How do Ties Affect the Uncertainty in Rank-Biased Overlap?

Corsi, Matteo; Urbano, Julián

DOI
10.1145/3673791.3698422

Publication date
2024

Document Version
Final published version

Published in
SIGIR-AP 2024 - Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region

How do Ties Affect the Uncertainty in Rank-Biased Overlap?

Matteo Corsi
Delft University of Technology
Delft, The Netherlands
m.corsi@tudelft.nl

Julián Urbano
Delft University of Technology
Delft, The Netherlands
j.urbano@tudelft.nl

Abstract

Rank-Biased Overlap (*RBO*) is a popular measure of the similarity between two rankings. A key characteristic of *RBO* is that it can be computed even when the rankings are not fully seen and only a prefix is known, but this introduces uncertainty in the computation. In such cases, one would normally compute the point estimate RBO_{EXT}, as well as bounds representing the best and worst cases; their difference is thus a residual quantifying the amount of uncertainty. Another source of uncertainty is the presence of tied items, because their actual relative order is unknown. Current approaches to this issue similarly provide a point estimate by considering the average *RBO* score over all the permutations of the ties, such as RBO^a. However, there is currently no approach to quantify and bound the uncertainty due to ties, just as there is for the uncertainty due to unseen items. In this paper we fill this gap and provide algorithmic solutions to the problem of finding the arrangements of tied items that yield the lowest and highest possible *RBO* scores, naturally leading to total bounds and residuals. We also show that the current RBO^a estimate only equals the average *RBO* over permutations when the rankings have the same length, so we also generalize it to rankings of different lengths. In summary, this work provides a full account for the uncertainty in *RBO*, allowing practitioners to make more sensible decisions on the grounds of rank similarity. The main realization is that residuals can actually be *much* larger once we account for both sources of uncertainty. To illustrate this, we present empirical results using both synthetic and TREC data, demonstrating that a realistic picture for the residual of *RBO* can only be provided by considering both sources of uncertainty.

CCS Concepts

• **Information systems** → **Evaluation of retrieval results**; • **Mathematics of computing** → **Exploratory data analysis**.

Keywords

Rank correlation, rank similarity, rank-biased overlap, ties, uncertainty, bounds

ACM Reference Format:
Matteo Corsi and Julián Urbano. 2024. How do Ties Affect the Uncertainty in Rank-Biased Overlap?. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP '24), December 9–12, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3673791.3698422

SIGIR-AP '24, December 9–12, 2024, Tokyo, Japan

ACM ISBN 979-8-4007-0724-7/24/12
https://doi.org/10.1145/3673791.3698422

1 Introduction

In Information Retrieval (IR) and Recommender Systems (RecSys) we often compare rankings created by different criteria, such as rankings of items recommended to a user, candidate terms for query expansion, topics for document clustering, or systems by evaluation metric score [1, 7, 10, 14, 22, 27, 30, 37, 43]. Some of the most frequently used measures of ranking similarity are based on the notion of rank correlation, such as τ [15], ρ [31], D [17], τ_* [24], d_{rank} [8], τ_{ap} [35, 41], K^* and F'^* [18], and τ_w [36].

But rank correlation cannot be used when the rankings are non-conjoint or incomplete. This happens for example when Web search engines likely have different portions of the Web in their indexes, so that some pages retrieved by system A cannot even be retrieved by system B, or when they truncate the results at different depths. Alternatives to handle non-conjoint rankings include adaptations of the Hoeffding distance [33], Spearman's footrule [3, 13], and IR metrics [4, 34]. The most popular alternative is Rank-Biased Overlap (*RBO*), by Webber et al. [40]. In addition to non-conjoint and incomplete rankings, it can handle rankings of different lengths and is top-weighted. *RBO* is often employed in IR and RecSys research for example to assess consistency of systems to query variations [2], compare system outputs [7, 27, 38], measure topic similarity [1, 22], or compare rankings in general [6, 9, 25, 29, 42].

1.1 Uncertainty in *RBO*

When computing *RBO*, we need to recognize two sources of uncertainty: unseen items and tied items. **Uncertainty due to unseen items** arises because rankings are usually truncated after a certain depth: due to the very nature of rankings, what items appear after a sufficiently deep rank is negligible. For example, a push notification may contain only the top 1 recommendation, and a typical TREC run contains only the top 1,000 documents per topic. As a consequence, rankings actually consist of a *seen* part or prefix, and an *unseen* part. As acknowledged by Webber et al. [40] when presenting *RBO*, these unseen items introduce uncertainty because it is unknown if they overlap or not. To account for this uncertainty, they proposed to calculate a point estimate, named RBO_{EXT}, and the bounds that *RBO* would take in the best and worst possible arrangements of the unseen parts, namely RBO_{MAX} and RBO_{MIN} (see Figure 1). To quantify uncertainty in a single number, Webber et al. suggested computing the residual $RBO_{MAX} - RBO_{MIN}$, and reporting it whenever its magnitude is relevant.

Uncertainty due to tied items arises whenever two or more items appear at the same rank because their relative order, for whatever reason, is unknown. These items are said to be "tied", and they also introduce uncertainty when calculating *RBO*. In IR, ties may appear for example when two documents have the same retrieval status value for a query [20, 23, 28]. The treatment of such ties in correlation problems dates back to the early 1900's [26, 32],

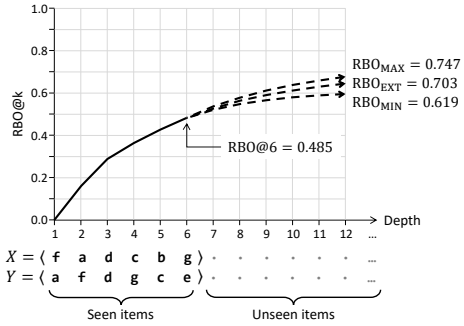

Figure 1: Uncertainty due to unseen items ($p = 0.8$). *RBO* at the end of the seen part is 0.485. The best and worst possible arrangements of the unseen items are represented by the upper and lower dashed lines, extending up to infinity. The middle dashed line represents the point estimate RBO_{EXT}.

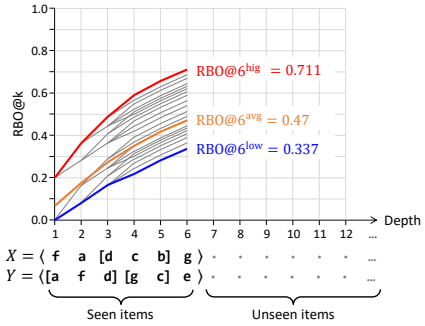

Figure 2: Uncertainty due to tied items (from Figure 1; square brackets indicate groups of tied items). Grey lines represent *RBO* scores for all possible permutations of the ties. The red, orange and blue lines represent the highest, average and lowest *RBO* score over permutations, respectively.

where the work by Kendall [16] is probably the most recognizable one. He developed two variants of his τ, namely τ_a and τ_b, via an stochastic interpretation of ties. In particular, he considered the expected correlation over all the possible permutations of the tied items, so that they would appear in one order half the times, and in the reverse order the other half; the difference later strives in that the b-variant further corrects the score by the amount of ties present in the rankings. More recently, the same approach has been adopted in the IR literature to develop tie-aware variants of τ_{ap} [35] and a weighted variant of τ_b [36].

For *RBO*, we have recently developed a- and b- variants as well [11]. Let us consider the sample rankings from Figure 1 but with some tied items, as illustrated in Figure 2. There are $3! = 6$ possible arrangements of the tied items in X, and $2! \times 3! = 12$ arrangements of Y, for a total of 72 possible pairs of rankings to compute *RBO*. In the absence of any other information, all those 72 arrangements are equally likely, so it is natural to compute the average *RBO* over all permutations, as a sort of expected *RBO* if one was to break ties at random. In the spirit of Kendall's τ_a, this is precisely what the a-variant, namely RBO^a, computes [11].

1.2 Should We Care about Uncertainty?

It is important to note that *RBO* does *not* create this uncertainty. When a ranking is represented just with a prefix, it is the ranking, in and on itself, the one that bears uncertainty, which is then propagated on to the *RBO* calculation. Being aware of this uncertainty allows for a more sensible use of *RBO*. For instance, Figure 1 reflects an example in which the *RBO* score could be anywhere between 0.619 and 0.747 if we were able to fully observe the rankings. In some situations, this may be just too much uncertainty to make a decision. At the very least, it allows us to acknowledge and quantify how much uncertainty there is in case a decision is made.

Likewise, a ranking containing ties bears uncertainty, which is propagated on to the *RBO* calculation. As Figure 2 shows, scores may vary substantially depending on the specific arrangement of tied items, and in particular the best and worst possible cases differ by as much as 0.374. This is an extremely high amount of variability that would go unnoticed when reporting a single number.

One could choose to ignore uncertainty. Indeed, once the prefixes are sufficiently long the uncertainty due to unseen items becomes negligible. However, while practitioners may often be able to control the prefix length, they have no control over the presence or absence of ties, for they are an intrinsic property of the rankings themselves. To illustrate, Cabanac et al. [5] and Corsi and Urbano [11] showed that the majority of typical TREC runs do contain ties, with dozens of documents having the same retrieval status values. One could still counter-argue that ties can always be broken, but it does not seem reasonable to artificially modify a ranking just for the sake of comparing it to another one. In addition, breaking ties at random unnecessarily introduces noise, and breaking by a deterministic criterion is even worse because it inflates scores [11]. Simply neglecting uncertainty is not an option. We need to acknowledge it, embrace it, and make it an integral part of our work.

1.3 Contribution

Simply put, RBO_{EXT} just *handles* the uncertainty due to unseen items by providing a point estimate, but it is RBO_{MIN} and RBO_{MAX} who actually *quantify* the amount of uncertainty around this estimate. In the same way, the formulations for RBO^a and RBO^b simply handle the uncertainty due to tied items, but there is currently no work on the quantification of that uncertainty, neither with bounds nor with residuals. Therefore, in this paper we address this gap:

(1) Section 3 provides solutions for the lowest and highest possible scores over permutations of the ties, namely RBO^{low} and RBO^{hig}, as well as a proper formulation for the average, namely RBO^{avg}, which deals with a subtle pitfall of RBO^a.
(2) Section 4 illustrates the importance of accounting for uncertainty with real world TREC data and synthetic data.
(3) Section 5 presents detailed discussion and guidelines about how to compute and report *RBO*.

Together with [11, 40], this work thus provides a full account for the uncertainty in Rank-Biased Overlap, as illustrated in Figure 6. All the results can be reproduced with data and code available online.[1]

[1] https://github.com/matteo-corsi/sigir_ap24

2 Ranked-Biased Overlap

Given two indefinite rankings L and S, RBO is defined as an infinite sum of weighted agreements at increasing depths:

$$RBO = \frac{1-p}{p} \sum_{d=1}^{\infty} A_d \cdot p^d \,, \tag{1}$$

where A_d is the agreement between S and L at depth d, and p^d is the corresponding weight; the persistence parameter p determines how the weight decays as a function of the depth [40].

In the absence of ties, the agreement is defined as the fraction of items up to depth d that are common to both rankings:

$$A_d = \frac{X_d}{d} = \frac{|L_{:d} \cap S_{:d}|}{d} \,, \tag{2}$$

where X_d is the overlap between S and L up to depth d. RBO ranges between a maximum of 1 when two identical rankings are compared and a value of 0 when the rankings have no items in common.

2.1 Uncertainty due to Unseen Items

Despite being defined for infinite rankings, in practice RBO is always computed from a prefix or seen part of the rankings. To better illustrate, let s and l refer to the lengths of S and L, respectively, where S is generally the shorter one (i.e. $s \leq l$). Following Corsi and Urbano [11], let us rewrite RBO in eq. (1) as follows:

$$RBO = \frac{1-p}{p} \left[\underbrace{\sum_{d=1}^{s} A_d p^d}_{1} + \underbrace{\sum_{d=s+1}^{l} A_d p^d}_{2} + \underbrace{\sum_{d=l+1}^{\infty} A_d p^d}_{3} \right], \tag{3}$$

where items in the first summation are seen in both rankings, items in the second summation are seen only in the longer ranking, and items in the third summation are not seen in any ranking. The unseen items in the second and third summations are the ones responsible for the first source of uncertainty.

The lower bound, RBO_{MIN}, is computed by assuming that all these unseen items do not overlap [40]. In other words, it is assumed that the rankings are non-conjoint. Noting that overlap is always X_l in the third summation, after some rearrangement we obtain:

$$RBO_{MIN} = \frac{1-p}{p} \left[\sum_{d=1}^{l} A_d \; p^d + X_l \left[\ln\left(\frac{1}{1-p}\right) - \sum_{d=1}^{l} \frac{p^d}{d} \right] \right]. \tag{4}$$

The opposite assumption is made for the upper bound RBO_{MAX}: all unseen items match items that remained unmatched from the other ranking [40]. In particular, in the second summation it is assumed that every unseen item in S matches an item from L. In the third summation, every unseen item in L matches something still unmatched from S and vice-versa, thus increasing overlap by +2 at every depth. This happens up to depth $f = l + s - X_l$, after which it is assumed that the same item appears in both rankings, increasing overlap by +1. After some rearrangement, we obtain:

$$RBO_{MAX} = \frac{1-p}{p} \left[\sum_{d=1}^{s} A_d \; p^d + \sum_{d=s+1}^{l} \frac{X_d + d - s}{d} p^d + \right. \\ \left. + \sum_{d=l+1}^{f} \frac{2d - l - s + X_l}{d} p^d \right] + p^f \,. \tag{5}$$

In Figure 1, RBO_{MIN} and RBO_{MAX} values at increasing depths are represented by the lower and upper dashed lines. The residual due to unseen items can thus be calculated as $RES_U = RBO_{MAX} - RBO_{MIN}$.

For the point estimate RBO_{EXT}, it is assumed that the seen agreement remains constant throughout the unseen parts [40]. In order to achieve this, the $d - s$ unseen items in S in the second summation are assumed to contribute fractionally by an amount equal to $A_s = X_s/s$. For the third summation, agreement assumed at depth l continues up to infinity, leading to:

$$RBO_{EXT} = \frac{1-p}{p} \left[\sum_{d=1}^{s} A_d \; p^d + \sum_{d=s+1}^{l} \frac{X_d + (d-s)A_s}{d} p^d \right] + \\ + \frac{X_l + (l-s)A_s}{l} p^l \,. \tag{6}$$

A stochastic interpretation of these choices is detailed in [11, Section 4]. In Figure 1, RBO_{EXT} values at increasing depths are represented by the mid dashed line.

2.2 Treatment of Ties

Webber et al. [40] briefly contemplated the case where tied items *really* occur at the same rank (i.e. the "sports" rankings), later coined RBO^w by Corsi and Urbano [11]. But this interpretation of ties does not bear to the idea of uncertainty, so they introduced variants RBO^a and RBO^b, akin to Kendall's τ_a and τ_b, thus following the interpretation typically found in the Statistics literature. Specifically, they approached the problem stochastically by considering all the possible arrangements of the tied items, and computing the expected overlap, X^a_d, over all such permutations. RBO^a is then computed using the expected agreement $A^a_d = X^a_d/d$ (see Section 3.3 for more details). In contrast, the agreement for RBO^b does not normalize X^a_d with d, but with the amount of measurable overlap at depth d. In other words, it corrects for the amount of ties. Finally, they followed the same rationale by Webber et al. to quantify the uncertainty due to unseen items in these three tie-aware variants. As such, the current literature offers ways to handle both sources of uncertainty, but it is still unknown how to *quantify* the uncertainty introduced by tied items. We fill this gap next.

3 Uncertainty due to Ties

Section 2.1 described how to deal with the uncertainty due to unseen items by defining bounds and a point estimate for RBO. Inspired by this, in this section we deal with the uncertainty due to ties in the seen part by similarly deriving bounds and a point estimate. Specifically, let us consider all the possible arrangements of the tied items: the permutations that minimize overlap at each depth will lead to the lowest possible score, namely RBO^{low}, and those that maximize it will lead to the highest possible score, namely RBO^{hig}. Our goal is to derive such permutations to compute the bounds.

A brute-force approach that calculates RBO for all possible permutations is off the table, for the number of permutations grows factorially with the number of ties. To put this into perspective, we note that rankings by a typical TREC Web run have more permutations than atoms in the observable universe [12]. In addition, we note that there may be multiple solutions to this problem. For instance, permutations ⟨a d c i m h b e⟩ and ⟨m e b c d a n⟩

$d =$ 1 2 3 4 5 6 7 8 9 10 11 12 …

$L = \langle$ a [i d m c] [e b h]⟩ · · · · ·

$S = \langle$ m [b a e c d] n⟩ · · · · · ·

Figure 3: Main example used in Section 3. Colored letters represent tied items, and square brackets represent tie groups.

lead to the RBO^{low} for the example in Figure 3, but so do permutations ⟨a c d i m h e b⟩ and ⟨m b e d c a n⟩. Finally, we note that finding RBO^{low} and finding RBO^{hig} are *not* dual problems. Let us use $\overleftarrow{X}$ to denote X reversed, and X^{low} for a permutation leading to RBO^{low}. One could intuitively hope that $Y^{low}|X = Y^{hig}|\overleftarrow{X}$ and vice-versa, so that only one algorithm would really be needed, but this is unfortunately not the case. The reason is again that there are multiple solutions to $Y^{low}|X$, so that an algorithm would ultimately still be needed to compute a valid Y^{hig}.[2]

3.1 Permutations with Lowest Score: RBO^{low}

In order to derive the permutations that lead to the lower bound RBO^{low}, we must decide how to arrange tied items and in which order. Every choice we make as to the rank of an item has a direct influence on the possible choices we can make for the other items. Therefore, it is essential to establish an order of priority, for which we consider two principles:

- P1: delay overlap. Items should be ranked as deep as possible, so that their contribution to overlap is delayed the most.
- P2: leave room. P1 should apply to one ranking only; in the other one the item should actually be ranked as early as possible so that it leaves room for other items to also delay overlap.

Let us consider the example in Figure 3. Principle P1 tells us to fix c at rank 6 in S so that its overlap is delayed the most. But then, it does not need to be ranked deep in L, because it would not make a difference with respect to the delay. Instead, rank 5 in L could be taken by another item from the group in order to delay its overlap, such as d. Therefore, P2 tells us to rank c at rank 2 in L, so as to make up the most room to delay overlaps in the red group.

To accommodate principle P1, let us define the overlap interval for an item e as the set of all ranks at which that item could finally contribute to overlap; let us refer to the top and bottom ranks of such interval as τ_e and β_e, respectively. The length of the overlap interval gives an indication of how much room we have to delay the item's overlap, so we first prioritize items with long intervals. In order to accommodate principle P2, let us define δ_e as the maximum distance between the ranks at which item e may appear in both rankings. For example, the distance for item b is maximized when it is placed at the top of the blue group and at the bottom of the green one. This distance can then be used as the second criterion to prioritize which items to fix, as it would maximize the room left for other items to delay their overlap. In practice then, by P1 we will place an item as deep as possible in one ranking, and by P2 we will place it as early as possible in the other ranking.

Algorithm 1 presents a summary of procedure PermutationsLowest, which generates two permutations of S and L that lead to the lowest possible *RBO* score.[3] The algorithm aims precisely at prioritizing principles P1 and P2 above:

(1) Lines 3–6: Ω represents the items in common between the two rankings. For each of these, we compute the deepest position that they can take in each ranking (i.e. the bottom ranks b_{eS} and b_{eL}), and the earliest (i.e. the top ranks t_{eS} and t_{eL}). In Figure 3, we would have for example $b_{eL} = b_{bL} = b_{hL} = 8$ and $t_{bS} = t_{aS} = t_{eS} = t_{cS} = t_{dS} = 2$. We also compute the bounds of the overlap interval, $\tau_e = \max(t_{eS}, t_{eL})$ and $\beta_e = \max(b_{eS}, b_{eL})$, and the maximum distance $\delta_e = \max(b_{eS} - t_{eL}, t_{eS} - b_{eL})$. For c in the example, we would have $\tau_c = 2$, $\beta_c = 6$ and $\delta_c = 4$ (when ranked at the top of the red group and at the bottom of the blue one).
(2) Lines 12–13: the item f to be fixed next is selected, prioritizing longer overlap intervals and then longer distances. By default, its new ranks r_S and r_L are set at the bottom of its groups. In the example, item m would initially be assigned $r_S = 1$ and $r_L = 5$.
(3) Lines 14–20: we identify the ranking where the chosen item can be placed the deepest, and in the other ranking it is placed as

Algorithm 1 Compute permutations for RBO^{low} scores.

```
1:  procedure PermutationsLowest(S, L)
2:      # Compute top/bottom ranks and possible overlaps
3:      Ω ← {S ∩ L}
4:      for all e ∈ Ω do
5:          compute t_eS, t_eL, b_eS, b_eL, τ_e, β_e and δ_e
6:      end for
7:      # Initialize output rankings
8:      S*, L* ← empty rankings of lengths s and l
9:
10:     while |Ω| > 0 do
11:         # Select next item to fix and decide its final ranks
12:         f ← arg_{e∈Ω} by max(β_e − τ_e) then by max δ_e
13:         r_S ← b_fS and r_L ← b_fL
14:         if f is tied in S or f is tied in L then
15:             if b_fL > b_fS or (b_fL = b_fS and t_fS < t_fL) then
16:                 r_S ← t_fS
17:             else
18:                 r_L ← t_fL
19:             end if
20:         end if
21:         # Fix in the output rankings
22:         S*_{r_S} ← L*_{r_L} ← f and Ω ← Ω/{f}
23:
24:         # Update top/bottom ranks and possible overlaps
25:         for all e ∈ Ω do
26:             recompute t_eS, t_eL, b_eS, b_eL, τ_e, β_e and δ_e
27:         end for
28:     end while
29:
30:     # Items in only one ranking do not make a difference
31:     for all e ∈ {S ∪ L}/{S ∩ L} do
32:         fix e in a random empty spot
33:     end for
34:
35:     return S*, L*
36: end procedure
```

[2] We do not elaborate further because of space restrictions. The interested reader may refer to the toy example $X = \langle$d [a c]⟩, $Y = \langle$[c d] b⟩ to illustrate this.

[3] For simplicity, both Algorithm1 and 2 are described for rankings of the same length, but the generalization is straightforward with proper checks for $d \leq s$.

early as possible, thus complying with principles P1 and P2.For instance, if the deepest rank in L is greater than the one in S (i.e. $b_{fL} > b_{fS}$), then f will be assigned rank t_{fS} in S. For instance, item c would be assigned rank 2 in L and rank 6 in S.

(4) Line 22: the selected item f is finally fixed in the output rankings S^* and L^* at ranks r_S and r_L, and it is removed from Ω.
(5) Lines 25–27: for the remaining items in Ω, we update the top and bottom ranks, as well as the bounds of the overlap interval and the maximum distance.
(6) Steps (2)–(5) are repeated until Ω is empty.
(7) Lines 31–33: all items that were not initially in Ω (i.e. they appear in only one ranking) can be placed at random in the remaining spots. Indeed, these items do not influence overlap once all the other items have been placed.

This algorithm returns two permutations of the originals, namely X^{low} and Y^{low}, that have no ties and yield the lowest possible *RBO* score among all possible permutations; these are illustrated in Figure 6 with solid blue lines. Computing eqs. (4), (5) and (6) with these two permutations, we can add the uncertainty due to unseen items on top of the lower bound of the uncertainty due to ties, resulting in the RBO^{low}_{*} scores displayed in Figure 6 with blue dashed lines. Again, recall that there may be multiple valid solutions, for example when there are several items with the same $\beta - \tau$ and δ in line 12, and one has to be chosen at random.

3.2 Permutation with Highest Score: RBO^{hig}

Similarly, we can find the permutations of S and L that lead to the highest possible *RBO* score. In this case, we want items to match as early as possible so that overlap is maximized. We can differentiate four cases at any given rank d:

- Case 1: in the best scenario, overlap may increase by +2 if the items appearing at that depth both match an item from the other ranking that was still unmatched.
- Case 2: the same item appears at rank d in both rankings, thus increasing overlap by +1.
- Case 3: if in one ranking we place an item still unmatched in the other ranking, and in this other ranking we place an item that does not match anything, overlap also increases by +1.
- Case 4: in the worst scenario, placing two still unseen items does not increase overlap.

Algorithm 2 presents a summary of procedure PermutationsHighest to generate the two permutations of S and L that lead to the highest possible *RBO* score. The algorithm aims precisely at prioritizing allocation of items following the four cases above:

(1) Lines 3–5: for each item, we calculate the deepest and earliest position that they can take in each ranking.
(2) Line 13: at a given depth d, we first look for possible items to fix, namely f_S and f_L, according to Case 1 above: if the item at that rank is not tied then it will be chosen to be fixed; if it is tied, we randomly select an item from the group that would match something in the other ranking, if possible. In the example, at rank $d = 2$ we would select $f_S =$ a and $f_L =$ m, increasing the overlap by +2.
(3) Lines 14–16: if there was no success looking for Case 1 (i.e. f_S or f_L are undefined), then we check for the possibility of Case 2: if there are candidate items shared by both rankings at depth d, we choose one at random to be fixed. In the example, item d is common to both rankings for rank $d = 3$, falling within Case 2. Item c is another valid candidate; let us assume it gets fixed at $d = 4$.
(4) Lines 17–19: at this point we may have made a decision about what to fix in one ranking but not the other. We check for the possibility of Case 3: we simply select an item at random from the group. In the example, item i would be fixed in L at rank 5 because it is the only available spot left in the group, and either b or e can be fixed in S, increasing overlap by +1.
(5) Lines 20–22: in case f_S or f_L remain undefined, Case 4 is used to fix two unseen items at random that will not increase overlap.

Algorithm 2 Compute permutations for RBO^{hig} scores.

```
1:  procedure PermutationsHighest(S, L)
2:      # Compute top/bottom ranks
3:      for all e ∈ {S ∪ L} do
4:          compute t_eS, t_eL, b_eS, and b_eL
5:      end for
6:      # Initialize output rankings
7:      S*, L* ← empty rankings of lengths s and l
8:      # Initialize sets of still unmatched items
9:      S̄ ← L̄ ← ∅
10:
11:     for d ∈ {1, ..., l} do
12:         # Select items that would maximize overlap, if possible
13:         f_S, f_L ← NextItemsCase1(S, L, d, S̄, L̄)
14:         if f_S =⊥ or f_L =⊥ then
15:             f_S, f_L ← NextItemsCase2(S, L, d, S̄, L̄, f_S, f_L)
16:         end if
17:         if f_S =⊥ or f_L =⊥ then
18:             f_S, f_L ← NextItemsCase3(S, L, d, S̄, L̄, f_S, f_L)
19:         end if
20:         if f_S =⊥ or f_L =⊥ then
21:             f_S, f_L ← NextItemsCase4(S, L, d, S̄, L̄, f_S, f_L)
22:         end if
23:
24:         # Fix in the output rankings
25:         S*_d ← f_S and L*_d ← f_L
26:
27:         # Update sets of still unmatched items
28:         if f_S ≠ f_L and f_S ∉ L̄ then
29:             S̄ ← S̄ ∪ {f_S} # New unmatched
30:         else
31:             L̄ ← L̄/{f_S} # Not unmatched anymore
32:         end if
33:         if f_L ≠ f_S and f_L ∉ S̄ then
34:             L̄ ← L̄ ∪ {f_L} # New unmatched
35:         else
36:             S̄ ← S̄/{f_L} # Not unmatched anymore
37:         end if
38:         # Update top ranks
39:         for all e ∈ {S ∪ L}/{S* ∪ L*} do
40:             recompute t_eS and t_eL
41:         end for
42:     end for
43:
44:     return S*, L*
45: end procedure
```

(6) Line 25: the selected items are actually fixed at rank d in the output rankings S^* and L^*.
(7) Lines 28–37: we update the sets of unmatched items $\bar{S}$ and $\bar{L}$, to reflect the matching or unmatching of the selected items.
(8) Lines 39–41: all the other items in the group crossed by d can no longer appear at rank d, so we update their top ranks to $d+1$ for the next iteration.
(9) Steps (2)–(8) are repeated for all depths from 1 to l.

This algorithm returns X^{hig} and Y^{hig}. They have no ties and yield the highest possible *RBO* score among all possible permutations; these are illustrated in Figure 6 with solid red lines. In the example from Figure 3, they are $X^{hig} = \langle$a m d c i b e h$\rangle$ and $Y^{hig} = \langle$m a d c b e n$\rangle$. Computing eqs. (4), (5) and (6) with these two permutations, we can add the uncertainty due to unseen items on top of the upper bound of the uncertainty due to ties, resulting in the RBO_*^{hig} scores displayed in Figure 6 with red dashed lines.

3.3 Average Score over Permutations: RBO^{avg}

Just as for the RBO^a variant by Corsi and Urbano [11], we ask the following question: what is the expected *RBO* when breaking ties at random? Their solution involves determining the average overlap over all permutations of the ties, from which an a-variant of agreement, and ultimately of *RBO*, could be formulated. To do this, they redefined overlap X_d as follows:

$$X_d^a = \sum_{e \in \Omega} c_{e,S|d} \cdot c_{e,L|d} \,, \tag{7}$$

where $\Omega = \{S \cup L\}$ represents all items, and the item contribution $c_{e,S|d}$ equals the fraction of permutations of S in which item e is ranked at or above depth d (untied items have a unitary contribution). As an example, consider ranking L in Figure 3 at depth $d = 4$: only three of the four tied items in the red group can be ranked at or above d in any given permutation. Across all possible permutations, each item is ranked at or above d precisely $3/4{-}th$ of the times. Defining the a-variant agreement as $A_d^a = X_d^a/d$, and plugging it into eq. (3), they were able to define RBO^a.

The problem arises when formulating RBO_{EXT}^a for rankings of different lengths, because in the second part (depths $s+1$ to l) some assumption needs to be made about unseen items in S. They computed the assumed agreement $\tilde{A}_d^a$ as the sum of two separate terms: overlap due to seen and unseen items:

$$\tilde{A}_d^a = \frac{\overbrace{X_d^a}^{\text{seen}} + \overbrace{(d-s) \cdot A_s^a \cdot \bar{c}_{L|d}}^{\text{unseen}}}{d} \,, \tag{8}$$

where, in the unseen section, A_s^a represents the probability that an unseen item in S matches an item still to be matched from L, while $\bar{c}_{L|d}$ is the average contribution of the still unmatched items in L. For details, please refer to [11, Section 4.1].

While A_d^a *is* precisely the "average agreement over permutations", $\tilde{A}_d^a$ actually makes RBO_{EXT}^a *different* from the desirable "average RBO_{EXT} over permutations" whenever the rankings have different lengths. Let us illustrate with a toy example: $\langle$a [b c d]$\rangle$ and $\langle$b a$\rangle$. According to (8), $\tilde{A}_3^a$ is $5/9 + 4/27 = 0.70$. However, if we computed the actual average agreement A_3 over all 6 permutations, we would obtain $(1{+}1{+}5/6{+}5/6{+}1/2{+}1/2)/6 = 0.78$, resulting from adding observed and extrapolated overlap in every permutation. Indeed, if we compare the tie-unaware extrapolation in eq. (6, 2nd summation) with the tie-aware one in eq. (8), we see that the average observed overlap is precisely X_d^a, the average overlap to extrapolate is precisely A_s^a, but the extra term $\bar{c}_{L|d}$ becomes redundant. Corsi and Urbano justified its inclusion in a larger context of several tie-aware variants of *RBO*, but it actually needs to be removed if we want an extrapolated formulation precisely equal to the average RBO_{EXT} over permutations:

$$\tilde{A}_d^{avg} = \frac{X_d^a + (d-s)A_s^a}{d} \,. \tag{9}$$

This way, we can then plug eqs. (7) and (9) into eqs. (4), (5) and (6) to compute RBO_{MAX}^{avg}, RBO_{MIN}^{avg} and RBO_{EXT}^{avg}. These coefficients *are* equal to the average of bare RBO_{MAX}, RBO_{MIN} and RBO_{EXT} computed over all possible permutations of tied items.[4]

Table 1: Summary of *RBO* residuals when accounting for items in the unseen part (RES_U), ties in the seen part (RES_S), and both (RES_{S+U}). M for medium residuals in $(0.01, 0.1]$, and L for large in $(0.1, 1]$. TREC data.

	RES_U				RES_S				RES_{S+U}			
p	Avg.	Max.	M	L	Avg.	Max.	M	L	Avg.	Max.	M	L
0.80	0.00	0.04	0%	0%	0.04	1.00	6%	7%	0.04	1.00	6%	7%
0.90	0.00	0.18	0%	0%	0.04	1.00	7%	7%	0.04	1.00	7%	7%
0.95	0.00	0.37	0%	0%	0.03	1.00	8%	5%	0.03	1.00	8%	6%

4 Experimental Demonstrations

In this section we illustrate the impact of computing partial residuals instead of total residuals, and how this might affect decisions made on the grounds of rank similarity. In particular, we compare the partial residual $RES_U = RBO_{MAX}^{avg} - RBO_{MIN}^{avg}$ due to unseen items, the partial residual $RES_S = RBO_{EXT}^{hig} - RBO_{EXT}^{low}$ due to ties in the seen part, and the total residual $RES_{S+U} = RBO_{MAX}^{hig} - RBO_{MIN}^{low}$ due to both (see Figure 6 for clarity). We will do this with two datasets, first with real TREC data and then with synthetic data, also varying the persistence parameter p between typical values 0.8, 0.9 and 0.99.

4.1 TREC Data

A clear application of *RBO* is comparing an experimental retrieval system with a baseline. Specially when relevance judgments are scarce or nonexistent, comparing the rankings they produce for a set of queries may provide a useful indication of their similarity. To explore this scenario, we use the adhoc runs from the TREC 2009–2014 Web track, comprising 255 systems by 95 groups ran over various sets of 50 topics per year. In particular, we compare every pair of systems by the same group and for every available topic, for a total of 12,750 comparisons. Of these, we focus on the 9,256 cases (73%) where the rankings contained tied documents. On average, they were 978 documents long, with a maximum of 1,000.

Table 1 shows a summary of the partial and total *RBO* residuals. First, we can appreciate that the residuals RES_U due to the unseen

[4] It can be shown that $RBO_{MIN}^a = RBO_{MIN}^{avg}$ because the extrapolated overlap is zero. However, $RBO_{MAX}^a \leq RBO_{MAX}^{avg}$ for a reason similar to that described above.

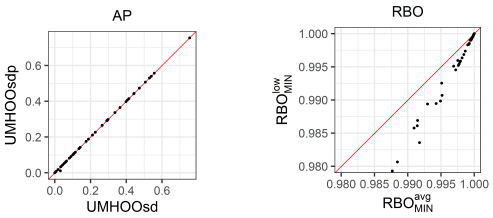

Figure 4: Left: comparison of per-topic *AP* scores of TREC 2009 Web runs UMHOOsd and UMHOOsdp. Right: comparison of per-topic RBO^{avg}_{MIN} and RBO^{low}_{MIN} scores (topic 29 not plotted for clarity).

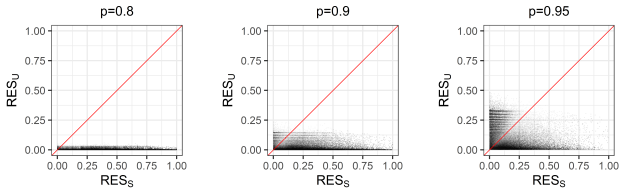

Figure 5: RES_U vs RES_S as a function of p. Synthetic data.

parts are essentially zero, because the rankings are long and the weight of the unseen items is negligible. But bounding only the unseen parts would have given the false impression of highly reliable comparisons. When we look at the uncertainty induced by ties, we can see that RES_S consistently captures higher residuals. This confirms that the majority of the uncertainty in *RBO* scores occurs in the seen parts. Given the characteristics of TREC data, the total residual RES_{S+U} is indeed essentially the same as RES_S. The average size of residuals tells us that *RBO* scores are overall quite reliable, but there are enough examples where this is not the case to warrant caution. Let us classify residuals into large ($0.1 \leq RES_*$), medium ($0.01 \leq RES_* < 0.1$) or small ($RES_* < 0.01$), roughly representing differences in the first, second, or third decimal digit of a reported *RBO* score, respectively; the first two are identified as L and M in Table 1. We can see that residuals are of medium size about 7% of the times, while large cases appear in another 7%. Therefore, residuals make a substantial difference in about 14% of the comparisons.

We can better illustrate the impact of the residuals with the example of two TREC 2009 runs: UMHOOsd used the Markov Random Field framework with features about individual term occurrences and term-dependencies, while UMHOOsdp adds clique pruning to optimize retrieval efficiency [19]. It is thus natural to consider UMHOOsd as the baseline system, and UMHOOsdp as the experimental one. Efficiency is indeed improved, but the question is whether effectiveness gets harmed in return. As shown in Figure 4 (left), this particular question had a positive answer because the *AP* scores remained the same in all but one topic (topic 29, bottom-left).

But imagine this experiment in a setting with low resources where judgments are scarce or unavailable. One could alternatively calculate the rank similarity between the outputs from both systems, hoping that it remains above some threshold, say 0.99, for most topics. Because these systems produce ties (17% of documents) one could calculate RBO^{avg}, but in order to prevent misjudgments due to the uncertainty brought by unseen items, it would be better to compute RBO^{avg}_{MIN}. Doing so, it would be found that similarity stays above the threshold in 44 of the 50 topics. However, someone now also aware of the uncertainty brought by tied items should compute RBO^{low}_{MIN} instead, that is, the total lower bound. Doing so, it would be found that similarity stays above the threshold in 37 topics (see Figure 4-right). Given this outcome, it might be decided to not replace the baseline system on the grounds of too much uncertainty, or perhaps to collect some relevance judgments and actually compare the effectiveness of the 13 topics where *RBO* fell below the threshold before making the decision. This is an example where full awareness of the total *RBO* residual would make practitioners take action with more caution.

4.2 Synthetic Data

While the previous results with TREC data are of interest to the general IR reader, they do not generalize to non-IR settings: TREC rankings are so long that $RES_U \approx 0$, but in other settings they may be considerably smaller. In order to provide more general results, here we resort to synthetic data. Specifically, we generate two rankings over the same 1,000 items and a certain degree of similarity (Kendall's τ randomly chosen between 0.5 and 1). Then, we randomly induce ties independently in each ranking, ensuring that at least 10% of items are tied. Finally we randomly truncate the rankings such that they have lengths between 10 and 100 items. This procedure is repeated a total of 100,000 times, producing pairs of rankings with an average length of 55 items, a length difference of 30 items, and 54% of ties on average.

The size of RES_S relative to RES_U depends on the interplay between the length of the rankings and the persistence parameter p (see Figure 5). Indeed, low p gives more weight to the top of the rankings, and as a consequence it places more importance on the ties, if present. At the same time, low p minimizes the impact of the unseen part because items in the tail have low weight.

Table 2 reports the size of residuals faceted by the length of the shorter ranking. The table confirms that RES_U increases with p but decreases with s. With small and medium rankings, RES_U can easily have medium and large sizes, but with large rankings this only happens in conjunction with a high p. In contrast, RES_S decreases with p but is less affected by s. Overall, the size of RES_S is noticeable most of the cases, and even large about half the times. Similarly, Table 3 reports the size of residuals faceted by the amount of ties present in the rankings. As expected, the amount of ties does not affect RES_U, but it can have a major impact on RES_S. Indeed, with a low-to-moderate number of ties we can see a noticeable residual in *all* cases, and with a moderate-to-high number of ties most of these differences are actually larger than 0.1. Again, we can see that RES_S tends to decrease with p. Overall, we see that the total residual RES_{S+U} is noticeable in virtually all pairs of rankings in our synthetic dataset, with the majority of cases displaying a strikingly large residual.

The main message to take from the TREC and synthetic results is that *RBO* residuals may actually be too large under the right conditions. Specifically, while RES_U may become negligible for long rankings, RES_S can still become an issue. These results demonstrate the potential drawbacks of quantifying uncertainty only due to unseen items or, even worse, not quantifying uncertainty at all.

Table 2: Summary of *RBO* residuals when accounting for items in the unseen part (RES_U), ties in the seen part (RES_S), and both (RES_{S+U}). Top table for $s \leq 25$, middle for $25 < s \leq 50$ and bottom for $s > 50$. M for medium residuals in $(0.01, 0.1]$, and L for large in $(0.1, 1]$. Synthetic data.

$s \leq 25$												
	RES_U				RES_S				RES_{S+U}			
p	Avg.	Max.	M	L	Avg.	Max.	M	L	Avg.	Max.	M	L
0.80	0.01	0.06	32%	0%	0.24	1.00	27%	63%	0.25	1.00	30%	65%
0.90	0.06	0.25	80%	20%	0.18	1.00	32%	61%	0.24	1.00	19%	81%
0.95	0.20	0.50	1%	99%	0.12	1.00	46%	47%	0.32	1.00	0%	100%
$25 < s \leq 50$												
	RES_U				RES_S				RES_{S+U}			
p	Avg.	Max.	M	L	Avg.	Max.	M	L	Avg.	Max.	M	L
0.80	<.01	<.01	0%	0%	0.24	1.00	27%	62%	0.24	1.00	27%	62%
0.90	0.01	0.03	20%	0%	0.19	1.00	31%	62%	0.19	1.00	33%	64%
0.95	0.05	0.16	97%	3%	0.14	0.99	45%	50%	0.19	0.99	24%	76%
$s > 50$												
	RES_U				RES_S				RES_{S+U}			
p	Avg.	Max.	M	L	Avg.	Max.	M	L	Avg.	Max.	M	L
0.80	<.01	<.01	0%	0%	0.24	1.00	26%	62%	0.24	1.00	26%	62%
0.90	<.01	<.01	0%	0%	0.19	1.00	32%	61%	0.19	1.00	32%	62%
0.95	0.01	0.03	43%	0%	0.14	0.96	44%	50%	0.15	0.96	44%	55%

Table 3: Same as Table 2, but top table for ties ≤ 40%, middle for 40% < ties ≤ 60% and bottom for ties > 60%.

ties ≤ 40%												
	RES_U				RES_S				RES_{S+U}			
p	Avg.	Max.	M	L	Avg.	Max.	M	L	Avg.	Max.	M	L
0.80	<.01	0.06	10%	0%	0.10	0.99	39%	29%	0.10	0.99	42%	30%
0.90	0.02	0.24	33%	7%	0.07	0.93	51%	26%	0.09	0.93	52%	36%
0.95	0.09	0.47	50%	33%	0.05	0.85	66%	14%	0.14	0.85	45%	54%
40% < ties ≤ 60%												
	RES_U				RES_S				RES_{S+U}			
p	Avg.	Max.	M	L	Avg.	Max.	M	L	Avg.	Max.	M	L
0.80	<.01	0.06	10%	0%	0.21	1.00	31%	62%	0.22	1.00	31%	62%
0.90	0.02	0.25	33%	6%	0.16	1.00	36%	61%	0.19	1.00	30%	69%
0.95	0.09	0.50	50%	33%	0.12	0.99	53%	45%	0.20	0.99	23%	77%
ties > 60%												
	RES_U				RES_S				RES_{S+U}			
p	Avg.	Max.	M	L	Avg.	Max.	M	L	Avg.	Max.	M	L
0.80	<.01	0.06	10%	0%	0.35	1.00	15%	83%	0.36	1.00	15%	84%
0.90	0.02	0.23	33%	7%	0.27	1.00	15%	84%	0.29	1.00	12%	88%
0.95	0.09	0.47	50%	33%	0.20	1.00	25%	75%	0.28	1.00	9%	91%

5 Discussion

5.1 Residuals

In their original work, Webber et al. [40] acknowledged the uncertainty due to unseen items, and proposed to compute and report $RES_U = RBO_{MAX} - RBO_{MIN}$ to quantify this uncertainty. This is illustrated in Figure 1, where $RES_U = 0.128$. In their work on the treatment of ties, Corsi and Urbano [11] followed the same path and provided solutions to the computation of bounds, therefore restricting the quantification of uncertainty to just RES_U as well. This is similarly illustrated in Figure 6, where $RES_U = 0.128$ too. In contrast, in this paper we address the uncertainty due to ties in the seen part, and similarly propose to quantify it through the residual RES_S computed as the difference between the best and worst cases over all permutations of the ties. These bounds are illustrated in Figure 2, where $RES_S@6$ is already 0.374.

RES_S and RES_U may of course be large or small depending on the rankings and p. For example, RES_U may be negligible with small p or with long rankings, as is typical in IR data. In addition, and everything else being equal, longer rankings are also expected to have more ties, leading to an increase in RES_S. But even small rankings may have a high RES_S if ties appear toward the top, as they would have higher influence on the scores. It is therefore evident that one should consider uncertainty due to *both* unseen and tied items, that is, $RES_{S+U} = RBO^{hig}_{MAX} - RBO^{low}_{MIN}$. As illustrated in Figure 6, in our example $RES_{S+U} = 0.502$, which is probably just too high by any reasonable standard.

However, it is in general hard to interpret the magnitude of a residual in *RBO*. For comparison, let us consider a correlation coefficient such as Kendall's τ, and imagine that one calculates $\tau = 0.17$ with lower bound $\tau_{MIN} = -0.05$ and upper bound $\tau_{MAX} = 0.39$. Such a result would make us somewhat wary of concluding there is a positive correlation, because we can not rule out the possibility of $\tau = 0$. In fact, such bounds could even make us reconsider the possibility of no correlation at all. This kind of reasoning is possible with correlation coefficients because the value 0 is used as a reference with a well-known meaning: the expected correlation between independent rankings. However, to the best of our knowledge such reference is unknown in *RBO*, complicating the interpretation of bounds and residuals, at least in absolute terms. This issue should be addressed in future research, especially in practical settings where rankings are not infinite and probably conjoint.

5.2 Which Coefficients Should be Computed?

To deal with unseen items, Webber et al. [40] proposed RBO_{EXT} as a point estimate of the *RBO* score with the full, infinite rankings. When ties are present, Corsi and Urbano [11] similarly provided formulations of RBO_{EXT} for the *w*-, *a*- and *b*- tie-aware variants. Specifically, when ties do represent uncertainty as to the actual order of items, the most sensible choice is probably RBO^a, as it precisely computes the average score over permutations of the ties. However, as we noted in Section 3.3, their assumptions regarding the unseen items makes their formulation not entirely correct for rankings of different lengths. In contrast, our formulation RBO^{avg}_{EXT} *is* equal to the average over permutations in all cases,[5] so we strongly recommend it when reporting a point estimate for *RBO*.

It is common practice in the literature to report a single *RBO* score, most likely RBO_{EXT}, without mention of bounds or residual (see e.g. [1, 21, 25, 29, 39]). As mentioned in Section 5.1, this is likely because the residual due to unseen items, RES_U, is usually negligible in IR data because systems typically retrieve hundreds or thousands of documents. This is evidenced in Table 1, where RES_U is essentially 0 in TREC data. However, in the presence of ties, the total residual RES_{S+U} can be large (substantial in about 14% of the cases).

[5] RBO^a_{EXT} and RBO^{avg}_{EXT} differ only from s+1 to l, so their difference is mostly negligible in TREC data (medium-sized in < 1% of cases). However, with arbitrary lengths and unevenness, differences may be larger (5% of cases in our synthetic data).

In addition, the size of RES_S and RES_{S+U} increases with the amount of ties, as shown in Table 3. In light of this, we strongly suggest to *always* compute the total residual RES_{S+U} covering seen and unseen parts, or alternatively the total bounds RBO^{hig}_{MAX} and RBO^{low}_{MIN}. Whenever the residuals are relevant, they should be reported next to the point estimates RBO^{avg}_{EXT}.

5.3 *b*-variant

When ties represent uncertainty as to the actual order of items, both RBO^a and RBO^b are suitable options [11]. They follow a stochastic approach by computing the expected overlap over permutations of the ties, but the b-variant goes a step further and corrects this overlap by the measurable overlap. In other words, the b-variant corrects by the amount of ties, and as a byproduct make $RBO^b \geq RBO^a$. This correction by the amount of ties may be desirable in some applications, but it has three drawbacks.

(1) It can be deceiving because it tends to inflate the similarity between rankings. Take for example the two rankings in Figure 6; the average RBO_{EXT} is 0.688, but the worst and best possible permutations lead to 0.555 and 0.929, respectively. It turns out that RBO^b_{EXT} is 0.788, which is quite higher than the average and probably too close to the best case.
(2) This is much more problematic when the rankings tend to tie the same items, because they are essentially ignored in the computation: there is no difference between them being fully overlapping or being fully tied. As a result, RBO^b actually computes a sort of best-case similarity. To illustrate, consider in the same Figure the extreme case of comparing Y with itself; the average RBO_{EXT} is 0.8, while $RBO^b_{EXT} = 1$, that is, a perfect score! Again, this might be desirable in certain applications that require a metric space, but it is in general deceiving: RBO_{EXT} would be 1 *only* in the best possible arrangement of ties, but it can also be as low as 0.694 in the worst case.
(3) Even worse, and again because RBO^b corrects for the amount of ties, it can be the case that it yields a score higher than is even attainable by the best arrangement of the ties. For example, when comparing ⟨a [b c d]⟩ and ⟨a e [b c d]⟩, we find $RBO^b_{EXT} = 0.869$ but $RBO^{hig}_{EXT} = 0.831$.

It is only fair to state that this behavior, in the general case, is rather unreasonable and even dangerous. As such, we strongly *discourage* the use of the b-variant, and note that exactly the same arguments apply to the b-variant of Kendall's τ [16] and Yilmaz's τ_{ap} [35]. Unfortunately, τ_b is computed by default in popular software implementations such as R, SciPy in Python, MATLAB, or Apache Commons in Java. Practitioners barely ever mention which variant they use, so it is only fair to assume that they report τ_b and may therefore suffer from the three issues described above; future research should explore the implications of this choice.

6 Conclusions and Future Work

Properly accounting for measurement error is critical for Science in general, and for decision making in particular. In the context of IR and RecSys, the similarity between rankings is very often the criterion to make decisions, and *RBO* is a very popular choice for measuring that similarity. A source of uncertainty when measuring

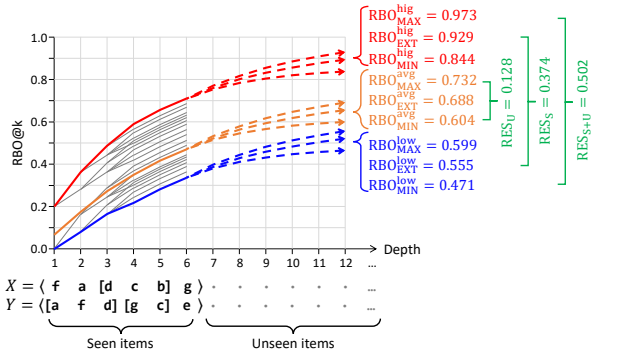

Figure 6: Uncertainty due to both unseen items and ties in the seen part (from Figure 2).

rank similarity are the unseen items ranked beyond the observed prefixes. The literature advises to report the point estimate RBO_{EXT}, quantifying uncertainty with the residual (difference between the best and worst possible arrangements of unseen items). Another source of uncertainty are the items tied in the seen parts. In this case, the literature provides tie-aware variants, most notably RBO^a, but again calculates only the residual due to unseen items, leaving open the question of how to quantify uncertainty due to ties.

In this paper we provided algorithmic solutions to this problem, allowing us to quantify a *total* residual due to both unseen and tied items. With this residual we can better grasp the potential variability in *RBO* measurements to make sensible decisions. Empirical demonstrations with TREC data showed that, while the residual due to unseen items can often be neglected because rankings are too long, the residual due to ties is substantially larger and may be just too high in many cases. With more general synthetic data, we were able to show the interplay between the length of the rankings, the amount of ties and the amount of uncertainty. In addition, we showed that the existing formulation for RBO^a deviates from the intended "average over permutations" when the rankings have different length. Although these deviations are very small in practice, we provide a reformulation, namely RBO^{avg}, that *is equal* to the "average over permutations" in all cases. Finally, we also provided arguments for the discontinuation of the b-variants of rank similarity and correlation measures.

We identify three main points for further research. First, we note that the current measurement of uncertainty through the achievable bounds of the residual may very well lead to overly conservative decisions. A better approach would be a probabilistic account for the residual, akin to confidence intervals. Second, as intuitive as our algorithmic solutions may be, we did not provide proofs of correctness. We did validate them with a brute-force approach over a dataset of 100, 000 synthetic pairs of rankings, but a formal proof or even more efficient solutions should also be explored. Third, similar work should aim for quantifying uncertainty in rank correlation coefficients in the presence of ties.

Acknowledgments

Work facilitated by computational resources of the Delft AI Cluster at TU Delft. Grazie per tutto, Rafa. Gracias por todo, Rafa.

References

[1] Aly Abdelrazek, Yomna Eid, Eman Gawish, Walaa Medhat, and Ahmed Hassan Yousef. 2022. Topic Modeling Algorithms and Applications: A Survey. *Information Systems* 112 (2022).

[2] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2017. Retrieval Consistency in the Presence of Query Variations. In *PInternational ACM SIGIR Conference on Research and Development in Information Retrieval*. 395–404.

[3] Judit Bar-Ilan, Mazlita Mat-Hassan, and Mark Levene. 2006. Methods for Comparing Rankings of Search Engine Results. *Computer Networks* 50, 10 (2006), 1448–1463.

[4] Chris Buckley. 2004. Topic Prediction Based on Comparative Retrieval Rankings. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 506–507.

[5] Guillaume Cabanac, Gilles Hubert, Mohand Boughanem, and Claude Chrisment. 2010. Tie-breaking Bias: Effect of an Uncontrolled Parameter on Information Retrieval Evaluation. In *International Conference of the Cross-Language Evaluation Forum for European Languages*. 112–123.

[6] Rocío Cañamares, Pablo Castells, and Alistair Moffat. 2020. Offline Evaluation Options for Recommender Systems. *Information Retrieval Journal* 23, 4 (2020), 387–410.

[7] Bruno Cardoso and João Magalhães. 2011. Google, Bing and a New Perspective on Ranking Similarity. *ACM International Conference on Information and Knowledge Management*, 1933–1936.

[8] Ben Carterette. 2009. On rank correlation and the distance between rankings. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 436–443.

[9] Charles L. A. Clarke, Mark D. Smucker, and Alexandra Vtyurina. 2020. Offline Evaluation by Maximum Similarity to an Ideal Ranking. In *ACM International Conference on Information and Knowledge Management*. 225–234.

[10] Gordon V Cormack and Maura R Grossman. 2018. Beyond pooling. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1169–1172.

[11] Matteo Corsi and Julián Urbano. 2024. The Treatment of Ties in Rank-Biased Overlap. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 251–260.

[12] Arthur Eddington. 1939. *The Philosophy of Physical Science*. Cambridge University Press.

[13] Ronald Fagin, Ravi Kumar, and Dakshinamurthi Sivakumar. 2003. Comparing Top k Lists. *SIAM Journal on Discrete Mathematics* 17, 1 (2003), 134–160.

[14] Soumyajit Gupta, Mücahid Kutlu, Vivek Khetan, and Matthew Lease. 2019. Correlation, Prediction and Ranking of Evaluation Metrics in Information Retrieval. In *European Conference on Information Retrieval*. 636–651.

[15] Maurice G. Kendall. 1938. A New Measure of Rank Correlation. *Biometrika* 30, 1 (1938), 81–93.

[16] Maurice G. Kendall. 1945. The Treatment of Ties in Ranking Problems. *Biometrika* 33, 3 (1945), 239–251.

[17] A. Kolmogorov. 1933. Sulla Determinazione Empirica di una Legge di Distribuzione. *Giornale dell'Istituto Italiano degli Attuari* 4 (1933), 83–91.

[18] Ravi Kumar and Sergei Vassilvitskii. 2010. Generalized Distances Between Rankings. In *International Conference on World Wide Web*. 571–580.

[19] Jimmy Lin, Tamer Elsayed, Lidan Wang, and Donald Metzler. 2009. Of Ivory and Smurfs: Loxodontan MapReduce Experiments for Web Search. In *Text REtrieval Conference TREC*, Vol. 500-278.

[20] Jimmy Lin and Peilin Yang. 2019. The Impact of Score Ties on Repeatability in Document Ranking. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1125–1128.

[21] Mykola Makhortykh, Aleksandra Urman, and Roberto Ulloa. 2020. How search engines disseminate information about COVID-19 and why they should do better. *Harvard Kennedy School Misinformation Review* 1, 3 (2020).

[22] Mika V Mantyla, Maelick Claes, and Umar Farooq. 2018. Measuring LDA topic stability from clusters of replicated runs. In *Proceedings of the 12th ACM/IEEE international symposium on empirical software engineering and measurement*. 1–4.

[23] Frank McSherry and Marc Najork. 2008. Computing Information Retrieval Performance Measures Efficiently in the Presence of Tied Scores. In *European Conference on Information Retrieval*. 414–421.

[24] Massimo Melucci. 2007. On Rank Correlation in Information Retrieval Evaluation. *ACM SIGIR Forum* 41, 1 (jun 2007), 18–33.

[25] Alistair Moffat, Peter Bailey, Falk Scholer, and Paul Thomas. 2017. Incorporating User Expectations and Behavior into the Measurement of Search Effectiveness. *ACM Transactions on Information Systems* 35, 3 (2017).

[26] Karl Pearson. 1907. *On Further Methods of Determining Correlation*. Cambridge University Press.

[27] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2018. Tranco: A research-oriented top sites ranking hardened against manipulation. *arXiv preprint arXiv:1806.01156* (2018).

[28] Vijay Raghavan, Peter Bollmann, and Gwang S. Jung. 1989. A Critical Investigation of Recall and Precision as Measures of Retrieval System Performance. *ACM Transactions on Information Systems* 7, 3 (1989), 205—-229.

[29] Chiman Salavati, Alireza Abdollahpouri, and Zhaleh Manbari. 2018. BridgeRank: A Novel Fast Centrality Measure based on Local Structure of the Network. *Physica A: Statistical Mechanics and its Applications* 496 (2018), 635–653.

[30] Avi Segal, Kobi Gal, Guy Shani, and Bracha Shapira. 2019. A difficulty ranking approach to personalization in E-learning. *International Journal of Human-Computer Studies* 130 (2019), 261–272.

[31] Charles Spearman. 1904. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology* 15, 1 (1904), 72–101.

[32] Student. 1921. An Experimental Determination of the Probable Error of Dr. Spearman's Correlation Coefficients. *Biometrika* 13, 2/3 (1921), 263–282.

[33] Mingxuan Sun, Guy Lebanon, and Kevyn Collins-Thompson. 2010. Visualizing Differences in Web Search Algorithms Using the Expected Weighted Hoeffding Distance. In *International Conference on World Wide Web*. 931–940.

[34] Luchen Tan and Charles L. A. Clarke. 2015. A Family of Rank Similarity Measures Based on Maximized Effectiveness Difference. *IEEE Transactions on Knowledge and Data Engineering. Knowl. Data Eng.* 27, 11 (2015), 2865–2877.

[35] Julián Urbano and Mónica Marrero. 2017. The Treatment of Ties in AP Correlation. In *ACM SIGIR International Conference on the Theory of Information Retrieval*. 321–324.

[36] Sebastiano Vigna. 2015. A Weighted Correlation Index for Rankings with Ties. In *International Conference on World Wide Web*. 1166–1176.

[37] Sergey Volokhin and Eugene Agichtein. 2018. Towards intent-aware contextual music recommendation: Initial experiments. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1045–1048.

[38] Sanne Vrijenhoek, Mesut Kaya, Nadia Metoui, Judith Möller, Daan Odijk, and Natali Helberger. 2021. Recommenders with a Mission: Assessing Diversity in News Recommendations. In *ACM SIGIR Conference on Human Information Interaction and Retrieval*. 173–183.

[39] Yu Wang, Yuying Zhao, Yi Zhang, and Tyler Derr. 2023. Collaboration-aware graph convolutional network for recommender systems. In *Proceedings of the ACM Web Conference 2023*. 91–101.

[40] William Webber, Alistair Moffat, and Justin Zobel. 2010. A Similarity Measure for Indefinite Rankings. *ACM Transactions on Information Systems* 28, 4 (2010), 1–38.

[41] Emine Yilmaz, Javed A. Aslam, and Stephen Robertson. 2008. A New Rank Correlation Coefficient for Information Retrieval. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 587–594.

[42] Oleg Zendel, Anna Shtok, Fiana Raiber, Oren Kurland, and J. Shane Culpepper. 2019. Information Needs, Queries, and Query Performance Prediction. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 395–404.

[43] Guido Zuccon. 2016. Understandability biased evaluation for information retrieval. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*. Springer, 280–292.