5 for each possible order.



Using these guidelines, help us evaluate the search results for the following query:

Obama family tree

Rank how relevant the search results and summaries are. You can click on the link to view the content of the website (in new window).

Result1:

Michelle *Obama's family tree* has roots in a Carolina slave ...
GEORGETOWN SC-Tiny wooden cabins line the dirt road once known as Slave Street as it winds its way through Friendfield Plantation.
www.chicagotribune.com › News › Chicagoland

Result2:

The Barack *Obama Family* - Barack Michelle Malia and Natasha
Barack *Obama family tree*. Learn about the genealogy and family life of Barack Obama and his wife Michelle Robinson Obama in this genealogy story of the ...
www.makemy**familytree**.com/.../barack_**obama_family_tree**.html

Result3:

Is *Obama* the predicted Holy One?
24 Oct 2008 ... In Africa the *Obama family's* roots in the Kenya location called the "cradle of man" the location where all men globally come from ...
www.prweb.com/releases/2008/10/prweb1509974.htm

Result1 relevance:	**Result2 relevance:**	**Result3 relevance:**
○ 0 - highly irrelevant	○ 0 - highly irrelevant	○ 0 - highly irrelevant
○ 1 - irrelevant	○ 1 - irrelevant	○ 1 - irrelevant
○ 2 - marginally relevant	○ 2 - marginally relevant	○ 2 - marginally relevant
○ 3 - relevant	○ 3 - relevant	○ 3 - relevant
○ 4 - highly relevant	○ 4 - highly relevant	○ 4 - highly relevant

Figure 3: Results for the query "Obama family tree", with $r_1, r_2 \in High, r_3 \in Low$. The user is requested to submit three relevance judgements, but we ignore the response for r_3 because it is used as decoy here.

a query. Additionally, V is equipped with an *unknown* matrix $\{w(u,v)\}_{u,v\in V}$, where $w(u,v)$ measures the extent to which u should be preferred over v. Assume here that $w(u,v)+w(v,u) = 1$ for all $u,v \in V$. We wish to output a permutation π minimizing the following loss:

$$L(V,\pi) = \binom{|V|}{2}^{-1} \sum_{\substack{u <_\pi v \\ u,v\in V}} w(v,u) \ .$$

The notation $u <_\pi v$ means that u is preferred over v in π. The normalization $\binom{|V|}{2}$ is chosen for convenience but is immaterial. Other losses of a quadratic nature can be considered as well, but we concentrate on this standard loss to demonstrate our analysis. We are allowed to query $w(u,v)$ for any u,v for a unit cost. This cost should be thought of as the price we pay for soliciting pairwise preference from a small rater population sample. Alternatively, $w(u,v)$ may be "bought" by applying an expensive machine learned model for determining our belief in predicting preference between u and v, using features of u and v (as in the setting of [2, 5]). A ranking algorithm, given V, is allowed to query w at any chosen location and output a permutation π on the elements of V. The queries may be adaptive and depend on answers to previous queries. The goal is to design a ranking algorithm ALG minimizing a cost which we denote by $L(V,w,\mathrm{ALG}(V))$. We define the query complexity of ALG as the total cost paid for probing the matrix w. We are *not* interested in computational complexity here. We are only interested in the informational aspects of the problem.

We introduce the following useful notation. For any $u,v \in V$, and any permutation π of V, abuse notation by defining $\pi(u,v) = 1$ if $u <_\pi v$ and 0 otherwise (we view π as a $\{0,1\}$ vector of $n(n-1)$ coordinates). For any pair u,v, let

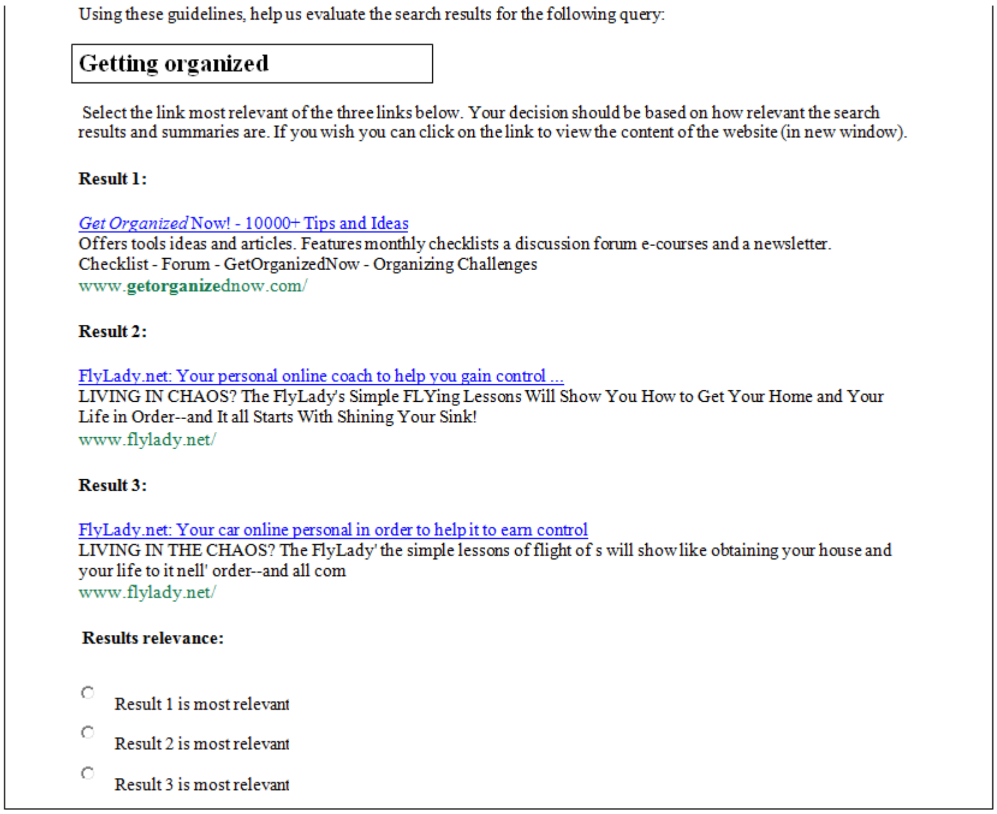
$$L_{u,v}(V,w,\pi) = \pi(u,v)w(v,u) + \pi(v,u)w(u,v) \ .$$

The loss function $L(V,w,\pi)$ can now be rewritten as

$$L(V,\pi) = \binom{|V|}{2}^{-1} \sum_{\substack{u<v \\ u,v\in V}} L_{u,v}(V,w,\pi) \ .$$

(By $u < v$ we mean that, formally, the pair u,v is unordered in the sum). This notation suggests that the pairs $u,v \in V$ are drawn from a uniform distribution over all pairs. The cost $L(V,\pi)$ now looks like a standard binary classification cost. Note that the concept space of permutations on V, denoted $S(V)$ (viewed as a subset of $\{0,1\}^{n(n-1)}$), is highly structured. In fact, it contains only $n!$ vectors. Summoning statistical learning theory, we are tempted to declare the VC dimension of the space as $\log(n!) = O(n \log n)$. (Recall that the VC dimension of a set of binary vectors is the largest integer d_{VC} such that there exists a set of d_{VC} indices realizing all possible $2^{d_{VC}}$ assignments. Such a set of indices is said to be *shattered*.) A careful analysis shows that the true VC dimension is actually linear.

PROPOSITION 4.1. *Let V be a set of elements of size n. View-*



Using these guidelines, help us evaluate the search results for the following query:

Getting organized

Select the link most relevant of the three links below. Your decision should be based on how relevant the search results and summaries are. If you wish you can click on the link to view the content of the website (in new window).

Result 1:

Get Organized Now! - 10000+ Tips and Ideas
Offers tools ideas and articles. Features monthly checklists a discussion forum e-courses and a newsletter.
Checklist - Forum - GetOrganizedNow - Organizing Challenges
www.**getorganized**now.com/

Result 2:

FlyLady.net: Your personal online coach to help you gain control ...
LIVING IN CHAOS? The FlyLady's Simple FLYing Lessons Will Show You How to Get Your Home and Your Life in Order--and It all Starts With Shining Your Sink!
www.flylady.net/

Result 3:

FlyLady.net: Your car online personal in order to help it to earn control
LIVING IN THE CHAOS? The FlyLady' the simple lessons of flight of s will show like obtaining your house and your life to it nell' order--and all com
www.flylady.net/

Results relevance:

○ Result 1 is most relevant
○ Result 2 is most relevant
○ Result 3 is most relevant

Figure 4: An obfuscated version of the second result is presented third. The user is requested to submit preference judgment.

ing $S(V)$ as an $n(n-1)$ dimensional space over $\{0,1\}$, the VC dimension of $S(V)$ is n.

PROOF. To see why the VC dimension cannot exceed $n-1$, let S' be a set of n ordered distinct pairs of elements. We will show that S' cannot be shattered. Assume otherwise. We observe that, in that case, S' cannot contain both (u,v) and (v,u), because for any $\pi \in S_n$, $\pi(u,v) = 1 - \pi(v,u)$, hence not all possible combinations can be assigned to the pair $\pi(u,v)$ and $\pi(v,u)$. Consider therefore the undirected graph $G = (V,E)$ with $(u,v) \in E$ if and only if $(u,v) \in S'$ or $(v,u) \in S'$. By the last observation, $|E| = n$. Hence it contains an undirected cycle which clearly cannot be shattered. Indeed, an undirected cycle cannot by directed cyclically by a permutation. To show that the VC dimension is $n-1$, it is enough to consider any directed spanning tree on G. Indeed, no matter how we direct the edges, the graph can be topologically sorted. □

Now let $\binom{V}{2}$ denote the set of all unordered pairs of elements in V. For a multi-set $A \subseteq \binom{V}{2}$, define the corresponding subsampled loss function $L_A(V,\pi)$ by

$$L_A(V,\pi) = |A|^{-1} \sum_{(u,v)\in A} L_{u,v}(V,w,\pi) \,.$$

(We are allowing A to be a multi-set so that we can obtain it by sampling with repetitions.) Using standard sampling tools from statistical learning theory, and using our above VC dimension bound in Proposition 4.1, we derive the following proposition:

PROPOSITION 4.2. *Assume A is a sample of m pairs chosen uniformly with repetitions from $\binom{V}{2}$. For any $\delta > 0$ with probability $1-\delta$,*

$$\sup_{\pi\in S(V)} \{|L_A(V,w,\pi) - L(V,w,\pi)|\} = O\left(\sqrt{\frac{n}{m}} + \sqrt{\frac{\log 1/\delta}{m}}\right) .$$

From this we conclude that for constant δ it suffices to take $m = O(n/\varepsilon^2)$ samples (with repetitions) for A in order to obtain (with constant probability of success) a permutation $\pi \in S(V)$ with cost $L(V,w,\pi)$ which is at most ε worse (additively) than that of the optimal π^*. This is done by algorithm ALG_ε, which samples A as above and optimizes $L_A(V,\cdot)$ as a surrogate for $L(V,\cdot)$. We say that the regret of ALG_ε is ε.

Is this the best we can do? First, let us see whether a regret of $O(\varepsilon)$ is at all useful. Consider the case where the cost $L(V,w,\pi^*)$ of the optimal solution for V is $O(\varepsilon)$ (otherwise the regret is same order of magnitude as the cost of the optimal solution). It is easy to show using a triangle inequality argument that in this case, $d_K(\pi,\pi^*)$ is also $O(\varepsilon)$, where d_K is the Kendall-Tau distance between permutations defined as:

$$d_K(\pi,\pi^*) = \binom{n}{2}^{-1} \sum_{u<_\pi v} \pi^*(v,u) \,.$$

But this means that, on average, elements $u \in V$ are swapped in π with an average of $O(\varepsilon n)$ elements with respect to π^*. More precisely, if $\mathrm{swap}(u)$ denotes the set of elements v for which u and v are ordered discordantly in π and in π^*, then the average size of

```
Algorithm $ALG_\varepsilon^{rec}(V)$
$n \leftarrow |V|$
if $n = O(1)$
    then return optimal solution for $V$ by exhaustive search
    else $\pi \leftarrow \mathrm{ALG}_\varepsilon(V)$
        $k \leftarrow$uniformly random chosen integer in $[n/3, 2n/3]$
        $V_L \leftarrow \{v \in V : v$ among top-$k$ in $\pi\}$
        $V_R \leftarrow \{v \in V : v$ among bottom-$(n-k)$ in $\pi\}$
        $\pi_L \leftarrow \mathrm{ALG}_\varepsilon^{rec}(V_L)$
        $\pi_R \leftarrow \mathrm{ALG}_\varepsilon^{rec}(V_R)$
        return concatenation of $\pi_L$ and $\pi_R$
```

Figure 5: Ranking algorithm $\mathrm{ALG}_\varepsilon^{rec}$

$\mathrm{swap}(u)$ over all $u \in V$ is $O(\varepsilon n)$. We can now use the well-known inequality by Diaconis and Graham [19] relating the Kendall-Tau distance to the Spearman-Footrule distance between permutations. The result tells us that elements in V are located in π an average distance of approximately $O(\varepsilon n)$ locations from their position in π^*. As a corollary, we get that in order to output a permutation placing the elements at distance at most n/C positions on average with respect to their position in π^*, we need to sample $O(nC^2)$ pairs.

4.1 Improving The Sample Complexity With Active Learning

We would like to improve the above sampling scheme in order to achieve a solution in which each element is dislocated (with respect to the minimizer π^* of $L(V, w, \cdot)$) by only a small (say, constant) distance. From the above discussion, a sub-quadratic sample will not be sufficient.

Intuitively, we should still be able to do better. After running ALG_ε for some small ε we should be fairly sure about pairs that are further away from each other with respect to the returned solution. We should therefore now sample pairs that are closer to each other. To demonstrate our approach, consider the recursive algorithm $\mathrm{ALG}_\varepsilon^{rec}$ (Figure 5). The algorithm applies ALG_ε recursively, randomly branching in a manner similar to that of QuickSort. In the recursion, the algorithm concentrates on pairs that are closer to each other.

Clearly, the query complexity of $\mathrm{ALG}_\varepsilon^{rec}$ is $O(n \operatorname{polylog} n)$, for fixed ε. We conjecture that $\mathrm{ALG}_\varepsilon^{rec}$ is a ranking algorithm with a regret that is better than that of $\mathrm{ALG}_{\varepsilon'}$ given parameters $\varepsilon, \varepsilon'$ that entail a similar query complexity for both algorithms. We were unable to prove this conjecture in this work. We leave this conjecture together with a related experimental study to future work. We draw our conjecture from a recent seminal result by Schudy and Kenyon [24], who design and analyze an algorithm for minimizing L to within a multiplicative factor of $1 + \delta$ for arbitrarily small δ (a PTAS) using complete knowledge of the entire preference matrix $w(u, v)$.

4.2 Ranking Close to the Top

Assume we are interested, as often is the case in IR ranking literature (e.g., [31]), in a ranking algorithm that ranks well at the top of the list. In other words, we decompose the problem into two parts: (1) Identify the top elements (2) Rank them. If t denotes the number of top elements we care about, we ignore preferences among the bottom $n - t$ elements. We could run $\mathrm{ALG}_\varepsilon^{rec}$ while abandoning parts of the recursion tree containing only elements that will clearly end up in the bottom $n - t$ places (see also analysis in [2] for a similar idea).

5. CONCLUSIONS AND DISCUSSIONS

In behavioral decision theory, a prevalent concept is that preferences are constructed by the user within the task and the context of the decision task [30, 33]. The constructive-preference approach argues that people often base their preferences in a given situation on information available at the time of the preference decision is made. This concept has been demonstrated in a variety of experiments.

We argue that relevance measuring for IR follows similar context effects. Relevance, as used by many metrics, is often assumed to be independent of context. That is, the perceived relevance of search result A should not depend on the perceived relevance of search result B. We provided empirical evidence for the contrary. In our experiments, we showed that the ordering of two results as induced from independent relevance responses statistically differs from that obtained from soliciting responses where the results provide mutual context In addition, we have demonstrated that when moving from pairs to triplets, the marginal preference information on pairs of results (contained in the triplet) does not change much. This means that pairwise comparisons are relatively stable within higher order tuples.

We also experimented with several types of contexts. From our results, one might deduce that asking users for relevance for similar results (as in the obfuscation experiment in section 3.7) might give more accurate results than asking for preference. On the other hand, asking users for preference on very quality-diverse results (section 3.6) might give more accurate results than relevance.

The classic evaluation metrics cannot be applied to rankings when the ground truth is built from pairwise comparisons. Indeed, these metrics are defined for independent graded relevance information. In attempting to provide alternative evaluation metrics, many researchers have tried to avoid the quadratic nature of the problem. We have shown that the arguably simplest such function is amenable to analysis using sampling ideas borrowed from statistical learning theory. Using a basic result, we conjecture that the query complexity (number of pairwise preferences we pay for) required for almost perfect optimization of the function for any practical purpose is $O(n \operatorname{polylog} n)$. This analysis could also be done on a weighted version of our function assigning more weight to pairs that are closer to the top of the output ranking, in the spirit of wpref [7, 11]. We leave this to future work.

There are many other directions for further research. We did not discuss topic sensitivity: How should ranking be performed in systems requiring result diversification when numerous incomparable topics are relevant? This aspect is of increasing importance in IR. We also did not investigate the effect of different query types (transactional, navigational, etc.) on our results. Similar empirical analysis should be done in each category.

6. REFERENCES

[1] N. Ailon. A simple linear ranking algorithm using query dependent intercept variables. In *ECIR '09: Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, pages 685–690, Berlin, Heidelberg, 2009. Springer-Verlag.

[2] N. Ailon and M. Mohri. An efficient reduction of ranking to classification. *Journal of Machine Learning Research (In Press)*, 2011.

[3] D. Ariely. *Predictably Irrational: The Hidden Forces That Shape Our Decisions*. HarperCollins, February 2008.

[4] D. Ariely, G. Loewenstein, and D. Prelec. Coherent arbitrariness: Stable demand curves without stable preferences. *Quarterly Journal of Economics*, 118:73–105, 2003.

[5] M. F. Balcan, N. Bansal, A. Beygelzimer, D. Coppersmith, J. Langford, and G. B. Sorkin. Robust reductions from ranking to classification. *Mach. Learn.*, 72(1-2):139–153, 2008.

[6] M. Braverman and E. Mossel. Noisy sorting without resampling. pages 268–276, 2008.

[7] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32, New York, NY, USA, 2004. ACM.

[8] Z. Cao, T. Qi, T. Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: From pairwise approach to listwise approach. In *Microsoft Tech Report MSR-TR-2007-40*, 2007.

[9] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1998.

[10] B. Carterette and P. N. Bennett. Evaluation measures for preference judgments. In *In Proceedings of SIGIR*, 2008.

[11] B. Carterette, P. N. Bennett, D. M. Chickering, and S. T. Here or there: Preference judgments for relevance. In *In Proceedings of the European Conference on Information Retrieval (ECIR)*, 2008.

[12] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 621–630, New York, NY, USA, 2009. ACM.

[13] C. L. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2009 web track. Technical report, no date.

[14] W. Cochran. Some methods for strengthening the common chi-square tests. *Biometrics*, 10:417–451, 1954.

[15] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. In *NIPS: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pages 451–457, Cambridge, MA, USA, 1998. MIT Press.

[16] V. Conitzer, A. Davenport, and J. Kalagnanam. Improved bounds for computing kemeny rankings. In *AAAI'06: Proceedings of the 21st national conference on Artificial intelligence*, pages 620–626. AAAI Press, 2006.

[17] K. Crammer and Y. Singer. Pranking with ranking. In *Advances in Neural Information Processing Systems 14*, pages 641–647. MIT Press, 2001.

[18] A. Das Sarma, A. Das Sarma, S. Gollapudi, and R. Panigrahy. Ranking mechanisms in twitter-like forums. In *WSDM: Proceedings of the third ACM international conference on Web search and data mining*, pages 21–30, 2010.

[19] P. Diaconis and R. Graham. Spearman's footrule as a measure of disarray. *J. of Royal Statistical Society*, 39(2):262–268, 1977.

[20] R. Herbrich, T. Graepel, and K. Obermayer. *Advances in Large Margin Classifiers*. MIT Press, 200.

[21] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artif. Intell.*, 172(16-17):1897–1916, 2008.

[22] K. Jarvelin and J. Kekalainen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20:422–446, 2002.

[23] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA, 2002. ACM.

[24] C. Kenyon Mathieu and W. Schudy. How to rank with few errors. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 95–103, New York, NY, USA, 2007. ACM.

[25] A. Kittur, H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proc. CHI 2008, ACM Pres*, pages 453–456, 2008.

[26] Y. Lan, T. Y. Liu, Z. Ma, and H. Li. Generalization analysis of listwise learning-to-rank algorithms. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 577–584, New York, NY, USA, 2009. ACM.

[27] S. Mizzaro. A new measure of retrieval effectiveness (or: What's wrong with precision and recall. In *In Proceedings of the International Workshop on Information Retrieval (IR'2001*, pages 43–52, 2001.

[28] A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Trans. Inf. Syst.*, 27:1–27, 2008.

[29] A. Oulasvirta, J. P. Hukkinen, and B. Schwartz. When more is less: the paradox of choice in search engine use. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 516–523, 2009.

[30] J. Payne, J. Bettman, and E. Johnson. *The adaptive decision maker*. Massachusetts: Cambridge University Press, 1993.

[31] C. Rudin. The P-Norm Push: A simple convex ranking algorithm that concentrates at the top of the list. *Journal of Machine Learning Research*, 10:2233–2271, Oct 2009.

[32] T. Saracevic. 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*, 58:2126–2144, 2007.

[33] P. Slovic, D. Griffin, and A. Tversky. Compatibility effects in judgment and choice. *Insights in decision making*, pages 5–27, 1990.

[34] K. Train. *Discrete Choice Methods with Simulation*. Massachusetts: Cambridge University Press, 2003.

[35] F. Xia, T. Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1192–1199, New York, NY, USA, 2008. ACM.

[36] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 391–398, 2007.