Bootstrapped nDCG Estimation in the Presence of Unjudged Documents

Maik Fröbe,[1] Lukas Gienapp,[2] Martin Potthast,[2,3] Matthias Hagen[1]

[1] Friedrich-Schiller-Universität Jena
[2] Leipzig University
[3] ScaDS.AI

Abstract Retrieval studies often reuse TREC collections after the corresponding tracks have passed. Yet, a fair evaluation of new systems that retrieve documents outside the original judgment pool is not straightforward. Two common ways of dealing with unjudged documents are to remove them from a ranking (condensed lists), or to treat them as non- or highly relevant (naïve lower and upper bounds). However, condensed list-based measures often overestimate the effectiveness of a system, and naïve bounds are often very "loose"—especially for nDCG when some top-ranked documents are unjudged. As a new alternative, we employ bootstrapping to generate a distribution of nDCG scores by sampling judgments for the unjudged documents using run-based and/or pool-based priors. Our evaluation on four TREC collections with real and simulated cases of unjudged documents shows that bootstrapped nDCG scores yield more accurate predictions than condensed lists, and that they are able to strongly tighten upper bounds at a negligible loss of accuracy.

1 Introduction

The Cranfield experiments [12, 13] were conducted on a collection of 1,400 documents and complete relevance judgments for 225 topics. Since collection sizes grew substantially, complete judgments became infeasible almost immediately thereafter. The current best practice at shared tasks in IR is to create per-topic pools of the submitted systems' top-ranked documents and then judge each topic's pool [40]. Systems that did not contribute to the pools may then later retrieve some unjudged documents. Thakur et al. [36] recently observed this for TREC-COVID [41], where dense retrieval models in post-hoc experiments retrieved many unjudged documents that turned out to be relevant. Typical reasons for "incomplete" judgments are lacking run diversity or time constraints—which was the case for TREC-COVID as per Roberts et al. [29]. When reusing shared task data, one thus often has to deal with unjudged documents.

Unjudged documents can be judged post hoc, but this can be costly and inconsistent with the original judging process. Typically, post-hoc evaluations either remove unjudged documents (condensing the results lists of a new system to the included judged documents in their relative order) [31], or the unjudged documents are assumed to either all being non- or highly relevant (naïve lower/upper

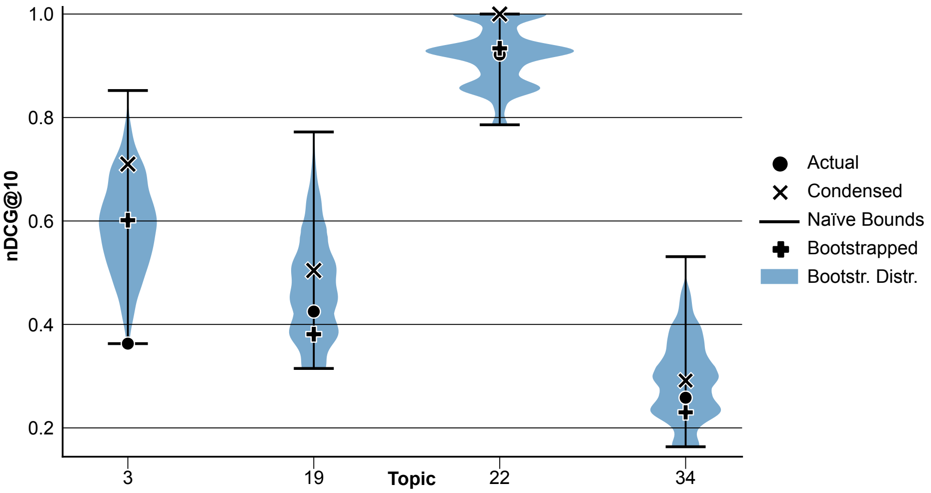

Figure 1. Actual (obtained via post-judging) and estimated nDCG@10 of the dense retrieval model ANCE for selected TREC-COVID topics with unjudged documents.

bounds) [25]. Both ideas have drawbacks: Condensed lists often overestimate effectiveness [33], and the difference between naïve lower and upper bounds can be very large [25]—especially for a recall-oriented measure such as nDCG [23], one of the most reported measures for many retrieval tasks [11, 15, 17, 36]. We further show that lower/upper bounds on nDCG are potentially incomparable to results reported based on complete judgments on the same data (Section 3.3).

To address the outlined problems, we propose a new bootstrapping approach to estimate nDCG in the presence of unjudged documents (Section 3). By repeatedly sampling judgments for unjudged documents using run- and/or pool-based priors, we derive a distribution of possible nDCG scores for a retrieval system on a topic. Figure 1 compares such distributions with the estimates of condensed lists and the naïve lower/upper bounds on selected TREC-COVID topics for the dense retrieval model ANCE [43] (which retrieved many unjudged documents deemed relevant [36]). The distributions help to identify topics with an extremely unlikely naïve upper bound (Topics 3, 19, 34), or where only a few nDCG scores between the bounds are very likely (Topic 22). In an evaluation on the Robust04, ClueWeb09, ClueWeb12, and TREC-COVID collections with real and simulated unjudged documents, we show the mode of the bootstrapped nDCG score distribution to be a more accurate estimate than those obtained from condensed lists and the, often default, naïve lower bound (Section 4). Moreover, bootstrapped nDCG bounds can be configured to be a lot tighter than the naïve upper bound at a negligible loss of accuracy. For future nDCG evaluations with unjudged documents, we share our data and code compatible with TrecTools [28].[4]

[4] https://github.com/webis-de/ECIR-23

2 Background and Related Work

We briefly review the nDCG evaluation measure, methods for dealing with unjudged documents, and previous applications of bootstrapping in IR.

Normalized Discounted Cumulative Gain (nDCG). The nDCG [23] is one of the most widely used IR evaluation measures (e.g., in the TREC Web and Deep Learning tracks [8, 17] or in the BEIR benchmark [36]). It is a normalized version of the discounted cumulative gain (DCG) that combines result ranks and graded relevance so that lower-ranked results contribute less "gain". The DCG is usually defined as

$$\text{DCG@}k = \sum_{i=1}^{k} \frac{2^{rel(d_i,q)} - 1}{\log_2(1+i)} \,,$$

where k is the maximum rank to consider, $rel(d_i, q)$ is the graded relevance judgment of the document returned at rank i for the query q, the logarithm ensures smooth reduction, and $2^{rel(d_i,q)}$ emphasizes highly relevant documents. The nDCG@k normalizes a system's DCG@k score by dividing by the DCG*@k score of the "ideal" top-k ranking of the pool (i.e., the ranking of the judged documents by relevance). Note that the ideal ranking may easily include documents that some systems do not return in their results.

Methods to Deal with Unjudged Documents. Only a few "specialized" retrieval effectiveness measures specifically target situations with unjudged documents (e.g., bpref [4] or RBP [27]). Yet, these measures are used in only a few scenarios like the TREC 2009 Web track [8] that aimed for minimal judgment pools [6]. Most retrieval studies instead usually report measures that assume all documents in the evaluated part of a ranking to have relevance judgments (e.g., nDCG). When evaluating a new retrieval system in the scenario of such a study, retrieved documents that were not in the original judgment pool cause problems [4, 46].

Typical methods [25] to deal with unjudged documents are: (1) assuming non-relevance, (2) predicting relevance, (3) condensing result lists, or (4) computing naïve bounds. Assuming non-relevance for unjudged documents is the standard in trec_eval, but only yields good results for "essentially" complete judgments [42] and favors systems that retrieve many (relevant) judged documents [35]. Since systems that retrieve unjudged but relevant documents might be severely underestimated [46], there have been attempts to automatically predict relevance [1, 2, 5, 7] (e.g., based on document content). However, such predictions can be problematic given that even experienced human assessors can struggle [38]. Also inferred measures like infAP [44] and infNDCG [45] could be viewed as prediction approaches. They exploit the probabilities with which documents were sampled for incomplete judgment pools with reduced overall effort [39]. But inference does not really work for post-hoc evaluation of systems that did not contribute to the original pool sampling since the sampling probabilities for newly retrieved high-ranked documents then can be undefined. Still, the general idea of sampling inspired our approach.

In the condensed list approach, all unjudged documents are removed from a ranked list before calculating effectiveness. The conceptual simplicity and the experimental evidence [31] that condensed lists give better results than the specially designed bpref helped condensed lists to become widely used—also in TrecTools [28] or PyTerrier [26]. But like relevance prediction, compressed lists also have the disadvantage of hiding the potential uncertainty created by unjudged documents. This motivates approaches that make this uncertainty "visible," such as calculating (naïve) lower or upper effectiveness bounds [25, 27].

Naïve bounds contrast the worst case with the best case by calculating the score a system would achieve if all unjudged documents were non-relevant or highly relevant. In the context of utility-based (based only on ranking) and recall-based (normalized by a "best possible" ranking) evaluation measures, the naïve bounds are designed for the former [25]. For utility-based measures, any actual effectiveness score of a system is guaranteed to be within the naïve bounds. However, for recall-oriented measures like nDCG, we show that the actual effectiveness of a system may lie outside the naïve bounds (cf. Section 3.3) and that expanding them often leads to meaningless 0.0 (lower) and 1.0 (upper) bounds.

Our new bootstrapping approach addresses the outlined shortcomings of the existing ideas for dealing with unjudged documents when using nDCG. By deriving a distribution of possible nDCG scores, we allow tighter bounds and more informed point estimates. Both improvements are based on the same underlying distribution of possible nDCG values, which also simplifies uncertainty assessment and interpretation.

Bootstrapping in Information Retrieval. Bootstrapping is a statistical technique in which repeated samples are drawn from data to obtain a distribution for subsequent statistical analyses [18]. It has been applied to various statistical problems in information retrieval, either as topic bootstrapping or corpus bootstrapping. Topic bootstrapping was probably the first use of bootstrapping in IR [34]. It refers to the repeated sampling of queries for some statistical analyses and has been used in significance tests [34, 35] or to assess the discriminatory power of effectiveness measures [30, 32, 47]. However, topic bootstrapping is not intended to assess the uncertainty created by unjudged documents.

In corpus bootstrapping, documents are sampled from a corpus to simulate different corpora [47]. Previous use cases of corpus bootstrapping include assessing the transferability of system comparisons between different corpora [16] or the robustness of evaluation measures [47] and significance tests [19]. The assumption underlying corpus bootstrapping is that observations should be stable between (slightly) different corpora. This inspired our idea of applying bootstrapping to evaluations with unjudged documents in the sense that an unjudged document should "behave" similarly to the judged documents in a run and/or pool. Bootstrapping has not yet been applied to the evaluation of unjudged documents, although the research reviewed above shows that bootstrapping enables similar applications. By making our code publicly available, we try to support Sakai's call for bootstrapping to get more attention in IR [30].

3 Bootstrapping nDCG Scores

After preparatory theoretical considerations, we propose a bootstrapping approach to generate nDCG score distributions by repeatedly sampling judgments for unjudged documents. Based on the lessons learned, we then reconsider current methods for estimating lower and upper bounds and propose improvements.

3.1 Preparatory Theoretical Considerations

As briefly discussed in Section 2, nDCG requires judgments to be complete up to the desired scoring depth k. Unjudged documents in the top-k results of a system must therefore either be post-judged, or be estimated otherwise based on some strategy. Post-judgments are costly and may lead to inconsistencies with prior judgments. This often leaves automatically estimating unjudged documents as the most feasible practical option.

A first idea could be to simply randomly sample relevance labels for unjudged documents. But without any further corrections, this approach can lead to invalid results. For instance, consider an evaluation setting with three relevance grades $\{0; 1; 2\}$ and a fictional judgment pool that contains nine highly relevant documents (grade 2), one relevant document (grade 1), and arbitrarily many non-relevant documents (grade 0) for some topic. Assume that a to-be-evaluated system A returns in its top-10 results the nine highly relevant documents from the pool and one unjudged document not part of the pool. Suppose that relevance grade 2 is randomly sampled for the unjudged document. Adding this sampled highly relevant document to the pool then improves the ideal ranking:

$$\mathrm{DCG}^*_{\text{original pool}}@10 \quad < \quad \mathrm{DCG}^*_{\text{pool with sample}}@10\,.$$

If $\mathrm{DCG}^*_{\text{pool with sample}}@10$ is used as the normalization denominator for computing the nDCG@10 of system A, the resulting scores are thus not directly comparable to nDCG scores of other systems calculated based on complete judgments for the original pool and $\mathrm{DCG}^*_{\text{original pool}}@10$. Comparability could be reestablished by recalculating the nDCG scores of the other systems using $\mathrm{DCG}^*_{\text{pool with sample}}@10$. Yet, recalculating scores might be biased towards the newly added system: in case the randomly sampled score is higher than the unjudged document's true relevance, recomputing diminishes the original systems' nDCG scores below their true value, yet increases the newly added systems' nDCG beyond its true value.

Conversely, also using $\mathrm{DCG}^*_{\text{original pool}}@10$ as the denominator to maintain comparability is not valid. In the example case of system A, this would cause

$$\mathrm{DCG}_{\text{system }A}@10 \quad > \quad \mathrm{DCG}^*_{\text{original pool}}@10 \quad \rightsquigarrow \quad \frac{\mathrm{DCG}_{\text{system }A}@10}{\mathrm{DCG}^*_{\text{original pool}}@10} \quad > \quad 1\,,$$

which exceeds the range of nDCG expected from normalization.

It follows that theoretically sound and empirically viable nDCG estimation approaches to handle unjudged documents *must not* change the pool's initial number of judgments per relevance grade in order to preserve the $\mathrm{DCG}^*@k$.

Algorithm 1 **Bootstrapping nDCG Scores**

```
Input:   $R$       top-$k$ ranking for query $q$ that contains unjudged documents
         $J$       pool of pairs $(d, rel(d,q))$ (i.e., documents with relevance judgments)
         $b$       number of desired bootstrapped nDCG scores
         $prior$   pool-, run-, or pool+run-based sampling probability
Output:  $Scores$  multiset of $b$ bootstrapped nDCG scores for $R$ based on $J$ and $prior$

$Scores \leftarrow \emptyset$
repeat $b$-times  ▷ following Sakai [30], we usually set $b = 1,000$
    $J' \leftarrow J, \quad S' \leftarrow \emptyset$  ▷ buffers for pool and judgm. sample of unjudg. documents
    for all unjudged documents $d \in R$ do  ▷ try to sample $prior$-based judgment
        select target relevance label $r$ for $d$ based on $prior$
        if $J'$ contains a document $d' \notin R$ with $rel(d', q) = r$ then
            $J' = J' \setminus \{(d', r)\}$
            $S' = S' \cup \{(d, r)\}$  ▷ desired judgment can be sampled from pool
        else if $J'$ contains a document $d' \notin R$ with $0 \leq rel(d', q) < r$ then
            let $d' \notin R$ be a document in $J'$ with highest $rel(d', q) < r$
            $J' = J' \setminus \{(d', rel(d', q))\}$
            $S' = S' \cup \{(d, rel(d', q))\}$  ▷ otherwise, sample best possible lower judgm.
        else
            $S' = S' \cup \{(d, 0)\}$  ▷ fallback: standard assumption of non-relevance
    $Scores \leftarrow Scores \cup \left\{ \frac{\text{DCG@}k \text{ of } R \text{ based on } J' \cup S'}{\text{DCG}^*\text{@}k \text{ of } J} \right\}$
```

3.2 Our Bootstrapped nDCG Estimation Approach

Algorithm 1 shows our approach. It meets the constraint of preserving the number of judgments per relevance grade in the pool by restricting the random sampling of relevance degrees to a *prior*. In each of the b bootstrap iterations, a relevance grade r is sampled for an unjudged document in the top-k ranking R from the judgment pool J according to one of three sampling *prior*s:

$$\begin{aligned}
&\text{pool-based} && P(rel = r \mid J) && = && \frac{|\{d \in J \ : \ rel(d,q) = r\}|}{|J|}, \\
&\text{run-based} && P(rel = r \mid R) && = && \frac{|\{d \in R \ : \ rel(d,q) = r\}|}{|\{d \in R \ : \ d \text{ is judged}\}|}, \text{ and} \\
&\text{pool+run-based} && P(rel = r \mid J, R) && = && \frac{P(rel = r \mid J) \ + \ P(rel = r \mid R)}{2}.
\end{aligned}$$

During sampling, our approach checks in each iteration whether the desired relevance grade r is still present in the pool. If not, the highest possible judgment that is below the desired grade is selected, with grade 0 as the default fallback option. This sampling strategy guarantees that the ideal ranking of the original pool J and the ideal ranking of the final "sampled" judgments $J' \cup S'$ have the same DCG*@k. The bootstrapped nDCG scores for R are thus directly comparable to nDCG scores of other rankings derived from the same pool J (e.g., to completely judged runs with nDCG scores computed on the initial pool).

Table 1. Examples with incorrect RBP-inspired / naïve nDCG@2 bounds or with very broad guaranteed nDCG bounds; relevance labels from 0 (not rel.) to 3 (highly rel.).

Bound	**Input**	**Truth**	**Estimated nDCG Bounds vs. Actual Score**				
	? = unjudged		Lower Bound	$\leq$	Actual	$\leq$	Upper Bound
RBP-insp.	$[1, \mathbf{?}]$	$[1, \mathbf{2}]$	$\frac{DCG([1,\mathbf{0}])}{DCG([1,\mathbf{0}])} = 1.00$	$\not\leq$	$\frac{DCG([1,\mathbf{2}])}{DCG([\mathbf{2},1])} = 0.80$	$\not\leq$	$\frac{DCG([1,\mathbf{3}])}{DCG([\mathbf{3},1])} = 0.71$
Naïve	$[\mathbf{?}, 1]$	$[\mathbf{2}, 1]$	$\frac{DCG([\mathbf{0},1])}{DCG([1,\mathbf{0}])} = 0.63$	$\leq$	$\frac{DCG([2,1])}{DCG([2,1])} = 1.00$	$\not\leq$	$\frac{DCG([\mathbf{0},1])}{DCG([1,\mathbf{0}])} = 0.63$
Guarant.	$[1, \mathbf{?}]$	$[1, \mathbf{2}]$	$\frac{DCG([1,\mathbf{0}])}{DCG([\mathbf{3},\mathbf{3}])} = 0.09$	$\leq$	$\frac{DCG([1,\mathbf{2}])}{DCG([\mathbf{2},1])} = 0.80$	$\leq$	$\frac{DCG([1,\mathbf{0}])}{DCG([1,\mathbf{0}])} = 1.00$

Efficient Implementation. Our bootstrapping approach computes nDCG scores in each iteration. To ensure efficiency, we precompute and tabulate the possible discounted gain values for each relevance grade at each of the top-k ranks, the DCG*@k of the ideal ranking of the given pool J, and the sum of the discounted gain values of the judged documents in R—all of these values do not change during bootstrapping. The nDCG score computation can then look up the sampled discounted gain values for unjudged documents, add them to the precomputed intermediate DCG of the judged part of R, and divide by the precomputed DCG*@k of J. On an AMD Epyc 1.8 GHz CPU, a TrecTools-based tabulated implementation of our approach takes an average of 2.84 seconds per topic (stddev: 0.01 seconds) to bootstrap nDCG@10 scores for the four runs that have the most unjudged documents in TREC-COVID (9–32% unjudged documents) as per Thakur et al. [36]—without tabulation: 17.62 seconds (stddev: 0.91 seconds). The fast run time shows that bootstrapping is practically applicable, especially since further massive parallelization is possible.

3.3 Conceptual Comparison

Our preparatory considerations from Section 3.1 also apply to the derivation of lower/upper bounds for nDCG. Bounds for nDCG inspired by RBP [25, 27] can be incomparable, too. Naïve bounds can easily be made comparable but we show that they and RBP-inspired bounds are not guaranteed to be correct. We thus devise guaranteed bounds, but show that they then "necessarily" are very broad.

Error Bounds for nDCG. Inspired by the error bounds proposed for the utility-based measure RBP [25, 27], lower/upper bounds for nDCG may be derived by either assigning a relevance grade of 0 or the highest relevance grade to all unjudged documents. But since the latter changes the ideal ranking, such an upper bound can lead to incomparable nDCG scores. Therefore, in order to yield comparable scores, we propose that an RBP-inspired "naïve" upper bound for nDCG should iteratively greedily assign the highest still available relevance judgment from the pool to the highest ranked unjudged document. If the pool's available non-zero grades are exhausted, 0 is assigned. This naïve bounding does not change the DCG*@k and thus yields scores comparable to other rankings on

Table 2. Characteristics of methods to deal with unjudged documents in nDCG scoring. Some are deterministic, some not (Det.), and they use different strategies with pool- and/or run-based priors. All are “comparable” (i.e., do not change the ideal DCG*@k).

Approach	**Det.**	**Selection/Sampling Strategy**	**Prior**		**Comp.**
			Run	Pool	
Condensed lists [31]	✓	Remove unjudged documents.	✓	✗	✓
Naïve low. b. [25, 27]	✓	Unj. = Non-relevant.	✗	✗	✓
Naïve upper bound	✓	Unj. = Highest remaining judgm.	✗	✓	✓
Pool-based bootstr.	✗	$P(rel = r \mid J) = \frac{\lvert\{d \in J : rel(d,q)=r\}\rvert}{\lvert J\rvert}$	✗	✓	✓
Run-based bootstr.	✗	$P(rel = r \mid R) = \frac{\lvert\{d \in R : rel(d,q)=r\}\rvert}{\lvert\{d \in R : d \text{ is judged}\}\rvert}$	✓	✗	✓
Pool+run-based bs.	✗	$P(rel = r \mid J, R) = \frac{P(r \mid J) + P(r \mid R)}{2}$	✓	✓	✓

the pool. However, the examples in Table 1 show that both the RBP-inspired and the naïve bounds can be incorrect. The RBP-inspired lower bound (and thus also the equivalent naïve lower bound) can be be too high (first row; the actual grade of 2 for the unjudged document increases DCG*@k more than DCG@k). Similarly, also the upper RBP-inspired and naïve bounds can be incorrect (first and second row). For a guaranteed correct lower bound, a hypothetical ideal ranking needs to be assumed that consists of only documents with the highest relevance grade, and all unjudged documents get a grade of 0. Computing a guaranteed correct upper bound is more complicated but in the end usually uses a different ideal ranking which makes the guaranteed bounds incomparable.

Discussion. Table 2 summarizes characteristics of methods that deal with unjudged documents but that preserve the ideal ranking. The methods rely on different priors (none, pool-, run-, or pool+run-based)—some only implicitly, like the upper bound method, which uses the pools highest remaining judgments. Our bootstrapping idea incorporates priors from both run and pool, and indicates the uncertainty introduced by unjudged documents through a probability distribution. Condensed lists and naïve bounds only generate point scores.

4 Evaluation

We experimentally compare our bootstrapping approach to naïve bounds and condensed lists on real and simulated scenarios with unjudged documents on the Robust04, ClueWeb09, ClueWeb12, and TREC-COVID collections. In the comparison, we assess the ability to predict actual nDCG scores, their effects on system rankings, and the tightness of potential bounds. For score prediction and the creation of subsequent system rankings, our approach uses the most likely nDCG score from the bootstrapped distribution, for tighter bounds, our approach uses fixed percentiles in the bootstrapped distribution. All experiments use nDCG@10, since it is predominant in shared tasks and the highest cut-off for which the four collections have complete judgments for the submitted runs.

Table 3. The prevalence of each relevance label in the judgment pool and the unjudged documents, respectively. For Robust04, ClueWeb09, and ClueWeb12, we show the simulated incompleteness averaged over groups; TREC-COVID is real incompleteness.

Corpus	**Judgement Pool**					**Unjudged Documents**				
	0	1	2	3	4	0	1	2	3	4
ClueWeb09	0.74	0.17	0.07	0.01	0.01	0.80	0.15	0.03	0.01	0.01
ClueWeb12	0.64	0.25	0.09	0.02	0.02	0.67	0.24	0.08	0.02	0.01
Robust04	0.80	0.18	0.02	0.00	0.00	0.96	0.04	0.00	0.00	0.00
TREC-COVID	0.63	0.16	0.21	0.00	0.00	0.75	0.02	0.23	0.00	0.00

4.1 Experimental Setup

We compare a run with unjudged documents in two setups against (1) runs without unjudged documents (measuring the accuracy of lower and upper bounds), and (2) other runs without unjudged documents (measuring correlations in system rankings). Score ties in a run are solved via alphanumeric ordering by document ID (following a recommendation by Lin and Yang [24]). To reduce the impact of low-performing systems, only the 75% of runs with the highest nDCG@10 are included (following a similar setup by Bernstein and Zobel [3]). The ClueWeb corpora have a high number of near-duplicates [20] that might invalidate subsequent evaluations [3, 21, 22]. We use pre-calculated lists [20] to deduplicate the run and qrel files. We follow trec_eval and replace negative relevance judgments with 0. All experiments use TrecTool's nDCG@10 implementation with default parameters, and we report statistical significance where applicable according to the Students' t-test with Bonferroni correction at $p = 0.05$.

Test Collections. Our evaluation is based on four collections: (1) Robust04 [37] (528,155 documents, 249 topics, 311,410 relevance judgments, pool: 111 runs by 14 groups), (2) ClueWeb09 (1 billion web pages, 200 topics, 58,414 judgments from TREC Web tracks [8, 9, 10, 11], pools: 32–71 runs by 12–23 groups), (3) ClueWeb12 (0.7 billion web pages, 100 topics, 23,233 judgments from TREC Web tracks [14, 15], pools: 34 + 30 runs by 14 + 12 groups), (4) TREC-COVID [41] (171,332 documents, 50 topics, 66,336 judgments).

Establishing Incompleteness. TREC-COVID allows a real case study on incompleteness. In post-hoc experiments [36], three models retrieved 17% to 41% unjudged documents in their top-10 that were post-judged [36]. For Robust04, ClueWeb09, and ClueWeb12, we simulate incomplete pools with the "leave one group out" method [38], adjusting the pool by removing documents solely contributed by the group submitting a run (i.e., only their runs have the document in the top-10 results), simulating that the group did not participate. This yields one incomplete pool per group, where runs of other groups remain fully judged.

Table 3 provides an overview of the ratios of relevance degrees in the pools and the unjudged documents. For simulated incompleteness, we report averages over all groups. None of the collections are complete, as all have relevant documents

Table 4. Overview of nDCG score prediction assessed by the actual RMSE, and the lower and upper bound RMSE (ignoring under/overestimations) on Robust04 (R04), ClueWeb09 (CW09), and ClueWeb12 (CW12). We report statistical significance according to Student's t-test with Bonferroni correction at p=0.05 to the naïve lower (†) and upper bound (‡), respectively condensed lists (∗).

Approach	**RMSE on R04**			**RMSE on CW09**			**RMSE on CW12**		
	Lower	Actual	Upper	Lower	Actual	Upper	Lower	Actual	Upper
Naïve (L)	**.004***‡	.058*‡	.058*‡	**.009***‡	.076*‡	.076*‡	**.007***‡	.113*‡	.113*‡
Conden.	.062†‡	.068†‡	.027†‡	.081†‡	.087†‡	.034†‡	.081†‡	.092†‡	.043†‡
Naïve (U.)	.210†*	.210†*	**.002**†*	.338†*	.338†*	**.000**†*	.307†*	.307†*	**.001**†*
Bootstr.$_P$	.078†*‡	.083†*‡	.027†‡	.086†‡	.097†*‡	.046†*‡	.093†*‡	.105*‡	.048†‡
Bootstr.$_R$	.007†*‡	.058*‡	.058*‡	.021†*‡	.077*‡	.075*‡	.059†*‡	.108*‡	.091†*‡
Bootstr.$_{P+R}$	.037†*‡	**.056***‡	.041†*‡	.046†*‡	**.074***‡	.058†*‡	.058†*‡	**.083**†‡	.060†*‡

among the unjudged ones. However, for Robust04, the high number of submitted runs and deep pooling ensured that the pools are "essentially complete", even for simulated incompleteness (4% of the unjudged documents are relevant). The remaining collections have 20% to 33% relevant documents among the unjudged ones, providing a good range of (in)completeness for our experiments.

4.2 Evaluation Results

For nDCG prediction experiments, accuracy is reported as root-mean-square error (RMSE), contrasted by two RMSE variants that assess lower and upper bounds. Furthermore, we measure the correlation of system rankings obtained by predicted nDCG scores to the ground truth rankings as Kendall's τ and Spearman's ρ. For experiments on tightening naïve bounds, we measure precision and recall in reconstructing per-topic system rankings. Evaluation is first conducted on simulated incompleteness and concludes with the TREC-COVID case study.

nDCG Score Predicion. Table 4 reports the nDCG@10 prediction accuracy of all tested approaches. We report the actual RMSE, a lower-bound RMSE (ignoring underestimations), and an upper-bound RMSE (ignoring overestimations). Cases with incorrect naïve bounds occur in practice but are rare. The naïve lower bound is slightly more inaccurate than the naïve upper bound (maximum violations of 0.009 on ClueWeb09 for the lower bound vs. 0.002 for the upper bound on Robust04). Similar to the incompleteness degrees of the collections (Table 3), the actual RMSE is rather small on Robust04, larger on ClueWeb09, and the highest on ClueWeb12. Consequently, the naïve lower bound that assumes unjudged documents are non-relevant has high accuracy on both collections, but is outperformed by condensed lists on ClueWeb12 (RMSE 0.113 vs. 0.92).

Our three bootstrapping variants with a prior from the pool (Bootstr.$_P$), the run (Bootstr.$_R$), or both (Bootstr.$_{P+R}$) show that priors from the run yield more accurate results than from the pool, and combining both yields the highest accuracy in all cases, significantly improving upon the naïve lower and upper bound,

Table 5. Overview of the correlation between system rankings obtained via predicted nDCG@10 scores on incompletely judged runs to those runs with complete judgments. We report Kendall's τ and Spearman's ρ on Robust04, ClueWeb09, ClueWeb12, and the mean over those three corpora.

Approach	**Robust04**		**ClueWeb09**		**ClueWeb12**		**Mean**	
	τ	ρ	τ	ρ	τ	ρ	τ	ρ
Naïve (L)	.936	**.997**	**.821**	**.959**	.646	.837	.801	.931
Conden.	.924	.978	.610	.744	.786	.889	.773	.870
Naïve (U.)	.189	-.268	-.411	-.656	-.097	-.250	-.106	-.391
Bootstr.$_{P}$	.911	.975	.644	.824	.781	.909	.779	.903
Bootstr.$_{R}$	.943	**.997**	.721	.878	.764	.908	.810	.927
Bootstr.$_{P+R}$	**.966**	.996	.716	.885	**.814**	**.924**	**.832**	**.935**

and condensed lists. This result is reasonable, as the combination of run priors and pool priors allows the bootstrapping approach to account for relationships between the topic and the run. The results show that bootstrapped nDCG scores from run and pool priors are highly applicable in practice as they yield the most accurate nDCG predictions in all our experiments. Additionally, by comparing the lower- and upper-bound RMSE of condensed lists with those of pool/run-based bootstrapping, we observe that condensed lists are inclined to overestimate on all corpora. In contrast, bootstrapped predictions are more balanced with a tendency for underestimations, which is preferable in practice [35].

System Ranking Reconstruction Against Incompletely Judged Runs. We contrast our experiments on the accuracy of predicted nDCG@10 scores by measuring the correlation of system rankings obtained via predicted scores on incompletely judged runs to the ground truth system ranking obtained via fully judged runs. Therefore, we predict the nDCG@10 sores of each run using the incomplete judgments for the run obtained via the "leave one group out" method [38]. Table 5 reports the correlation of the system rankings obtained on the incomplete judgments with the ground-truth system ranking measured as Kendall's τ and Spearman's ρ. Again, we observe that the judgment pool for Robust04 is, even with simulated incompleteness, highly reusable as all approaches (besides the naïve upper bound) achieve high correlations (pool/run- based bootstrapping having the highest Kendall's τ of 0.966). Our pool/run-based bootstrapping substantially outperforms condensed lists in all cases, and also achieves the highest correlation on average over all three corpora (Kendall's τ of 0.832).

System Ranking Reconstruction Against Fully Judged Runs. To assess pool/run-based bootstrapping for tightening naïve bounds, we compare different methods for score prediction w.r.t. their ability to reconstruct the topic-level ground-truth ranking of systems. Given a run with unjudged documents, we first calculate point estimates: the naïve lower bound, condensed list, and the most likely score according to pool/run-based bootstrapping. Then, score ranges are established,

Table 6. Precision, recall, and F1 in reconstructing topic-level system rankings with unjudged documents. We report significance (Student's t-test with Bonferroni correction at p=0.05) to the point estimate of list condensation (∗) and score ranges starting at the lower bound, ending at the naïve upper bound (†), resp. list condensation (‡).

Approach		**Reconstr. on R04**			**Reconstr. on CW09**			**Reconstr. on CW12**		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Point	Naïve (L.)	$.954^{\dagger *}$	$\mathbf{.954}^{\dagger * \ddagger}$	$\mathbf{.954}^{\dagger *}$	$.921^{\dagger * \ddagger}$	$\mathbf{.921}^{\dagger * \ddagger}$	$\mathbf{.921}^{\dagger * \ddagger}$	$.866^{\dagger \ddagger}$	$.866^{\dagger \ddagger}$	$.866^{\dagger}$
	Conden.	$.931^{\dagger \ddagger}$	$.931^{\dagger \ddagger}$	$.931^{\dagger}$	$.886^{\dagger \ddagger}$	$.886^{\dagger \ddagger}$	$.886^{\dagger}$	$.891^{\dagger \ddagger}$	$.891^{\dagger \ddagger}$	$.891^{\dagger}$
	$\mathrm{BS}_{R/P}$	$.946^{\dagger * \ddagger}$	$.946^{\dagger * \ddagger}$	$.946^{\dagger *}$	$.916^{\dagger * \ddagger}$	$.916^{\dagger * \ddagger}$	$.916^{\dagger * \ddagger}$	$.903^{\dagger \ddagger}$	$\mathbf{.903}^{\dagger \ddagger}$	$\mathbf{.903}^{\dagger \ddagger}$
Range	Naïve (U.)	$\mathbf{.987}^{*}$	$.775^{* \ddagger}$	$.865^{* \ddagger}$	$\mathbf{.995}^{* \ddagger}$	$.606^{* \ddagger}$	$.741^{* \ddagger}$	$\mathbf{.998}^{* \ddagger}$	$.547^{* \ddagger}$	$.693^{* \ddagger}$
	Cond.	$.973^{*}$	$.906^{\dagger *}$	$.936^{\dagger}$	$.969^{\dagger *}$	$.833^{\dagger *}$	$.892^{\dagger}$	$.957^{\dagger *}$	$.791^{\dagger *}$	$.862^{\dagger}$
	$\mathrm{BS}_{P+R@75}$	$.977^{*}$	$.868^{\dagger * \ddagger}$	$.917^{\dagger}$	$.972^{\dagger *}$	$.822^{\dagger *}$	$.888^{\dagger}$	$.971^{\dagger *}$	$.758^{\dagger *}$	$.847^{\dagger *}$
	$\mathrm{BS}_{P+R@90}$	$.985^{*}$	$.831^{\dagger * \ddagger}$	$.898^{\dagger * \ddagger}$	$.985^{\dagger * \ddagger}$	$.766^{\dagger * \ddagger}$	$.857^{\dagger * \ddagger}$	$.986^{\dagger * \ddagger}$	$.707^{\dagger * \ddagger}$	$.817^{\dagger * \ddagger}$
	$\mathrm{BS}_{P+R@95}$	$.986^{*}$	$.815^{\dagger * \ddagger}$	$.890^{\dagger * \ddagger}$	$.988^{* \ddagger}$	$.739^{\dagger * \ddagger}$	$.840^{\dagger * \ddagger}$	$.990^{* \ddagger}$	$.673^{\dagger * \ddagger}$	$.793^{\dagger * \ddagger}$

starting at the naïve lower bound and ending at different high points: the naïve upper bound, the score of condensed lists, and the upper 75%, 90%, or 95% percentiles of the bootstrapped distributions. Score ranges and point estimates for each run are compared against the scores of all other runs that contributed to the respective pool, emitting corresponding system preferences if the range/estimate is strictly below or above the exact score of another system.

Table 6 reports the reconstruction effectiveness as precision, recall, and F1 score. In recall-oriented settings, where score ranges are unsuitable, the naïve lower bound (recall of 0.954 on Robust04), or the bootstrapped prediction (recall of 0.903 on the ClueWeb12) should be used. In precision-oriented scenarios, naïve bounds achieve the highest precision at a high cost in recall (only 0.547 on the ClueWeb12). The pool/run-based bootstrapping at the 95% percentile provides significantly tighter naïve bounds (recall is always significantly better) at a negligible loss in precision (not significant in all cases). Hence, nDCG bounds can be substantially tightened without loss in accuracy using bootstrapping.

Real Incompleteness on TREC-COVID. As a final case study, we apply naïve bounds, condensed lists, and our pool/run-based bootstrapping to estimate the nDCG@10 of three dense retrieval models on the original TREC-COVID collection, for which the unjudged documents were post-judged [36]. The three dense retrieval systems operated in a zero-shot setting. Thus we compare them against the best run submitted to the first round of TREC-COVID, as those systems also had no access to training data.

Table 7 shows the results on the original (incomplete) TREC-COVID qrels and the post-hoc (complete) qrels for three selections of topics: (1) moderate levels of incompleteness (between 25% to 50% unjudged documents), (2) high incompleteness (more than 50% unjudged documents), and (3) all topics (only nDCG@10 scores in the setup with all topics are comparable between different systems). The original run files were not stored in the BEIR experiments [36], so

Table 7. The nDCG@10 on the original qrels (unjudged documents) from TREC-COVID and the expanded qrels (all documents judged) for topics with 25% to 50% unjudged documents (.25 to .5), topics with more than 50% unjudged documents (.5 to 1), and all topics. We report the proportion of unjudged documents (U@10), and predictions of the lower bound (Default), condensed lists (Cond.), pool/run-based bootstrapping (BS_{P+R}), and naïve and tightened upper bounds ($\mathrm{BS}_{P+R@95}$).

	Model	**Original Qrels**						**Ex. Qrels**
			nDCG@10			Upper Bound		nDCG@10
		U@10	Default	Cond.	BS_{P+R}	Naïve	$\mathrm{BS}_{P+R@95}$	
.25 to .5	ANCE	35.6%	0.489 -0.161	0.683 +0.033	0.660 +0.010	0.838 +0.188	0.795 +0.145	0.650
	ColBERT	33.3%	0.485 -0.141	0.641 +0.015	0.614 -0.012	0.770 +0.144	0.741 +0.115	0.626
	TAS-B	32.5%	0.597 ±0.000	0.875 +0.278	0.847 +0.250	0.902 +0.305	0.894 +0.297	0.597
.5 to 1	ANCE	65.6%	0.207 -0.150	0.547 +0.190	0.385 +0.028	0.769 +0.412	0.542 +0.185	0.357
	ColBERT	62.9%	0.337 -0.110	0.679 +0.232	0.517 +0.070	0.881 +0.434	0.645 +0.198	0.447
	TAS-B	73.8%	0.211 -0.119	0.584 +0.254	0.459 +0.129	0.918 +0.588	0.623 +0.293	0.330
All Topics	ANCE	22.4%	0.652 -0.083	0.772 +0.037	0.747 +0.012	0.853 +0.118	0.804 +0.069	0.735
	ColBERT	17.2%	0.680 -0.054	0.770 +0.036	0.741 +0.007	0.826 +0.092	0.789 +0.055	0.734
	TAS-B	41.0%	0.481 -0.074	0.705 +0.150	0.633 +0.078	0.871 +0.316	0.729 +0.174	0.555
	1st@TREC	0.0%	0.679 ±0.000	0.679 ±0.000	0.679 ±0.000	0.679 ±0.000	0.679 ±0.000	0.679

we reproduced them (only minor differences for ANCE, TAS-B, and ColBERT, but for DPR, we scores were substantially different and still had unjudged documents, so we exclude DPR). The default behaviour of assuming that unjudged documents are non-relevant (i.e., the naïve lower bound) underestimates the effectiveness for all dense retrieval models. At the same time, condensed lists substantially overestimate the effectiveness (e.g., for TAS-B by 0.150). Our proposed pool/run-based bootstrapping produces the best estimates in all cases. Tightening upper bounds with bootstrapping is very valuable, as the 95% percentile of bootstrapped nDCG scores is much tighter as the naïve upper bound.

5 Conclusion

Our new bootstrapping method to account for unjudged documents in post-hoc nDCG evaluations is efficient in practice and more effective than previous methods that derive a point estimate or bounds for a system's true nDCG. Packaged as a TrecTools-compatible software that is publicly available, bootstrapped estimation is directly applicable to retrieval studies.

As interesting directions for future work, we want to expand our bootstrapping approach to more evaluation measures (e.g., Q-Measure, MAP, or RBP) and combine it with approaches that predict the relevance of unjudged documents based on their content. This combination could lead to more informed bootstrap priors and might also tighten the resulting bootstrapped score distributions.

Acknowledgments

This work has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101070014 (OpenWebSearch.EU, https://doi.org/10.3030/101070014).

References

[1] Aslam, J.A., Pavlu, V., Yilmaz, E.: A statistical method for system evaluation using incomplete judgments. In: Efthimiadis, E.N., Dumais, S.T., Hawking, D., Järvelin, K. (eds.) SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006, pp. 541–548, ACM (2006)

[2] Aslam, J.A., Yilmaz, E.: Inferring document relevance from incomplete information. In: Silva, M.J., Laender, A.H.F., Baeza-Yates, R.A., McGuinness, D.L., Olstad, B., Olsen, Ø.H., Falcão, A.O. (eds.) Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007, pp. 633–642, ACM (2007)

[3] Bernstein, Y., Zobel, J.: Redundant documents and search effectiveness. In: Herzog, O., Schek, H., Fuhr, N., Chowdhury, A., Teiken, W. (eds.) Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005, pp. 736–743, ACM (2005)

[4] Buckley, C., Voorhees, E.M.: Retrieval evaluation with incomplete information. In: Sanderson, M., Järvelin, K., Allan, J., Bruza, P. (eds.) SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004, pp. 25–32, ACM (2004)

[5] Büttcher, S., Clarke, C.L.A., Yeung, P.C.K., Soboroff, I.: Reliable information retrieval evaluation with incomplete and biased judgements. In: Kraaij, W., de Vries, A.P., Clarke, C.L.A., Fuhr, N., Kando, N. (eds.) SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007, pp. 63–70, ACM (2007)

[6] Carterette, B., Allan, J., Sitaraman, R.K.: Minimal test collections for retrieval evaluation. In: Efthimiadis, E.N., Dumais, S.T., Hawking, D., Järvelin, K. (eds.) SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006, pp. 268–275, ACM (2006)

[7] Carterette, B., Jones, R.: Evaluating search engines by modeling the relationship between relevance and clicks. In: Platt, J.C., Koller, D., Singer, Y., Roweis, S.T. (eds.) Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007, pp. 217–224, Curran Associates, Inc. (2007)

[8] Clarke, C.L.A., Craswell, N., Soboroff, I.: Overview of the TREC 2009 Web track. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009, Gaithersburg, Maryland, USA, November 17-20, 2009, NIST Special Publication, vol. 500-278, National Institute of Standards and Technology (NIST) (2009)

[9] Clarke, C.L.A., Craswell, N., Soboroff, I., Cormack, G.V.: Overview of the TREC 2010 Web track. In: Proceedings of The Nineteenth Text REtrieval Conference, TREC 2010, Gaithersburg, Maryland, USA, November 16-19, 2010 (2010)

[10] Clarke, C.L.A., Craswell, N., Soboroff, I., Voorhees, E.M.: Overview of the TREC 2011 Web track. In: Proceedings of The Twentieth Text REtrieval Conference, TREC 2011, Gaithersburg, Maryland, USA, November 15-18, 2011 (2011)

[11] Clarke, C.L.A., Craswell, N., Voorhees, E.M.: Overview of the TREC 2012 Web track. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of The Twenty-First Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, USA, November 6-9, 2012, NIST Special Publication, vol. 500-298, National Institute of Standards and Technology (NIST) (2012)

[12] Cleverdon, C.: The Cranfield tests on index language devices. In: ASLIB Proceedings, pp. 173–192, MCB UP Ltd. (Reprinted in Readings in Information Retrieval, Karen Sparck-Jones and Peter Willett, editors, Morgan Kaufmann, 1997) (1967)

[13] Cleverdon, C.W.: The significance of the Cranfield tests on index languages. In: Bookstein, A., Chiaramella, Y., Salton, G., Raghavan, V.V. (eds.) Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Chicago, Illinois, USA, October 13-16, 1991 (Special Issue of the SIGIR Forum), pp. 3–12, ACM (1991)

[14] Collins-Thompson, K., Bennett, P.N., Diaz, F., Clarke, C., Voorhees, E.M.: TREC 2013 Web track overview. In: Proceedings of The Twenty-Second Text REtrieval Conference, TREC 2013, Gaithersburg, Maryland, USA, November 19-22, 2013 (2013)

[15] Collins-Thompson, K., Macdonald, C., Bennett, P.N., Diaz, F., Voorhees, E.M.: TREC 2014 Web track overview. In: Proceedings of The Twenty-Third Text REtrieval Conference, TREC 2014, Gaithersburg, Maryland, USA, November 19-21, 2014 (2014)

[16] Cormack, G.V., Lynam, T.R.: Statistical precision of information retrieval evaluation. In: Efthimiadis, E.N., Dumais, S.T., Hawking, D., Järvelin, K. (eds.) SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006, pp. 533–540, ACM (2006)

[17] Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.M.: Overview of the TREC 2019 Deep Learning track. In: Voorhees, E., Ellis, A. (eds.) 28th International Text Retrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, NIST Special Publication, National Institute of Standards and Technology (NIST) (Nov 2019)

[18] Efron, B., Tibshirani, R.: An introduction to the bootstrap. CRC press (1994)

[19] Ferro, N., Sanderson, M.: How do you test a test?: A multifaceted examination of significance tests. In: Candan, K.S., Liu, H., Akoglu, L., Dong, X.L., Tang, J. (eds.) WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022, pp. 280–288, ACM (2022)

[20] Fröbe, M., Bevendorff, J., Gienapp, L., Völske, M., Stein, B., Potthast, M., Hagen, M.: CopyCat: Near-duplicates within and between the ClueWeb and the Common Crawl. In: Diaz, F., Shah, C., Suel, T., Castells, P., Jones, R., Sakai, T. (eds.) 44th International ACM Conference on Research and Development in Information Retrieval (SIGIR 2021), pp. 2398–2404, ACM (Jul 2021)

[21] Fröbe, M., Bevendorff, J., Reimer, J., Potthast, M., Hagen, M.: Sampling bias due to near-duplicates in learning to rank. In: 43rd International ACM Conference on Research and Development in Information Retrieval (SIGIR 2020), pp. 1997–2000, ACM (Jul 2020)

[22] Fröbe, M., Bittner, J., Potthast, M., Hagen, M.: The effect of content-equivalent near-duplicates on the evaluation of search engines. In: Jose, J., Yilmaz, E., Magalhães, J., Castells, P., Ferro, N., Silva, M., Martins, F. (eds.) Advances in Information Retrieval. 42nd European Conference on IR Research (ECIR 2020), Lecture Notes in Computer Science, vol. 12036, pp. 12–19, Springer, Berlin Heidelberg New York (Apr 2020)

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

[24] Lin, J., Yang, P.: The impact of score ties on repeatability in document ranking. In: Piwowarski, B., Chevalier, M., Gaussier, É., Maarek, Y., Nie, J., Scholer, F. (eds.) Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019, pp. 1125–1128, ACM (2019)

[25] Lu, X., Moffat, A., Culpepper, J.S.: The effect of pooling and evaluation depth on IR metrics. Inf. Retr. J. **19**(4), 416–445 (2016)

[26] Macdonald, C., Tonellotto, N.: Declarative experimentation in information retrieval using PyTerrier. In: Balog, K., Setty, V., Lioma, C., Liu, Y., Zhang, M., Berberich, K. (eds.) ICTIR '20: The 2020 ACM SIGIR International Conference on the Theory of Information Retrieval, Virtual Event, Norway, September 14-17, 2020, pp. 161–168, ACM (2020)

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

[28] Palotti, J.R.M., Scells, H., Zuccon, G.: TrecTools: An open-source Python library for information retrieval practitioners involved in TREC-like campaigns. In: Piwowarski, B., Chevalier, M., Gaussier, É., Maarek, Y., Nie, J., Scholer, F. (eds.) Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019, pp. 1325–1328, ACM (2019)

[29] Roberts, K., Alam, T., Bedrick, S., Demner-Fushman, D., Lo, K., Soboroff, I., Voorhees, E.M., Wang, L.L., Hersh, W.R.: TREC-COVID: Rationale and structure of an information retrieval shared task for COVID-19. J. Am. Medical Informatics Assoc. **27**(9), 1431–1436 (2020)

[30] Sakai, T.: Evaluating evaluation metrics based on the bootstrap. In: Efthimiadis, E.N., Dumais, S.T., Hawking, D., Järvelin, K. (eds.) SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006, pp. 525–532, ACM (2006)

[31] Sakai, T.: Alternatives to bpref. In: Kraaij, W., de Vries, A.P., Clarke, C.L.A., Fuhr, N., Kando, N. (eds.) SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007, pp. 71–78, ACM (2007)

[32] Sakai, T.: On the reliability of information retrieval metrics based on graded relevance. Inf. Process. Manag. **43**(2), 531–548 (2007)

[33] Sakai, T.: Comparing metrics across TREC and NTCIR: The robustness to system bias. In: Shanahan, J.G., Amer-Yahia, S., Manolescu, I., Zhang, Y.,

Evans, D.A., Kolcz, A., Choi, K., Chowdhury, A. (eds.) Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008, pp. 581–590, ACM (2008)

[34] Savoy, J.: Statistical inference in retrieval effectiveness evaluation. Inf. Process. Manag. **33**(4), 495–512 (1997)

[35] Smucker, M.D., Allan, J., Carterette, B.: A comparison of statistical significance tests for information retrieval evaluation. In: Silva, M.J., Laender, A.H.F., Baeza-Yates, R.A., McGuinness, D.L., Olstad, B., Olsen, Ø.H., Falcão, A.O. (eds.) Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007, pp. 623–632, ACM (2007)

[36] Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., Gurevych, I.: BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In: Vanschoren, J., Yeung, S. (eds.) Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual (2021)

[37] Voorhees, E.: The TREC Robust Retrieval track. SIGIR Forum **39**(1), 11–20 (2005)

[38] Voorhees, E.M.: The philosophy of information retrieval evaluation. In: Peters, C., Braschler, M., Gonzalo, J., Kluck, M. (eds.) Evaluation of Cross-Language Information Retrieval Systems, Second Workshop of the Cross-Language Evaluation Forum, CLEF 2001, Darmstadt, Germany, September 3-4, 2001, Revised Papers, Lecture Notes in Computer Science, vol. 2406, pp. 355–370, Springer (2001)

[39] Voorhees, E.M.: The effect of sampling strategy on inferred measures. In: Geva, S., Trotman, A., Bruza, P., Clarke, C.L.A., Järvelin, K. (eds.) The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014, pp. 1119–1122, ACM (2014)

[40] Voorhees, E.M.: The evolution of Cranfield. In: Ferro, N., Peters, C. (eds.) Information Retrieval Evaluation in a Changing World - Lessons Learned from 20 Years of CLEF, The Information Retrieval Series, vol. 41, pp. 45–69, Springer (2019)

[41] Voorhees, E.M., Alam, T., Bedrick, S., Demner-Fushman, D., Hersh, W.R., Lo, K., Roberts, K., Soboroff, I., Wang, L.L.: TREC-COVID: Constructing a pandemic information retrieval test collection. SIGIR Forum **54**(1), 1:1–1:12 (2020)

[42] Voorhees, E.M., Soboroff, I., Lin, J.: Can old TREC collections reliably evaluate modern neural retrieval models? CoRR **abs/2201.11086** (2022)

[43] Xiong, L., Xiong, C., Li, Y., Tang, K., Liu, J., Bennett, P.N., Ahmed, J., Overwijk, A.: Approximate nearest neighbor negative contrastive learning for dense text retrieval. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net (2021)

[44] Yilmaz, E., Aslam, J.A.: Estimating average precision with incomplete and imperfect judgments. In: Yu, P.S., Tsotras, V.J., Fox, E.A., Liu, B. (eds.) Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management, Arlington, Virginia, USA, November 6-11, 2006, pp. 102–111, ACM (2006)

[45] Yilmaz, E., Kanoulas, E., Aslam, J.A.: A simple and efficient sampling method for estimating AP and NDCG. In: Myaeng, S., Oard, D.W., Sebastiani, F., Chua, T., Leong, M. (eds.) Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008, pp. 603–610, ACM (2008)

[46] Zobel, J.: How reliable are the results of large-scale information retrieval experiments? In: Croft, W.B., Moffat, A., van Rijsbergen, C.J., Wilkinson, R., Zobel, J. (eds.) SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia, pp. 307–314, ACM (1998)

[47] Zobel, J., Rashidi, L.: Corpus bootstrapping for assessment of the properties of effectiveness measures. In: d'Aquin, M., Dietze, S., Hauff, C., Curry, E., Cudré-Mauroux, P. (eds.) CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020, pp. 1933–1952, ACM (2020)