A Flexible Resource for Top-Weighted Comparisons Between Sets and Rankings

Alistair Moffat
The University of Melbourne
Melbourne, Australia
ammoffat@unimelb.edu.au

Joel Mackenzie
The University of Queensland
Brisbane, Australia
joel.mackenzie@uq.edu.au

Antonio Mallia
Pinecone Research
New York, United States
antonio@pinecone.io

Matthias Petri
Amazon AGI
Los Angeles, United States
mkp@amazon.com

Abstract

We describe `rbstar`, a toolkit of software for carrying out measurements when the goal is to determine how similar a system observation is to a gold-standard reference output. The resource covers all four combinations that arise when each of observation and reference can be either an unordered finite set in which element ordering is unimportant, or a finite prefix of an arbitrarily long ranking in which early elements are more important than later ones. Specifically, the package realizes four "rank-biased" measurement approaches that have been presented in a sequence of papers over a 15-year span, bringing them together into a single location with a uniform interface and efficient reference implementations. The provision of all of rank-biased precision, rank-biased overlap, rank-biased recall, and rank-biased alignment, with the latter two recent additions to the family, allows a wide range of measurement scenarios to be handled in a consistent manner.

CCS Concepts

• **Information systems** → **Retrieval effectiveness**; *Presentation of retrieval results*; Test collections.

Keywords

Evaluation; system comparison; set; ranking; precision; recall.

ACM Reference Format:
Alistair Moffat, Joel Mackenzie, Antonio Mallia, and Matthias Petri. 2025. A Flexible Resource for Top-Weighted Comparisons Between Sets and Rankings. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3726302.3730306

1 Introduction

Many measurement contexts involve the comparison of a *system observation* and a pre-defined gold-standard *reference output* to obtain a *numeric score*. For example, in information retrieval and web search the observation is often a ranked list of documents in decreasing score order according to some similarity heuristic, and the reference is typically a set of known-to-be-relevant documents identified via a separate judgment elicitation process. In this case the observation is a *ranking* and the reference is a *set*, that is, order is important in the former and absent from the latter; and the measurement device is a top-focused mechanism such as reciprocal rank, precision@k for some cutoff depth k, average precision [1], normalized discounted cumulative gain [4], and so on.

Moffat et al. [7] consider such measurement scenarios in detail, categorizing them into four classes: "set | set", "ranking | set", "set | ranking", and "ranking | ranking"; where "X | Y" means that an observation of type X is being measured in the context of a reference of type Y. Having established this taxonomy, Moffat et al. next discuss two existing members of the "rank-biased" family: *rank-biased precision* (RBP) [6] as a "ranking | set" measurement, and *rank-biased overlap* (RBO) [8] as a "ranking | ranking" measurement. Finally, Moffat et al. describe two new measurements: *rank-biased recall* (RBR), a "set | ranking" tool; and *rank-biased alignment* (RBA) another "ranking | ranking" facility.

The `rbstar` (as in `rb*`) software package that we describe here provides reference implementations for these four members of the rank-biased family. A particular feature of our implementations is the attention paid to tied items in rankings, noting the observations of Yang et al. [9] and Lin and Yang [5] as to the importance of handling these consistently and equitably.

2 Types and Functions

In information retrieval, system observations arise as sets or rankings of document identifiers, typically expressed as unique strings. This section first describes our chosen representations for sets and rankings, and then provides the details of four functions that compare them according to different modalities.

Representing Sets. Taking a document identifier to be a string,

docid ::= a unique sequence of alphanumerics

we define

set ::= [*pos_instances*, *neg_instances*]

where each of

pos_instances ::= [docid*]
neg_instances ::= [docid*]

SIGIR '25, July 13–18, 2025, Padua, Italy.

ACM ISBN 979-8-4007-1592-1/25/07
https://doi.org/10.1145/3726302.3730306

are lists of non-repeated document identifiers. For example,

$$s1 = \texttt{[[D03, D17, D12], [D02, D13]]}$$

is a set containing three documents, with two further documents explicitly labeled as being not members of $s1$. Either or both of *pos_instances* and *neg_instances* might be empty, and their intersection must always be empty. Both of *pos_instances* and *neg_instances* are order-agnostic, despite being represented as lists for presentational purposes, with

$$s2 = \texttt{[[D17, D12, D03], [D13, D02]]}$$

considered to be identical to set $s1$. Allowing both explicit specification of set membership and explicit set non-membership permits applications – in particular, in connection with the *qrels* relevance judgments employed in information retrieval applications – to impose different policies for items whose membership has not been determined. In particular, non-specification ("not judged at all") might be treated differently to the definite non-membership captured by presence in *neg_instances* ("judged, and determined to be not relevant").

Representing Rankings. A *ranking* is an ordered list of *groups* of equal-priority elements:

group ::= [docid*]
ranking ::= [group*]

The order of elements within each of the groups has no bearing on the ranking, but the ordering of the groups does – the first group in the ranking has the highest priority (that is, contains the most important items), and the last group has the lowest priority (that is, least important amongst those items listed), with other unlisted items potentially having even lower weights.

Groups may not contain repeated document identifiers; all pairwise intersections of groups must be empty; and the length of the ranking is the sum of the length of the groups it contains. A ranking may contain no groups, and a group may contain no items; in the latter case the effect is as if the empty group was not present at all. As an example, the ranking

$$r1 = \texttt{[[D17, D12], [D04], [D03, D13]]}$$

indicates that D17 and D12 are equally ranked in the two highest priority positions; that D04 is then the third-highest ranked document; and that D03 and D13 are equally ranked in fourth/fifth position. Elements other than these five might then follow in one or more groups of strictly lower priority, with any given ranking interpreted as being the visible prefix of an arbitrarily long sequence of groups of items. Note that order within groups is unimportant and empty groups are ignored. For example, the ranking

$$r2 = \texttt{[[D12, D17], [D04], [ ], [D03, D13]]}$$

is the same ranking as $r1$. On the other hand, changes that affect the group memberships or group ordering *are* important even if the documents are listed in the same order, meaning that

$$r3 = \texttt{[[D12, D17], [D04, D03], [D13]]}$$

is not the same ranking as $r1$ or $r2$.

We also employ obvious extensions of this notation. In particular, if B is a set and R is a ranking, then $|B|$ is the size of $B.pos\_instances$; $|R|$ is the combined size of all of the groups in R; and the expression $B \backslash R$ is the set of document identifiers that appear in $B.pos\_instances$ but not in any of the groups in R.

Computing Rank-Biased Precision. Given this terminology, rank-biased precision (RBP), originally defined by Moffat and Zobel [6], is a "ranking | set" measurement in which a ranking of documents – the observation, B – is measured relative to a reference set R of relevance judgments (*qrels*) in the context of a "patience" or "persistence" parameter $0 < \phi \le 1$ to yield a numeric score range:

$$\text{rb\_precision}(\text{ranking } B,\ \text{set } R,\ \phi) \ \rightarrow\ (score, upper)\,.$$

The value of *score* is established by occurrences in B of items in $R.pos\_instances$, weighted according to their ordinal positions in B following a geometric sequence with parameter ϕ; and *upper* is established by occurrences in B of items in $R.neg\_instances$, weighted in the same way:

$$RBP.score \leftarrow \left(\frac{1-\phi}{\phi} \sum_{e \in R.pos\_instances} \phi^{rank(B,e)}\right) \tag{1}$$

$$RBP.upper \leftarrow 1 - \left(\frac{1-\phi}{\phi} \sum_{e \in R.neg\_instances} \phi^{rank(B,e)}\right), \tag{2}$$

in which $rank(Y, e)$ yields the ordinal position in ranking Y at which document e appears, with the first position indexed at 1; and yields ∞ if the document is not present in the ranking. (Ties on ranks are discussed shortly.)

Moffat and Zobel [6] refer to the range between *score* and *upper* as the *residual*; it quantifies the extent to which the reference set R is suited to the measurement of B. They suggest that the *score* be taken as the value of the measurement (that is, erring on the side of pessimism) but that the residual also be reported, as a way of being alert to measurement uncertainty, even if (as is typically the case in IR) it is assumed that unjudged documents are non-relevant, and that documents not in *pos_instances* are members of *neg_instances*.

Note that in formulating the computation as two sums, one for *score* and one for *upper*, the bounded sum of the infinite sequence of decreasing weights associated with the elements in the unseen tail of B (the *tail residual* [6]) is automatically accounted for as part of the computed score range.

Computing Rank-Biased Recall. Moffat et al. [7] describe and motivate a closely related measurement for "set | ranking" contexts:

$$\text{rb\_recall}(\text{set } B,\ \text{ranking } R,\ \phi) \ \rightarrow\ (score, upper).$$

Their *rank-biased recall* (RBR) has a lower bound computed in an analogous manner to rank-biased precision:

$$RBR.score \leftarrow \left(\frac{1-\phi}{\phi} \sum_{e \in B.pos\_instances} \phi^{rank(R,e)}\right). \tag{3}$$

On the other hand, the range between *score* and *upper* is computed differently, with the score imprecision arising in an additive sense from not knowing where one or more of the *pos_instances* occur in the reference ranking R, and noting that the best that can happen is that they arise immediately after the visible prefix of R:

$$RBR.residual = \frac{1-\phi}{\phi} \sum_{i=1}^{|B\backslash R|} \phi^{|R|+i}\,, \tag{4}$$

$$RBR.upper = RBR.score + RBR.residual\,. \tag{5}$$

Ties in RBP and RBR. Moffat and Zobel [6] (in connection with RBP, where the observation B is a ranking) and Moffat et al. [7] (in connection with RBR, where the reference R is a ranking) propose that ties in rankings be handled by uniformly dividing the corresponding positional weights across all of the tied documents in each group of the ranking. In detail:

- The depth-based weight associated with depth d in any ranking is $w_d = (1-\phi)\phi^{d-1}$.
- A group of tied items in a ranking Y that spans depths t to b inclusive share their total depth-based weight equally. That is, counting depths as if the groups have been linearized into a single ordering, the effective weight $\hat{w}_{Y,e}$ of each item e in the equal-priority group $[t, t+1, \ldots, b] \in Y$ is:

$$\hat{w}_{Y,e} = \frac{\sum_{d=t}^{b} w_d}{b-t+1}\,.$$

- Items e that do not appear in Y have an effective weight $\hat{w}_{Y,e} = 0$.

The RBP and RBR computations then use the effective weights rather the than depth-based weights; that is, employ $\hat{w}_{B,e}$ for $e \in R.pos\_instances$ in the case of RBP (rather than $(1-\phi)\phi^{rank(B,e)-1}$ in Equation 1); and use $\hat{w}_{R,e}$ for $e \in B.pos\_instances$ in the case of RBR (rather than $(1-\phi)\phi^{rank(R,e)-1}$ in Equation 3).

For example, consider the sequence $r1$ used as an example above, and suppose that $\phi = 0.5$. The depth-based weights attached to the ranks $[1,2,3,4,5]$ are thus $[0.5, 0.25, 0.125, 0.0625, 0.03125]$ respectively. The two tied groups in $r1$ then mean that the effective weights become $[0.375, 0.375, 0.125, 0.046875, 0.046875]$, where (for example) $0.375 = (0.5+0.25)/2$. This process retains the $|r1|$-item total depth-based weight of 0.96875, and ensures that any RBP or RBR computation using $r1$ will yield the same numeric measurement as does the corresponding calculation using $r2$.

Computing Rank-Biased Alignment. Moffat et al. [7] also describe *rank-biased alignment* (RBA) as a "ranking | ranking" measurement, in which both observation B and reference R are rankings:

$$\text{rb\_alignment}(\text{ranking } B,\ \text{ranking } R,\ \phi) \ \rightarrow\ (score, upper).$$

Rank-biased alignment has much in common with RBP and RBR:

$$RBA.score = \frac{1-\phi}{\phi} \sum_{e \in B \cap R} \phi^{rank(B,e)/2 + rank(R,e)/2} \tag{6}$$

Moffat et al. [7] give pseudo-code for this computation, including computing an *upper* value that includes an assumed infinite tail of pairwise matched items beyond depth $|B \cup R|$. Our implementation mirrors that sketch.

Ties in RBA. Moffat et al. [7] do not provide guidance in regard to ties when computing RBA. Our implementation adopts the same "share the weight equally" approach as was already described in connection with RBP and RBR. The RBA lower score computation then makes use of effective item weights in B and R:

$$RBA.score = \frac{1-\phi}{\phi} \sum_{e \in B \cap R} \sqrt{\hat{w}_{B,e} \cdot \hat{w}_{R,e}}\,, \tag{7}$$

with the geometric mean in the "weights" space corresponding to the two exponent halvings in the "ϕ to the power of ranks" space employed in Equation 6. That is, an "effective rank" is computed from the corresponding effective weight of each item in the group $Y[t, t+1, \ldots, b]$ and then applied to each of the $t-b+1$ items in that group: $rank'_{Y,e} = \log_\phi(\phi \cdot w_{Y,e}/(1-\phi))$.

Initial rankings (used to compute *score*):

B = `[[D01, D23, D05], [D11], [D17, D15], [D12, D16]]`

R = `[[D01], [D11, D08], [D17], [D19, D15, D20]]`

Extended rankings (used to compute *upper*, plus tail sum):

B' = `[[D01, D23, D05], [D11], [D17, D15], [D12, D16], [D08], [D19, D20]]`

R' = `[[D01], [D11, D08], [D17], [D19, D15, D20], [D23, D05], [D12, D16]]`

Figure 1: Group-preserving extension of B and R to uniform depth $|B \cup R|$. The blue groups of unpaired items are added as the first step when computing RBA.*upper*(B, R). The tail sum $\phi^{|B \cup R|}$ is then added, since all further items might exactly match.

To compute *RBA.upper* the elements in $B \setminus R$ are assumed to be appended to R in B-priority order with any B-tied groups preserved, forming an augmented sequence R'; and the elements in $R \setminus B$ are appended to B in R-priority order with R-tied groups preserved, to form an augmented sequence B'. Figure 1 gives an example of what is meant by "priority-ordered group-preserving" appending. That augmentation allows "best possible" alignment computation through to depth $|B \cup R|$, and then a normal tail residual covers the possibility of infinite agreement thereafter. That is, *RBA.upper*(B, R) is computed as $\text{RBA}(B', R').score + \phi^{|B \cup R|}$.

Computing Rank-Biased Overlap. This measurement is another that assesses the quality of an observation ranking B relative to a reference ranking R [8]. If $Y_{1..i}$ is the first i items from a ranking Y (and is all of Y when $i \geq |Y|$), then RBO is defined as:

$$\text{rb\_overlap}(\text{ranking } B,\ \text{ranking } R,\ \phi) \ \rightarrow\ (score, upper)\,,$$

where

$$RBO.score = \frac{1-\phi}{\phi} \sum_{i=1}^{\infty} \frac{\phi^i}{i} |B_{1..i} \cap R_{1..i}|\,. \tag{8}$$

Webber et al. [8] also provide a closed form for *RBO.score*:

$$\frac{1-\phi}{\phi} \cdot \left(\sum_{i=1}^{m} \frac{\phi^i}{i} |B_{1..i} \cap R_{1..i}| + X_m \cdot \left[\ln \frac{1}{1-\phi} - \sum_{i=1}^{m} \frac{\phi^i}{i} \right] \right) \tag{9}$$

in which $m = |B \cup R|$ and $X_m = |B_{1..m} \cap R_{1..m}|$.

The upper bound for RBO is determined by extending B and R with the items in $R \setminus B$ and $B \setminus R$ respectively, as already shown in Figure 1, taking both to length $|B \cup R|$; and then further adding a tail-sum of $\phi^{|B \cup R|}$ that assumes that both sequences go to infinity with matched pairs of like items.

Ties in RBO. Webber et al. suggest that ties in the rankings B and/or R be handled by regarding all items in a group spanning depths $[t, t+1, \ldots, b]$ as occurring at depth t, adjusting other parts of the computation to match that intention. More recently, Corsi and Urbano [2, 3] presented a detailed study of ties in connection with RBO, suggesting that each tied group $[t, b]$ be regarded as requiring

```
python -m rbstar \
  --metric RBP --phi 0.80 --perquery \
  --observation ./systemX.trec \
  --reference ./trec-dl19-passage.qrels

=== Inputs ===
Observation (ranking) : ./systemX.trec
                      : 43 components
Reference   (set)     : ./trec-dl19-passage.qrels
                      : 43 components
Measurement type      : RBP (ranking | set)
Parameter phi         : 0.80

=== Per-component RBP measurements ===
component  score   resid   upper
1037798    0.1728  0.0320  0.2048
104861     0.8523  0.0183  0.8705
1063750    0.0020  0.0188  0.0208
...

=== Overall RBP measurements ===
system     cmpnts  score   resid   upper
systemX    43      0.6295  0.0160  0.6455
```

Figure 2: A sample `rbstar` execution computing per-query RBP scores ($\phi = 0.8$) for a run from the 2019 TREC Deep Learning track.

a random choice of one of the $(t - b + 1)!$ possible permutations, and that the expectation over all such possibilities be taken. This latter option is the one provided in our software.

Graded Judgments. The set type presented in this section is binary, with membership (or, conversely, non-membership) unambiguous. However in some measurement contexts set membership is on a graded scale; for example, a document might be *not relevant*, *partially relevant*, or *fully relevant* to a particular query. The `rbstar` software supports binary thresholding of qrels files on input, to allow (albeit, simplified) use of graded relevance judgments.

3 The Software Resource

We now briefly describe key elements of the `rbstar` toolkit.

Implementation and Interface. The `rbstar` suite is written in Python, and consists of approximately 1000 lines of code. It can be executed in stand-alone-mode via the command-line, or incorporated into other software as a Python module.

Outputs and Reporting. Figure 2 shows an example usage. Input options include the names of the files containing the observations and the reference data (here, a TREC run file `systemX.trec` is being scored relative to a TREC qrels file `trec-dl19-passage.qrels`); the measurement to be applied; and the parameter ϕ. The output first notes the input specification; then gives per-component measurements (assuming observation and reference are multi-component inputs, for example, over a set of topics); and then provides the arithmetic means over all observations.

Data Validation. It is important that ties in rankings be correctly recognized and consistently handled [2, 5, 9]. The `rbstar` input module validates TREC-format rankings by sorting by decreasing supplied numeric item score; then comparing that induced ordering against the supplied integer ranks. If the supplied ranks are consistent with the decreasing scores, including assigned rank values that span unbroken score ranges and/or single scores that have contiguous associated rank ranges, then the integer ranks are respected and used to form the tied groups. Or, if all of the supplied ranks are identical, tied groups are formed based on equality of score at the precision supplied in the run file; except if all ranks are equal and all scores are equal, the line ordering in the run file is used and there are no tied groups formed. If rank-score contradictions are noted, an error is raised.

Output Options. Output can be generated in text format (Figure 2); in JSON structured format for input to other downstream processing tasks; and in LaTeX `tabular` format for use in printed reports. Multiple observation files may be specified, in which case each input system will yield a row in the final output table.

4 Conclusion

We have presented a flexible public software resource – the `rbstar` toolkit – for computing a suite of top-weighted comparisons between pairs of observations and references. In particular, `rbstar` unifies all four combinations that arise when observation and reference can be either a *set* or *ranking*, bringing them together into a single software package. Written in Python, `rbstar` provides a command line tool as well as a library interface to allow easy adaptation to various tasks. We envision `rbstar` to be of interest both to information retrieval researchers and also to academics and practitioners in a wide array of other fields as well — anywhere observation sets or rankings need to be measured in a top-weighted manner relative to references that are also sets or rankings.

Acknowledgment. This work was supported by the Australian Research Council's *Discovery Projects* Scheme (project DP190101113) and a Google Research Scholar Grant.

Software. https://github.com/rankbiased/rbstar.

References

[1] C. Buckley and E. M. Voorhees. Retrieval system evaluation. In E. M. Voorhees and D. K. Harman, editors, *TREC: Experiment and Evaluation in Information Retrieval*, chapter 3, pages 53–78. MIT Press, 2005.

[2] M. Corsi and J. Urbano. The treatment of ties in rank-biased overlap. In *Proc. SIGIR*, pages 251–260, 2024.

[3] M. Corsi and J. Urbano. How do ties affect the uncertainty in rank-biased overlap? In *Proc. SIGIR-AP*, pages 125–134, 2024.

[4] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Sys.*, 20(4):422–446, 2002.

[5] J. Lin and P. Yang. The impact of score ties on repeatability in document ranking. In *Proc. SIGIR*, pages 1125–1128, 2019.

[6] A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Trans. Inf. Sys.*, 27(1):2, 2008.

[7] A. Moffat, J. Mackenzie, A. Mallia, and M. Petri. Rank-biased quality measurement for sets and rankings. In *Proc. SIGIR-AP*, pages 135–144, 2024.

[8] W. Webber, A. Moffat, and J. Zobel. A similarity measure for indefinite rankings. *ACM Trans. Inf. Sys.*, 28(4):20.1–20.38, 2010.

[9] Z. Yang, A. Moffat, and A. Turpin. How precise does document scoring need to be? In *Proc. Asia Info. Retri. Soc. Conf.*, pages 279–291, 2016.