Offline Retrieval Evaluation Without Evaluation Metrics

Fernando Diaz
Canadian CIFAR AI Chair
Google
Montréal, QC, Canada
diazf@acm.org

Andres Ferraro
Mila - Quebec Artificial Intelligence Institute
Montréal, QC, Canada
andresferraro@acm.org

ABSTRACT

Offline evaluation of information retrieval and recommendation has traditionally focused on distilling the quality of a ranking into a scalar metric such as average precision or normalized discounted cumulative gain. We can use this metric to compare the performance of multiple systems for the same request. Although evaluation metrics provide a convenient summary of system performance, they also collapse subtle differences across users into a single number and can carry assumptions about user behavior and utility not supported across retrieval scenarios. We propose recall-paired preference (RPP), a metric-free evaluation method based on directly computing a preference between ranked lists. RPP simulates multiple user subpopulations per query and compares systems across these pseudo-populations. Our results across multiple search and recommendation tasks demonstrate that RPP substantially improves discriminative power while correlating well with existing metrics and being equally robust to incomplete data.

CCS CONCEPTS

• **Information systems** → **Retrieval effectiveness**.

KEYWORDS

information retrieval; recommender systems; offline evaluation

ACM Reference Format:
Fernando Diaz and Andres Ferraro. 2022. Offline Retrieval Evaluation Without Evaluation Metrics. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3477495.3532033

1 INTRODUCTION

A fundamental step in the offline evaluation of search and recommendation systems is to determine whether a ranking from one system tends to be better than the ranking of a second system. This often involves, given item-level relevance judgments, distilling each ranking into a scalar evaluation metric μ, such as average precision (AP) or normalized discounted cumulative gain (NDCG). We can then say that one system is preferred to another if its metric values tend to be higher. We present a stylized version of this approach in Figure 1a.

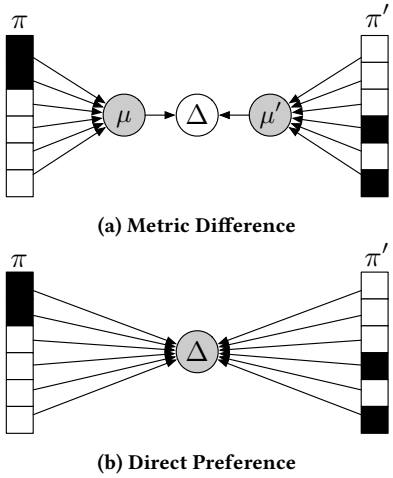


Figure 1: Metric Difference versus Direct Preference. System rankings π and π' are represented as boxes with shaded boxes indicating relevant item positions. A traditional evaluation metric μ such as average precision projects two system rankings to scalar values; the scalar metric difference indicates preference. Direct preference compares ranked lists explicitly, bypassing metric computation. Shaded nodes contrast the focus of research work between metrics and preferences.

Deriving a system preference from a metric difference can be problematic for two reasons. First, evaluation metrics, because they project a ranking onto a scalar value, can lose information about how two rankings differ. Take, as an example, the popular reciprocal rank metric (RR). Because RR only considers the rank position of the first relevant document, its value can be equal for two rankings that share the position of the first relevant document but differ dramatically at lower ranks. Although most salient for RR, metrics with smooth discount functions such as AP and NDCG still can collapse different rankings into the same or very similar scalar value by virtue of their sharp discounts. We refer to this as the problem of low *label efficiency*. Second, although most evaluation metrics are meant to model the quality of a ranking for system users, they can suggest similarity between systems that actually behave quite differently for different user populations. For example, RR might be an appropriate model for known-item search, but it does not capture higher-recall behaviors like electronic discovery and systematic review [29]. While metrics with smooth discounts can be interpreted as averaging performance across different possible user behaviors, they make very strong assumptions about the

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07.
https://doi.org/10.1145/3477495.3532033

distribution of behaviors. We refer to this as the problem of low *robustness to user behavior*.

We propose rank-paired preference (RPP), an evaluation method that addresses concerns both about label efficiency and robustness to user behavior. For a fixed request, RPP directly computes a preference between systems by modeling how different user subpopulations might prefer one algorithm over another. When aggregating these subpopulation preferences, each is weighted equally, rather than weighting users with lower recall requirements more heavily. We contrast RPP with metric-based evaluation in Figure 1b. By considering the contribution of lower-ranked relevant items, RPP more efficiently exploits available labels, resulting in higher sensitivity and discriminative power between systems compared to metric-based approaches.

We analyze RPP across a variety of search and recommendation tasks. Specifically, we show that (i) RPP is correlated with existing ranking metrics, (ii) RPP is equally robust to incomplete evaluation data compared to existing ranking metrics, and (iii) RPP has much higher discriminative power than existing ranking metrics. In particular, RPP's higher discriminative power suggests that preference-based evaluation should be further explored for offline evaluation.

2 MOTIVATION

In order to motivate our work, consider a retrieval scenario with binary relevance. Most ranked list evaluation metrics can be decomposed into a linear function of rank positions of the relevant items. Given a system ranking π, let f_i be the position of the ith relevant item. We can define many metrics as,

$$\mu(\pi) = \sum_{i}^{m} \delta(f_i)$$

where m is the number of relevant items and δ is a rank discount function (e.g., $\delta_{\text{DCG}}(i) = \frac{1}{\log(i+1)}$, $\delta_{\text{RBP}}(i) = \gamma^{i-1}$). In offline evaluation, we are interested in comparing this value to that of a second ranking π', using the difference in metric values to define preference. We can expand this difference into a sum of differences over the m positions of the relevant items,

$$\begin{aligned}\Delta\mu(\pi, \pi') &= \sum_{i}^{m} \delta(f_i) - \sum_{i}^{m} \delta(f_i') \\ &= \sum_{i}^{m} \delta(f_i) - \delta(f_i') \\ &= \sum_{i}^{m} \Delta\mu_i(\pi, \pi')\end{aligned}$$

This disaggregation by recall level lets us observe how the ith relevant item contributes to a change in $\Delta\mu$.

In Figure 2, we examine the behavior of $\Delta\mu_i$ under different evaluation metrics. In the left column, we show the relationship between $\Delta\mu_i$ and the position of the ith relevant item in the pair of ranked lists being compared (i.e. f_i and f_i'); in other words, this is the analytic relationship between f_i, f_i', and $\Delta\mu_i$ for different evaluation metrics. The first row considers rank-biased precision (RBP) with $\gamma = 0.5$ [20]; the second row, NDCG [14]; the last row, $\Delta\mu_i = \text{sgn}(\delta(f_i) - \delta(f_i'))$, reflecting the preference between rankings by

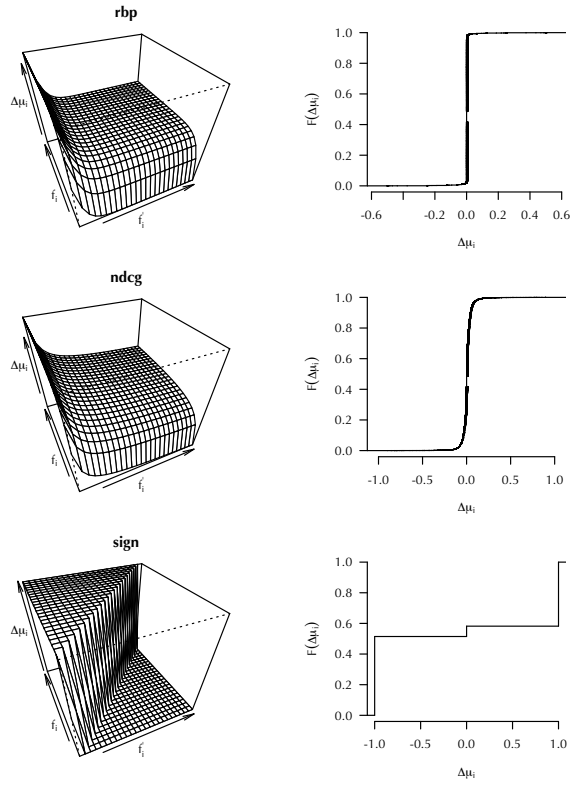


Figure 2: Surface of $\Delta\mu_i$ for comparing differences in the top 25 rank positions (left). Empirical cumulative distribution function for differences observed in all runs submitted to the TREC 2019 Deep Learning document ranking task (right).

a user seeking to find exactly i relevant items with as little effort as possible. Looking at the surface of $\Delta\mu_i$ for the RBP and NDCG metrics, we observe that, unless $\min(f_i, f_i')$ is small, the value of $\Delta\mu_i$ will be very small; as a result, the summation in $\Delta\mu$ will be dominated by changes in rank position amongst documents at the highest rank positions. This means that the relative preference of systems by users interested in higher recall will be overshadowed by the preferences of users interested in fewer relevant items.

In the right column of Figure 2, we show the associated empirical cumulative distribution function of $\Delta\mu_i$ for all runs submitted to the TREC 2019 Deep Learning document ranking task [11]. We can see that the distrbution of $\Delta\mu_i$ for RBP and NDCG is dominated by values close to zero. Looking at the sign of these differences in the last row, we observe that, for roughly 93% of the samples, $f_i \neq f_i'$.

This analysis provides evidence that the discounting of lower-ranked relevant items massively diminishes their contribution to metric differences, resulting lower label efficiency and robustness to user behavior. In the remainder of this paper, we address this by presenting an alternative evaluation method that more equally incorporates preferences of users with different recall requirements.

3 PRELIMINARIES

We are interested in comparing pairs of rankings for the same request from two different systems. Our methods apply to both search and recommendation contexts. As such, we will refer to queries and user profiles as *requests* and documents as *items*.

Given a request q and a corpus of n items, let $\mathbf{y} \in \mathfrak{R}^n$ be a vector of item relevance grades, where $\mathbf{y}_i$ is the relevance grade of item i. We refer to $\mathbf{y}$ as the relevance judgments for q and any grade greater than 0 as relevant. Given a request q, let $\pi \in S_n$ be a system's ranking of items in the corpus. In cases where a system returns a ranking only of the top k items, we assume a worst case ordering of the remaining items. For example, if a top k ranking X omits five items with grades $\{1, 1, 2, 4, 5\}$ from the retrieval, then we treat these items as ranked in *increasing* order of utility at the end of the total corpus ranking,

$$X = \underbrace{04300010300\ldots000}_{\text{top } k}\ \underbrace{0000\ldots0011245}_{\text{bottom } n-k}$$

Similarly, if a top k ranking Y omits three items with grades $\{1, 2, 4\}$ from the retrieval,

$$Y = \underbrace{30453001100\ldots000}_{\text{top } k}\ \underbrace{0000\ldots0000124}_{\text{bottom } n-k}$$

This worst case assumption provides a conservative lower bound on system performance.

When provided with two rankings π and π' for the same request, we are interested in determining if we prefer π to π', indicated as $\pi \succ \pi'$. *Metric-based evaluation* leverages a function $\mu : S_n \to \mathfrak{R}$ that computes the quality of a ranking defines a preference as,

$$\mu(\pi) > \mu(\pi') \to \pi \succ \pi'$$
$$\mu(\pi) < \mu(\pi') \to \pi \prec \pi'$$

Our work considers *preference-based evaluation*, an approach that directly defines a function $\Delta : S_n \times S_n \to \mathfrak{R}$ such that,

$$\Delta(\pi, \pi') > 0 \to \pi \succ \pi'$$
$$\Delta(\pi, \pi') < 0 \to \pi \prec \pi'$$

In this section, we will review and extend three concepts from the existing literature: position-based metrics, pseudo-populations, and preference-based evaluation. In Section 4, we will synthesize these concepts into our new evaluation method, recall-paired preference.

3.1 Position-Based Evaluation Metrics

Let the *search length* $f_i(\pi)$ be the position of the ith ranked relevant item in π.[1] In our example, $f_1(X) = 2$, $f_2(X) = 3$, $f_3(X) = 7$, and so forth. For clarity, we use f_i to refer to $f_i(\pi)$ and f_i' to refer to $f_i(\pi')$. Position-based evaluation metrics adopt the principle of minimal effort,

For a user interested in i relevant items,

$$f_i < f_i' \to \pi \succ \pi'$$

[1]This is slightly different from Cooper's definition of search length [10] which counts the number of non-relevant items above the ith relevant item, (i.e. $f_i(\pi) - i$).

Cooper [10] describes different types of users who may be interested in different levels of recall i. We refer to situations where a user is looking for one relevant item as *precision-oriented*. Historically, the reciprocal of the rank of the first relevant item (i.e. $\frac{1}{f_1}$) has been used for such tasks. We call f_i the *initial search length* (ISL) and it is related to Cooper's 'Type 1 search length.'

We refer to situations where a user is looking for all of the relevant items as *recall-oriented*. Assuming m relevant items, we refer to f_m, the position of the last relevant item, as the *total search length* (TSL). This is related to Cooper's 'Type 3 inclusive search length.'

In situations where we are uncertain if the user is precision- or recall-oriented, we can compute the expectation over all possible recall orientations. We can express the *average search length* (ASL) as,[2]

$$\text{ASL}(\pi) = \mathbb{E}_i\left[f_i\right] \tag{1}$$

Rocchio [23] refers to this as the *average rank* metric. More recently, ASL has been used in the context of unbiased learning to rank [16].

3.2 Pseudo-Populations

In the previous section, we described metrics that operate under the assumption that a user is interested in precisely i relevant items. In order to build on this work, we turn to how we can model different possible subpopulations of users who desire different numbers of relevant items.

Similar to Cooper, Robertson [21] suggested that a user may have a specific recall requirement when interacting with an information access system. In other words, for a request with m relevant items, a user may be in one of m recall conditions. We refer to $\mathcal{U}_i$ as the *pseudo-population* interested in i relevant items and $p(i) = p(u \in \mathcal{U}_i)$ is the probability of a user seeking exactly i relevant items. Robertson observed that, if $p(i) = \frac{1}{m}$, then AP can be interpreted as the expected precision for users from these pseudo-populations,

$$\text{AP}(\pi) = \mathbb{E}_i\left[\frac{i}{f_i}\right]$$

where $\frac{i}{f_i}$ is the precision for users with a recall requirement of i. Sakai and Robertson [25] extended this to arbitrary metrics, defining Normalized Cumulative Precision (NCP) as,

$$\mathbb{E}_i\left[\mu_i(\pi)\right] = \sum_{i=1}^{m} p(i)\mu_i(\pi)$$

where $\mu_i(\pi)$ is a *partial recall metric* based on the recall level i. Example partial recall metrics include $\mu_i(\pi) = \frac{i}{f_i}$ (i.e. precision at f_i, as used in AP) and $\mu_i(\pi) = f_i$ (as used in ASL).

The distribution $p(i)$ gives us a place where we can explicitly encode any information we have about user recall requirements. For example, when we do not know the distribution of recall requirements of users or if we want robust performance across a variety of possible recall requirements, a uniform distribution over all recall levels (i.e. $p(i) = \frac{1}{m}$) might be appropriate. In situations

[2]Two other metrics, bpref [4] and atomized search length [2], compute an expectation over relevant items, but, like Cooper, use the number of preceding nonrelevant items. By linearity of expectation, we observe that $\mathbb{E}_i\left[f_i - i\right] = \mathbb{E}_i\left[f_i\right] - \mathbb{E}_i\left[i\right]$, which Rocchio [23] refers as the *recall error*. For a fixed query, this value is rank-equivalent to $\mathbb{E}_i\left[f_i\right] = \text{ASL}(\pi)$.

where we have information about the true distribution of recall requirements or want to emphasize performance for a specific user behavior, we can adopt a non-uniform distribution for $p(i)$. For example, 'position bias' can be reflected in a definition of $p(i)$ that monotonically decreases with i. In contrast with most existing models, this distribution is over recall levels, as opposed to rank positions.

We can also consider pseudo-populations generated from other available information. For example, when we have multiple possible relevance grades, we can consider a pseudo-population of users satisfied by an item if its grade is above some threshold [22]. The utility for the pseudo-population of users interested in items with at least grade λ is,

$$\mathbb{E}_i\left[\mu_i(\pi)|\lambda\right] = \sum_{i=1}^{m_\lambda} p(i|\lambda)\mu_{i,\lambda}(\pi)$$

where m_λ is the number of items with a grade of at least λ, $p(i|\lambda)$ is the probability of a user in this population seeks exactly i relevant items, and $\mu_{i,\lambda}(\pi)$ is the binary partial recall metric assuming only items with grade greater than λ are relevant. We will use $\lambda = 1$ to refer to users who consider relevance as binary.

In situations where items are associated with attributes such as genres or per-request subtopics, we can define pseudo-populations based on these categories [1]. As a result, the utility for the pseudo-population of users interested in category t is,

$$\mathbb{E}_i\left[\mu_i(\pi)|t\right] = \sum_{i=1}^{m_t} p(i|t)\mu_{i,t}(\pi)$$

where $p(i|t)$ defines the probability that a user interested in subtopic t seeks i relevant items and $\mu_{i,t}(\pi)$ is the binary partial recall metric assuming only items from subtopic t are relevant.

3.3 Preference-Based Evaluation

Most existing approaches—including those in Sections 3.1 and 3.2—collapse the performance of a system into a single scalar number. An alternative to comparing metrics is to compare rankings directly.

Preference-based evaluation[3] assigns, for a pair of rankings, a preference between them. Traditionally, we elicit this preference from human judges in an interface that presents two rankings alongside each other [17, 27]. Sanderson et al. [26] demonstrated that this approach correlated well with metric-based evaluation across a variety of retrieval scenarios.

In the context of online experimentation, interleaving combines pairs of rankings and computes a preference between them based on user clicks [15]. Given a request from a user, two rankings π and π' are randomly interleaved so as to simulate a choice experiment for the user. The user then inspects the ranking, clicking on relevant items. We say that the ranking π is preferred to π' if it retrieves more clicked items at a rank cutoff k, a value based on the position of the last-clicked item. Because of randomness in both user behavior (e.g. their recall requirement) and the interleaving process itself, we can model k as a random variable. As a result, for a fixed query,

[3]We note that preference-based evaluation differs from evaluating with item preferences, which often still collapses rankings into a single scalar number [6, 9].

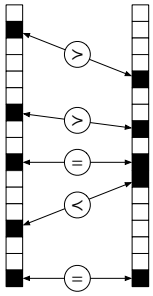


Figure 3: Recall-paired preference. A user interested in i relevant items will prefer the ranking that lets them satisfy their need with minimal effort.

we can define the interleaving preference as,

$$\mathrm{I}(\pi, \pi') = \mathbb{E}_k\left[\mathrm{sgn}(v_k(\pi) - v_k(\pi'))\right] \tag{2}$$

where $v_k(\pi)$ is a *partial precision metric* based on the rank position k; we contrast this with partial recall metrics, which are based on recall level. Example partial precision metrics include $v_k(\pi) = \frac{|\{f_i|f_i \le k\}|}{k}$ (i.e. 'precision at k', as used in interleaving). In online evaluation, we compute the interleaving metric by empirically estimating Equation 2 from sampled user requests and clicks on interleaved rankings. Chapelle et al. [8] demonstrated the sensitivity of interleaving experiments across a variety of online search scenarios.

4 RECALL-PAIRED PREFERENCE

We are now ready to combine the concepts from Section 3 into a new preference-based evaluation method.

We begin by describing conceptually how we compare two rankings. Consider our example rankings X and Y introduced in Section 3. First, we sample a user u based on our distribution over pseudo-populations. By appealing to the principle of minimal effort (Section 3.1), we can infer which ranking u prefers. For example, if we sample a user from $\mathcal{U}_1^{\lambda=1}$, then $f_1(X) = 2 > 1 = f_1(Y)$ and, since we prefer higher ranks, $Y \succ X$. If we sample a user from $\mathcal{U}_1^{\lambda=4}$, then, using $f_{1,\lambda=4}(\pi)$ to represent the position of the first ranked item with relevance grade at least 4, $f_{1,\lambda=4}(X) = 2 < 3 = f'_{1,\lambda=4}(X)$ and $X \succ Y$. We can repeatedly sample users, incrementing an accumulator by 1 if $X \succ Y$ and decrementing by 1 if $X \prec Y$. If the value of accumulator is positive, we say that $X \succ Y$; if it is negative, then $X \prec Y$. This is equivalent to computing the expected preference across the m paired positions of relevant items (Figure 3). Because we pair items according to equivalent recall levels, we refer to this metric as *recall-paired preference* (RPP).

More formally, for binary relevance and no subtopics, we define RPP between two rankings as the expected value of the preference,

$$\mathrm{RPP}(\pi, \pi') = \mathbb{E}_i\left[\mathrm{sgn}(f'_i - f_i)\right] \tag{3}$$

$$= \sum_{i=1}^{m} p(i) \times \mathrm{sgn}(f'_i - f_i) \tag{4}$$

where $p(i)$ is the probability of a user seeking exactly i relevant items. RPP takes a value in $[-1, 1]$ where positive values indicate stronger preference for π, negative values a preference for π', and zero indicating indifference. Moreover, RPP is a preference, so $\text{RPP}(\pi, \pi') = -\text{RPP}(\pi', \pi)$.

In practice, when we refer to RPP, we will use the graded version,

$$\text{RPP}(\pi, \pi') = \sum_{\lambda \in \Lambda} \sum_{i=1}^{m} p(i, \lambda) \times \text{sgn}(f'_{i,\lambda} - f_{i,\lambda}) \tag{5}$$

where Λ is the set of all possible grades for this request and $f_{i,\lambda}$ is the rank position of the ith relevant item with grade of at least λ. In the binary relevance case, this reduces to Equation 4.

The subtopic-aware version of RPP can be similarly defined,

$$\text{ST-RPP}(\pi, \pi') = \sum_{t \in \mathcal{T}} \sum_{i=1}^{m} p(i, t) \times \text{sgn}(f'_{i,t} - f_{i,t}) \tag{6}$$

where $\mathcal{T}$ is the set of all possible subtopics for this request and $f_{i,t}$ is the rank position of the ith relevant item with subtopic t.

4.1 Comparison to Existing Metrics

In this section, we compare RPP to the methods presented in Section 3.

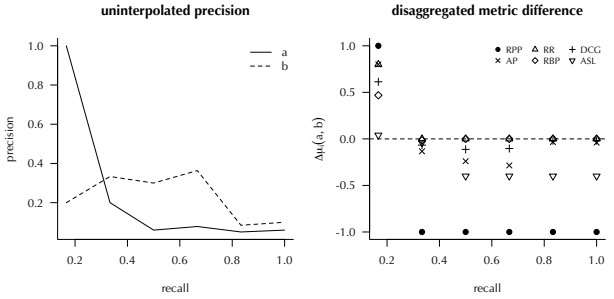
To begin, we can compare RPP with metric-based evaluation by analyzing how the disaggregated metric $\Delta\mu_i$ (Section 2) changes as a function of i. Figure 4a contains the disaggregated values for several standard evaluation metrics. These expressions encode the relative weight allocated to different pseudo-populations $\mathcal{U}_i$. Standard evaluation metrics such as RBP, AP, DCG, and RR all observe the largest contribution to metric differences at the highest rank positions. Even AP, which modulates the difference in inverse positions with a multiplicative recall level factor i is dominated by diminishing differences at low ranks. This confirms our claim of poor label efficiency (since relevant items at lower rank positions can contribute less) and poor robustness to user behavior (since the performance difference for users interested in higher recall levels is negligible). RR, ISL, and TSL also exhibit poor label efficiency and robustness to user behavior since, by design, they exclude all but a single difference. Meanwhile, ASL will tend to have the opposite effect of emphasizing differences at lower ranks since, at these higher recall levels, f_i and f'_i are likely to be separated by many more rank positions than earlier recall levels. In contrast, for RPP, the disaggregated magnitude is fixed across recall levels, resulting in both label efficiency (since all relevant items contribute equally regardless of rank position) and robustness to user behavior (since the performance differences for users at all recall levels contribute equally).

To illustrate the implication of these position biases, we can see how $\Delta\mu_i$ changes for two hypothetical rankings, a and b. In the left-hand plot of Figure 4b, we depict the uninterpolated precision-recall curves for a and b. Whereas AP approximates the area under the uninterpolated precision-recall curve, RPP can be interpreted as a sign test at sampled recall levels, similar to approaches taken for ROC curves [3].

In the right-hand plot of Figure 4b, we present $\Delta\mu_i(a, b)$ as a function of i. Notice that almost every traditional metric, including 'recall-oriented' metrics like AP, allocate most of the mass to the top position. Conversely, ASL allocates more weight to later values of i. RPP, on the other hand, treats all recall levels equally.

(a) Disaggregated recall-paired metrics

metric	$\Delta\mu_i$
RBP [20]	$\gamma^{f_i} - \gamma^{f'_i}$
AP	$i\left(\frac{1}{f_i} - \frac{1}{f'_i}\right)$
NDCG [14]	$\frac{1}{\log_2(f_i+1)} - \frac{1}{\log_2(f'_i+1)}$
RR	$\frac{1}{f_i} - \frac{1}{f'_i}, (i = 1)$
ISL [10]	$f'_i - f_i, (i = 1)$
TSL [10]	$f'_i - f_i, (i = m)$
ASL, bpref [2, 4, 23]	$f'_i - f_i$
RPP	$\text{sgn}(f'_i - f_i)$



(b) Metric differences for two systems' rankings of one request with six relevant items.

Figure 4: Disaggregated metric behavior.

Although position-based metrics like ASL more evenly allocate weight across recall levels, Magdy and Jones [19] note that the range of this metric can be quite large, sensitive to outliers, and difficult to reason with. Unretrieved items, often assumed to be ranked at the bottom of a ranking of the full corpus (Section 3), can exacerbate the variance in the metric, even for moderately sized corpora. In contrast, RPP, because it only considers relative positions, can be computed without an exact corpus size.

In comparison to interleaving, while Equations 2 and 4 appear similar, there are a few important differences. First, the event space for interleaving is the set of all rank positions rather than the set of all recall levels. This subtle difference means that interleaving emphasizes the number of relevant items collected at a rank position rather than the effort taken to collect the same number of relevant items. Moreover, in online interleaving, because $p(k)$ is derived from behavioral data, it will be skewed toward higher rank positions (i.e. users introduce position bias) and preferences at lower positions will be overshadowed by those in higher positions.

5 EXPERIMENTS

Our main thesis is that RPP, by more uniformly measuring performance across recall levels, more efficiently uses relevance labels in evaluation compared to existing retrieval metrics. As such, experiments are centered around three questions, (i) how does RPP correlate with existing metrics? (ii) how robust is RPP to incomplete

Table 1: Datasets used in experiments.

	requests	runs	rel/request	subtopics/request
core (2017)	50	75	180.04	0
core (2018)	50	72	78.96	0
deep-docs (2019)	43	38	153.42	0
deep-docs (2020)	45	64	39.27	0
deep-pass (2019)	43	37	95.40	0
deep-pass (2020)	54	59	66.78	0
web (2009)	50	48	129.98	4.98
web (2010)	48	32	187.63	4.17
web (2011)	50	62	167.56	3.36
web (2012)	50	48	187.36	3.90
web (2013)	50	61	182.42	3.18
web (2014)	50	30	212.58	3.12
robust	249	110	69.93	0
ml-1M	6005	21	18.87	0
libraryThing	7227	21	13.15	0
beerAdvocate	17564	21	13.66	0

data? (iii) how effective is RPP at discriminating between runs? In order to answer these questions, we use a variety of information access scenarios covering both search and recommendation tasks.

5.1 Data

We present details of the data used in our experiments in Table 1. We include runs submitted to multiple TREC tracks, including the Deep Learning Document Ranking (2019, 2020), Deep Learning Passage Ranking (2019, 2020), Common Core (2017, 2018), Web (2009-2014), and Robust (2004). We downloaded all data from NIST, including runs and relevance judgments. Web track data includes subtopic judgments.

Additionally, we used a variety of recommendation systems runs prepared by Valcarce et al. [28] for the MovieLens 1M, LibraryThing, and Beer Advocate datasets.[4] Consistent with their work, we converted graded judgments to binary labels by considering any rating below 4 as nonrelevant and otherwise relevant.

5.2 Methods

In order to measure the similarity between RPP and metric-based approaches, we measured the Kendall's τ correlation between system ordering by RPP with system ordering by baseline metrics (Section 5.5). We describe how to compute an ordering of systems from RPP in Section 5.4.

We evaluated the robustness to incomplete data under two conditions. Our first experiment tests how well a metric with fewer *judged requests* can order systems compared to the same metric with the complete set of judged requests. This simulates the scenario where we have a paucity of requests but, for those requests, we have ample labeled items. Our second experiment tests how well a metric with fewer *judgments per request* can order systems compared to the same metric with the complete set of judgments. This simulates the scenario where we have ample requests but sparse judgments for each request.

In order to evaluate the sensitivity of a metric, we adopt Sakai's method of computing discriminative power [24]. For a single data set (row in Table 1), we compute the RPP or metric differences for all pairs of runs over all requests. We then measure what fraction of system pairs achieve a p-value lower than 0.05 for each metric. In order to compute p-values, we use two methods: a Student's t-test with Bonferonni correction and Tukey's honestly significant difference (HSD) test. We adopt the randomized HSD as proposed by Carterette [7].

5.3 RPP Variants

For binary relevance with no subtopics, we consider several definitions of $p(i)$. In addition to $p(i) = \frac{1}{m}$, we include versions that consider non-uniform, top-heavy distributions of pseudo-populations,

$$p_{\text{DCG}}(i) \propto \frac{1}{\log_2(i+1)} \qquad p_{\text{inverse}}(i) \propto \frac{1}{i}$$

which reflect the rank importance for NDCG and RR. Note that these discounts are a function of recall level, rather than rank position.

When we adopt graded RPP for evaluation, we assume independence between recall requirements and grade, $p(i, \lambda) = p(i)p(\lambda)$, and define $p(\lambda)$ for $\lambda \in \Lambda$ as,

$$p(\lambda) \propto |\{i | \mathbf{y}_i \geq \lambda\}|$$

When conducting subtopic evaluation, we again assume independence between recall requirements and subtopic, $p(i, t) = p(i)p(t)$, with $p(t)$ defined as,

$$p(t) \propto |\{i | \mathbf{y}_i > 0 \wedge t \in \mathbf{s}_i\}|$$

where $\mathbf{s}_i \subseteq \mathcal{T}$ indicates the subtopics associated with item i. In addition to these $|\mathcal{T}|$ pseudo-populations, we consider a background interest t^* pseudo-population satisfied by *any* subtopic (i.e. standard relevance, $p(t^*) \propto |\{i | \mathbf{y}_i > 0\}|$).

5.4 Aggregating RPP

RPP gives us a preference between a pair of rankings for the same request but we are often interested in generating an ordering of more than two systems. Given a set of runs $\tilde{S}_n$ for the same request, we can compute the win rate for a ranking $\pi \in \tilde{S}_n$ as,

$$\text{RPP}_{\tilde{S}_n}(\pi) = \sum_{\pi' \in \tilde{S}_n} \text{RPP}(\pi, \pi') \tag{7}$$

We can then use a preference aggregation scheme to order systems for a set of queries $\boldsymbol{Q}$. In experiments where we need an ordering of systems for a set of requests, we adopt Markov chain aggregation, due to its effectiveness across a variety of domains [12]. Note that $\text{RPP}_{\tilde{S}_n}(\pi)$ reflects the relative position of π within $\tilde{S}_n$ and may vary across different sets of runs.

5.5 Baseline Metrics

As baseline metrics, we used AP, NDCG, and RR with no rank cutoff as implemented in NIST `trec_eval`.[5] We implemented ASL by moving unranked items to the end of the corpus, using a corpus size equal to the number of unique items in union of all rankings and the relevant items. For subtopic metrics, we use intent-aware mean average precision (MAP-IA) with no rank cutoff and intent-aware expected reciprocal rank (ERR-IA) and subtopic recall (strec), both

[4] https://github.com/dvalcarce/evalMetrics

[5] https://github.com/usnistgov/trec_eval

Table 2: Correlation with Existing Metrics. Kendall's τ between rankings of runs for pairs of metrics averaged across all datasets.

(a) Single Topic Metrics

	invRPP	RPP	dcgRPP	AP	NDCG
RR	0.61	0.45	0.49	0.49	0.50
invRPP	-	0.79	0.85	0.82	0.80
RPP	-	-	0.93	0.86	0.87
dcgRPP	-	-	-	0.88	0.88
AP	-	-	-	-	0.89

(b) Subtopic Metrics

	st-dcgRPP	MAP-IA	st-invRPP	ERR-IA	strec
st-RPP	0.88	0.73	0.70	0.26	0.30
st-dcgRPP	-	0.77	0.79	0.34	0.36
MAP-IA	-	-	0.74	0.39	0.36
st-invRPP	-	-	-	0.52	0.49
ERR-IA	-	-	-	-	0.66

with a rank cutoff of 20, as adopted for Web tracks and implemented in `ndeval`.[6]

In order to compare with interleaving, we developed *offline interleaving* (OI) based on a simulated user. Carterette [5] observed that many existing metric definitions implicitly include a model of user behavior. As such, given relevance information and a pair of system rankings to compare, we can simulate user interaction and compute an offline interleaving preference. To do so, given two rankings π and π', we can generate the two possible interleaved rankings $\tilde{\pi}$ and $\tilde{\pi}'$. Then, we can use a browsing model and relevance information to simulate an online interleaving experiment and estimate Equation 2.

6 RESULTS

6.1 Correlation with Existing Metrics

In order to get a sense of the relationship between baseline metric differences and RPP, we sampled pairs of runs for random queries in the Robust dataset. We computed baseline metrics for each ranking and then plotted the metric differences against the RPP for the same pair of runs for the same query (Figure 5). Although the sign agreement of RPP with AP, NDCG, and ASL is close to 0.90, it drops to 0.50 for RR. This result is consistent with the sign agreement between RR and AP (0.56), NDCG (0.58), and ASL (0.47).

These results can be explained by two properties of RR. First, because RR ignores recall levels higher than 1, metrics that measure higher recall levels will incorporate information that can reverse the order of systems. Second, the small number of unique RR values results in a number of ties between systems which are often resolved by metrics that consider more recall levels.

The sign *disagreement* between RPP and AP, NDCG, and ASL tends to occur for small differences in performance, with metrics largely agreeing for dramatic differences in performance. This indicates that, even when the top ranked relevant items largely agree in classic metrics, there is enough disagreement at higher recall levels to differ from RPP.

[6] https://github.com/trec-web/trec-web-2014

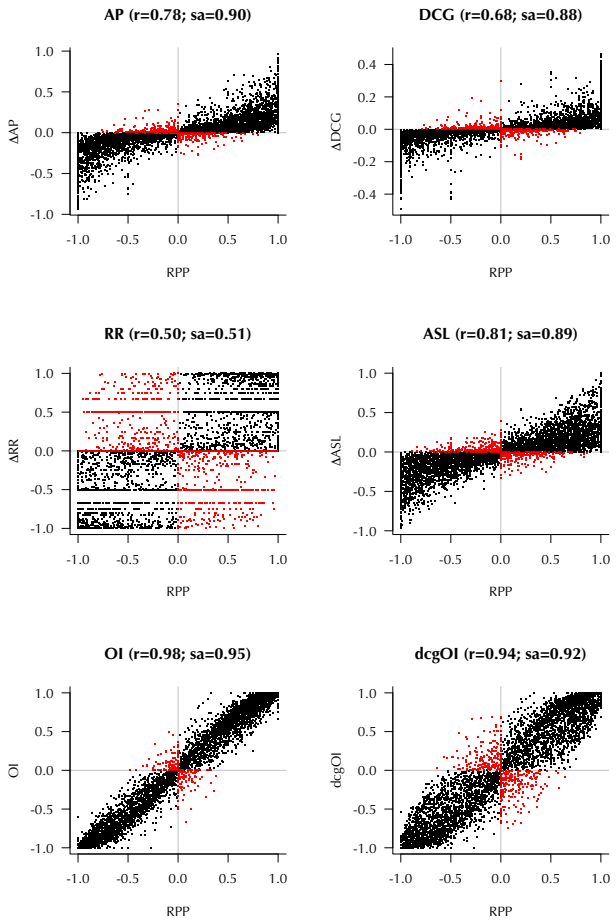


Figure 5: Query-level metrics differences for sampled runs from Robust. Points in red indicate a difference in run ordering. Titles include the Pearson correlation (r) and fraction of points where RPP and the metric difference agree in sign (sa).

We implemented offline interleaving with two different user models, uniform and DCG-based. We present the relationship between OI and RPP preference in the bottom row of Figure 5. We found that the sign agreement was higher between RPP and OI compared to RPP and baseline metrics. Moreover, the relationship between the preference magnitudes shows a strong linear correlated ($r = 0.98, p < 0.001$). Comparing to OI with a DCG-style user model, the agreement and correlation degrade ($r = 0.94, p < 0.001$) but are still higher than baseline metrics.

In addition to pairwise preference agreement, we were interested in the similarity between an ordering of runs induced from RPP preferences (Section 5.4) and an ordering induced from baseline metrics. To this end, we computed the Kendall's τ between the rankings of runs for each dataset. We present the average correlation across these datasets in Table 2a. We first notice that RPP with uniform position weighting (labeled RPP) correlates with AP

and NDCG at a level close to how those metrics correlate with each other. If we replace the uniform position weighting with a DCG-style non-uniform position weighting (dcgRPP), this correlation improves close to their correlation with each other. As suggested by Figure 5, the correlation between RR and RPP, AP, and NDCG is low. Although correlation between dcgRPP with RR (0.49) is higher than RPP with RR (0.45), it remains comparable to that of RR with AP and NDCG. Using the reciprocal rank for the position discount (invRPP) improves this correlation further (0.61), suggesting that our pseudo-population modeling works as expected. That said, we do not expect this metric to correlate perfectly with RR since invRPP considers items below the first relevant item.

We also include correlations for subtopic metrics in Table 2b. We observe similar patterns to single topic metrics. MAP-IA, a subtopic version of AP, correlates well with the subtopic versions of RPP and dcgRPP. Top-heavy metrics ERR-IA and strec, on the other hand, correlate well with the subtopic version of invRPP, consistent with earlier results.

Taken together, these results indicate that RPP metrics effectively capture a variety of aspects of baseline metrics but do not correlate perfectly, suggesting that they add information to evaluation. Moreover, they demonstrate the ability to adapt RPP to different scenarios (e.g. position bias, novelty).

6.2 Robustness to Incomplete Data

Because missing data is common in offline evaluation, we explored the behavior of RPP under two degradation schemes. Due to space constraints, we provide representative results for news search (Robust), web search (2020 Deep Learning Passage Ranking), and recommendation (MovieLens 1M).

We show the sensitivity of results when evaluating with fewer requests in Figure 6. For each metric, we calculate the correlation between system rankings with missing requests and system rankings with all requests; an insensitive metric will have higher correlation with fewer requests. Consistent with other work, across all datasets, RR correlation degrades the fastest, suggesting that removing a few requests can alter system ordering. In general, the RPP family of metrics degrades as gracefully as or better than existing metrics, AP and NDCG.

We show the sensitivity of results when evaluating with fewer labeled items in Figure 7. Here, for each metric, we calculate the correlation between system rankings with missing labels and system rankings with all labels. As with missing requests, RR degrades poorly across datasets, which is expected since the removal of the top ranked item is likely to substantially perturb performance (Section 6.1). AP also degrades poorly, especially when more than 50% of the judgments are missing. RPP variants degrade more gracefully and are comparable to NDCG, a metric considered less sensitive to missing label [28].

6.3 Discriminative Power

We present measurements of discriminative power in Tables 3 and 4. Although results are largely consistent for both the HSD test and t-test, we include both to further support our analysis.

One fundamental impact we should expect with poorer label efficiency is a reduced ability to distinguish pairs of systems. Across almost all datasets, we observe that RPP-style preferences have substantially more discriminative power compared to baseline metrics. Both AP and RR tend to have lower discriminative power than NDCG, consistent with previous results [28]. The low discriminative power of RR certainly arises from both the poor label efficiency and the large number of ties (Section 6.1). And, although non-uniform position-weighting (dcgRPP, invRPP) sometimes improves discriminative power slightly, uniform position-weighting (RPP) consistently has high discriminative power compared to baseline metrics.

The discriminative power of RPP is present in subtopic evaluations as well, at times dramatically so compared to existing subtopic metrics. We note that this may be, in part, due to the addition of a background pseudo-population reflecting binary relevance.

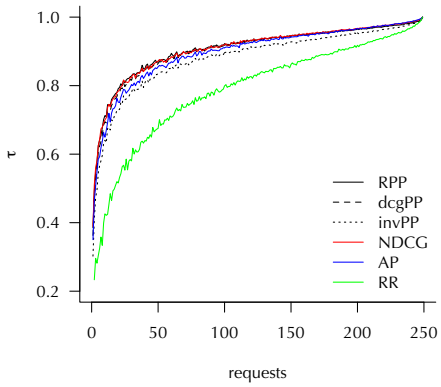
We observe that the number of detectable differences improves for all methods when more requests are present (e.g. ml-1M, libraryThing, beerAdvocate). This should be expected since, regardless of metric, more evaluation data will result in better performance estimates. That said, even in these regimes, RPP-style evaluation is more sensitive. Moreover, if conducting segment analysis (e.g. for fairness evaluation), under-represented groups, by definition, will have substantially less data.

Although the discriminative power of RPP is not alone sufficient to demonstrate effectiveness, it does provide an important property when considering it for model development or evaluation. Moreover, given that RPP and its position-weighted variants correlate well with existing metrics, these results suggest that the RPP family may be a more sensitive set of instruments for the same phenomenon.

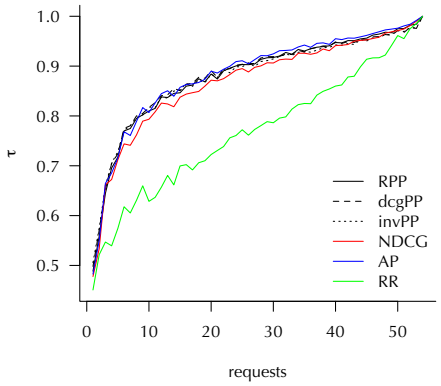
7 DISCUSSION

Our experiments were designed to understand if RPP captured properties of existing metrics with the benefit of added sensitivity because of better label efficiency. Our correlation results (Section 6.1) support the claim that RPP and variants can measure aspects similar to existing metrics while our robustness to incomplete data experiments (Section 6.2) demonstrate that RPP is as robust to incomplete data as NDCG, an existing metric known to be robust to incomplete data. Our strongest result suggests that, while capturing these properties of existing metrics, RPP is substantially more sensitive (Section 6.3). As a result, RPP can complement the existing suite of evaluation metrics, including less sensitive but more realistic metrics based on a domain's user behavior.

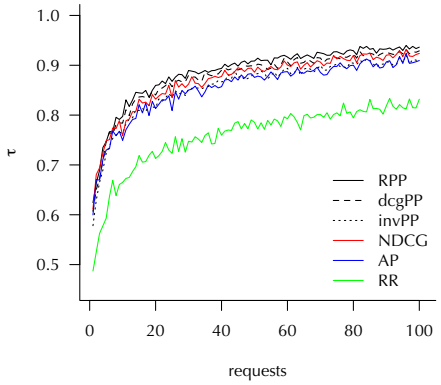
Our philosophy in designing RPP was to minimize the number of assumptions about user behavior, while being flexible enough to model them, as needed. We demonstrated that incorporating user models through, for example, $p(i)$ could increase correlation with existing metrics (Section 6.1) while maintaining RPP's strong discriminative power (Section 6.3). We believe that careful incorporation of models of user behavior can further improve the grounding of RPP while preserving its discriminative power. For example, referring to Figure 1b, given a labeled preference, we can imagine learning the weights on different pseudo-populations. Labeled preferences could come from editorial data or from behavioral data such as an interleaving experiment. The former would be similar to the approach taken by Hassan Awadallah and Zitouni [13] for top k rankings.



(a) Robust

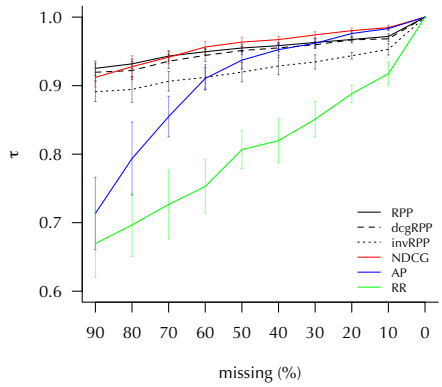


(b) Deep Learning Passage Ranking (2020)

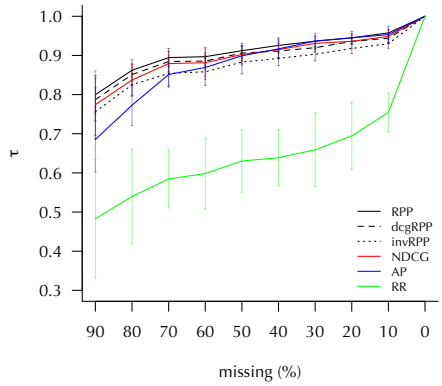


(c) MovieLens 1M

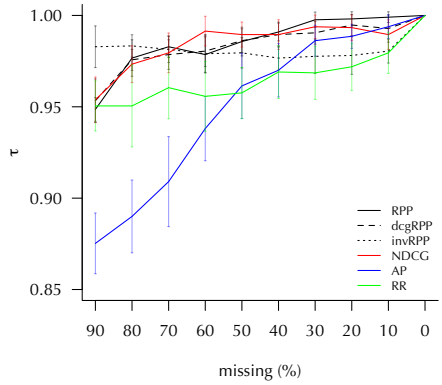
Figure 6: Kendall's τ of a system ranking given k requests with a system ranking given all requests. Only the first 100 users are shown for ml-1M since metrics converge quickly after.



(a) Robust



(b) Deep Learning Passage Ranking (2020)



(c) MovieLens 1M

Figure 7: Kendall's τ of a system ranking given missing judgments with a system ranking given all judgments.

Table 3: Percentage of run differences detected at $p < 0.05$ using Tukey's HSD test.

(a) Single Topic Metrics

	RPP	dcgRPP	invRPP	AP	NDCG	RR
core (2017)	**49.91**	49.44	40.76	36.54	39.21	14.67
core (2018)	**52.35**	49.22	42.33	35.29	34.66	27.97
deep-docs (2019)	**42.53**	40.83	30.01	33.57	41.11	6.97
deep-docs (2020)	18.85	**19.35**	17.31	17.71	18.06	2.88
deep-pass (2019)	**42.34**	42.19	36.34	30.03	26.73	6.76
deep-pass (2020)	45.70	**47.34**	45.82	30.10	20.87	22.21
web (2009)	32.00	33.69	**35.02**	18.35	31.56	23.23
web (2010)	**43.75**	39.31	28.43	27.82	37.50	18.15
web (2011)	**45.16**	43.36	34.06	31.62	40.61	14.17
web (2012)	**41.05**	36.44	23.49	26.06	32.09	12.77
web (2013)	**48.42**	44.10	22.30	29.02	41.53	5.96
web (2014)	48.74	48.28	36.55	44.14	**49.89**	18.85
robust	**63.47**	60.52	51.08	52.89	54.53	22.84
ml-1M	**94.29**	**94.29**	92.38	88.57	88.57	83.81
libraryThing	96.67	97.14	**97.62**	95.24	95.71	93.33
beerAdvocate	94.76	94.76	**95.71**	93.33	94.29	91.90

(b) Subtopic Metrics

	ST-RPP	ST-R	ERR-IA	MAP-IA
web (2009)	**29.43**	28.99	24.20	17.11
web (2010)	**40.12**	28.02	22.18	20.77
web (2011)	**46.11**	17.93	16.45	26.12
web (2012)	**44.15**	12.15	16.58	25.09
web (2013)	**49.89**	4.48	5.52	24.04
web (2014)	**50.11**	12.87	17.93	40.46

Table 4: Percentage of run differences detected at $p < 0.05$ using the Student's t-test with Bonferroni correction.

(a) Single Topic Metrics

	RPP	dcgRPP	invRPP	AP	NDCG	RR
core (2017)	**54.13**	53.66	45.26	34.13	36.54	9.55
core (2018)	**55.59**	53.99	48.63	35.84	40.18	21.91
deep-docs (2019)	**49.64**	44.81	37.27	28.45	36.13	4.84
deep-docs (2020)	18.01	**19.15**	17.36	14.34	15.58	0.50
deep-pass (2019)	43.99	**45.05**	42.34	24.17	23.72	1.65
deep-pass (2020)	51.49	54.59	**55.93**	34.83	33.31	18.35
web (2009)	34.22	**37.06**	36.70	15.25	34.04	19.50
web (2010)	**52.22**	45.16	30.04	22.58	31.25	13.71
web (2011)	**54.42**	49.60	35.64	32.36	40.93	6.98
web (2012)	**44.59**	39.98	24.47	22.25	32.45	10.64
web (2013)	**59.24**	51.26	23.01	25.52	40.82	2.51
web (2014)	59.31	56.09	38.16	51.49	**62.30**	13.10
robust	**65.00**	62.29	54.21	44.82	49.34	16.21
ml-1M	**96.19**	95.71	**96.19**	91.43	94.29	85.71
libraryThing	**98.57**	98.10	**98.57**	94.29	96.67	92.38
beerAdvocate	95.24	95.24	**95.71**	91.43	**95.71**	91.90

(b) Subtopic Metrics

	ST-RPP	ST-R	ERR-IA	MAP-IA
web (2009)	**33.24**	25.62	19.06	14.36
web (2010)	**49.40**	16.94	14.72	11.69
web (2011)	**56.32**	8.99	9.52	23.64
web (2012)	**43.71**	7.98	12.06	14.45
web (2013)	**60.77**	2.35	2.90	19.45
web (2014)	**63.68**	9.89	12.18	46.44

Chapelle et al. [8] demonstrated the sensitivity of interleaving experiments across a variety of online search scenarios. At the same time, the distribution $p(k)$ is likely to be skewed toward top rank positions, resulting in an under-weighting of higher values of k in Equation 2. Because this can result in lower label efficiency, using a more uniform $p(k)$ could improve the sensitivity of online interleaving.

Although we have presented RPP as a way to evaluate systems, how to optimize RPP is an area for future research. On the one hand, uniformly weighing the importance of different recall levels is similar to methods that train models with a sequence of tasks based on a sampled relevant item combined with sampled negative items [18].Under these approaches, the model learns to rank all relevant items, weighting them equally. In the context of evaluation, this is similar to uniformly weighting all recall levels (i.e. $p(i) = \frac{1}{m}$). Our results demonstrate that this may be a more robust way to optimize rankers.

8 CONCLUSION

We have presented recall-paired preference (RPP), a method for evaluating rankings that avoids first computing an evaluation metric. Through extensive experimentation, we demonstrate that RPP reflects many properties of existing metrics with a substantially improved sensitivity. We believe that this approach can be extended in multiple directions, including refining user models while preserving its sensitivity.

REFERENCES

[1] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying Search Results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM '09)*. Association for Computing Machinery, New York, NY, USA, 5–14.

[2] John Alex, Keith Hall, and Donald Metzler. 2022. Atomized Search Length: Beyond User Models. *CoRR* abs/2201.01745 (2022). arXiv:2201.01745 https://arxiv.org/abs/2201.01745

[3] Thomas M. Braun and Todd A. Alonzo. 2007. A modified sign test for comparing paired ROC curves. *Biostatistics* 9, 2 (10 2007), 364–372.

[4] Chris Buckley and Ellen M. Voorhees. 2004. Retrieval Evaluation with Incomplete Information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '04)*. ACM, New York, NY, USA, 25–32.

[5] Ben Carterette. 2011. System effectiveness, user models, and user utility: a conceptual framework for investigation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR '11)*. ACM, New York, NY, USA, 903–912.

[6] Ben Carterette and Paul N. Bennett. 2008. Evaluation measures for preference judgments. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, New York, NY, USA, 685–686.

[7] Benjamin A. Carterette. 2012. Multiple testing in statistical analysis of systems-based information retrieval experiments. *ACM Trans. Inf. Syst.* 30, 1 (March 2012), 4:1–4:34.

[8] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. 2012. Large-Scale Validation and Analysis of Interleaved Search Evaluation. *ACM Trans. Inf. Syst.* 30, 1 (March 2012).

[9] Charles L. A. Clarke, Alexandra Vtyurina, and Mark D. Smucker. 2021. Assessing Top-k Preferences. *ACM Trans. Inf. Syst.* 39, 3 (may 2021).

[10] William S. Cooper. 1968. Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American Documentation* 19, 1 (1968), 30–41.

[11] Nick Craswell, Bhaskar Mitra, Daniel Campos, Emine Yilmaz, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 Deep Learning Track. In *Proceedings of the Twenty-Eighth Text REtrieval Conference (TREC 2019)*.

[12] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. 2001. Rank Aggregation Methods for the Web. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. Association for Computing Machinery, New York, NY, USA, 613–622.

[13] Ahmed Hassan Awadallah and Imed Zitouni. 2014. Machine-Assisted Search Preference Evaluation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM '14)*. Association for Computing Machinery, New York, NY, USA, 51–60.

[14] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *TOIS* 20, 4 (2002), 422–446.

[15] Thorsten Joachims. 2003. *Evaluating Retrieval Performance Using Clickthrough Data.* Physica/Springer Verlag, 79–96.

[16] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17)*. Association for Computing Machinery, New York, NY, USA, 781–789.

[17] Jinyoung Kim, Gabriella Kazai, and Imed Zitouni. 2013. Relevance Dimensions in Preference-Based IR Evaluation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '13)*. Association for Computing Machinery, New York, NY, USA, 913–916.

[18] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, New York, NY, USA, 1748–1757.

[19] Walid Magdy and Gareth J.F. Jones. 2010. PRES: A Score Metric for Evaluating Recall-Oriented Information Retrieval Applications. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10)*. Association for Computing Machinery, New York, NY, USA, 611–618.

[20] Alistair Moffat and Justin Zobel. 2008. Rank-biased Precision for Measurement of Retrieval Effectiveness. *ACM Trans. Inf. Syst.* 27, 1 (Dec. 2008), 2:1–2:27.

[21] Stephen Robertson. 2008. A New Interpretation of Average Precision. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*. Association for Computing Machinery, New York, NY, USA, 689–690.

[22] Stephen E. Robertson, Evangelos Kanoulas, and Emine Yilmaz. 2010. Extending Average Precision to Graded Relevance Judgments. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10)*. Association for Computing Machinery, New York, NY, USA, 603–610.

[23] Joseph John Rocchio. 1966. *Document Retrieval Systems - Optimization and Evaluation.* Ph.D. Dissertation. Harvard University, Cambridge, MA.

[24] Tetsuya Sakai. 2014. Metrics, statistics, tests. In *Bridging Between Information Retrieval and Databases - PROMISE Winter School 2013, Revised Tutorial Lectures (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics))*. Springer Verlag, 116–163.

[25] Tetsuya Sakai and Stephen Robertson. 2008. Modelling A User Population for Designing Information Retrieval Metrics. In *Proceedings of The Second International Workshop on Evaluating Information Access (EVIA)*.

[26] Mark Sanderson, Monica Lestari Paramita, Paul Clough, and Evangelos Kanoulas. 2010. Do User Preferences and Evaluation Measures Line Up?. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10)*. Association for Computing Machinery, New York, NY, USA, 555–562.

[27] Paul Thomas and David Hawking. 2006. Evaluation by Comparing Result Sets in Context. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM '06)*. Association for Computing Machinery, New York, NY, USA, 94–101.

[28] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. 2020. Assessing ranking metrics in top-N recommendation. *Information Retrieval Journal* 23, 4 (2020), 411–448.

[29] Haotian Zhang, Mustafa Abualsaud, Nimesh Ghelani, Mark D. Smucker, Gordon V. Cormack, and Maura R. Grossman. 2018. Effective User Interaction for High-Recall Retrieval: Less is More. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 187–196.