INST: An Adaptive Metric for Information Retrieval Evaluation

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

Peter Bailey
Microsoft, Australia
pbailey@microsoft.com

Falk Scholer
RMIT University, Australia
falk.scholer@rmit.edu.au

Paul Thomas
CSIRO, Australia
paul.thomas@csiro.au

ABSTRACT

A large number of metrics have been proposed to measure the effectiveness of information retrieval systems. Here we provide a detailed explanation of one recent proposal, INST, articulate the various properties that it embodies, and describe a number of pragmatic issues that need to be taken in to account when writing an implementation. The result is a specification for a program `inst_eval` for use in TREC-style IR experimentation.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*performance evaluation*.

Keywords

User behavior, test collections, relevance measures

1. INTRODUCTION

Effectiveness metrics are an essential element of IR research. By crystallizing a ranked list of documents into a single score, they allow retrieval systems to be measured and compared, and for system improvements to be monitored and assured. Many effectiveness metrics have been proposed, and each embodies, either explicitly or implicitly, a number of assumptions about what makes an IR system "good"; since such systems are typically employed by users seeking to satisfy an information need, considerations typically include how a user interacts with a search system, and how they accrue benefit from such an interaction [2]. Widely used metrics include average precision [7, 9], normalized discounted cumulative gain [4], expected reciprocal rank [3], and time-biased gain [8].

Weighted precision effectiveness metrics are an important category in which the numeric score that is generated has a direct interpretation as being the rate at which the user of the system gains relevance, measured in units of "gain" per document inspected. The virtue of this style of metric is that each distinct weighting corresponds to a precise *user model*. For example, in the rank-biased precision (RBP) metric of Moffat and Zobel [5], the user is assumed to always inspect the first document in the ranking; thereafter, having viewed the i th document in the ranking, the user is assumed to proceed to the $i+1$ th with conditional probability p, for some fixed value p. Conversely, having reached depth i, the user is modeled as ending their perusal of documents at that depth with probability $1-p$. A probabilistic user with persistence p accrues gain at some overall rate as they proceed through the ranking and eventually end their scan; that rate is the computed RBP score.

If the document at depth i in the ranking is assumed to have a gain (relevance) of r_i, then the score assigned by RBP to a ranking is $\sum_{i=1}^{\infty} W(i)\cdot r_i$, where $W(i)=(1-p)p^{i-1}$. Note that in all weighted-precision metrics, gain can be either binary ($r_i=0$ or $r_i=1$), or can be graded ($0\le r_i$). While it is usual, there is no particular requirement that the r_i values lie in $[0,1]$, and regardless of the nature or range of the r_i values, the sum $\sum_{i=1}^{\infty} W(i)\cdot r_i$ is the expected rate at which gain is accrued from the ranking by a user who has a probability $W(i)$ of viewing the document at rank i.

Another way of defining a user model is via the derived function $C(i)$, the conditional probability of the user examining the document at depth $i+1$, given that they have just examined the document at depth i in the ranking [6]. Provided that $W(i+1)\le W(i)$, there is a clear relationship between $W(i)$ and $C(i)$, with $C(i)=W(i+1)/W(i)$. That is, a sequence of $C(i)$ values also defines a user model, and RBP is equally defined by $C(i)=p$. Any set of weights $W(i)$ that sum to one can be used as the basis of a weighted-precision metric, or any sequence of values $0\le C(i)\le 1$.

2. THE METRIC INST

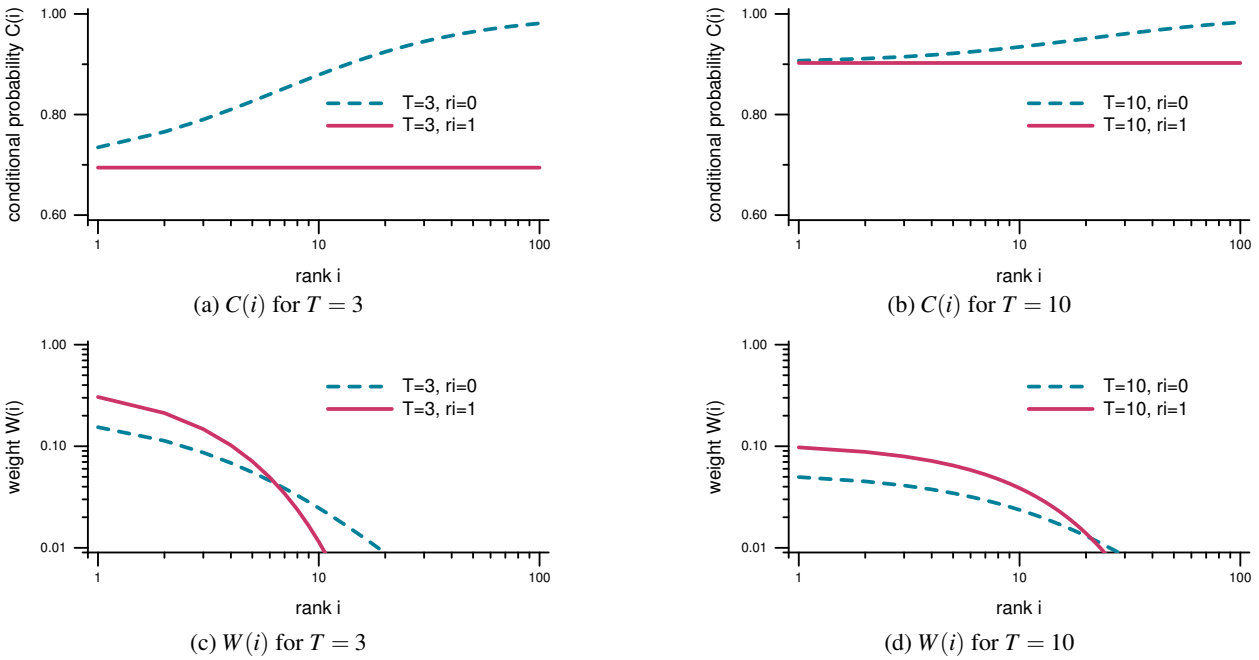
Definition of INST In recent work, Bailey et al. [1] describe a weighted precision metric called INST, defined by the function:

$$C(i)=\left(\frac{i+T+T_i-1}{i+T+T_i}\right)^2, \qquad (1)$$

where T is the number of useful pages that the searcher expects they will need in order to satisfy their information need, $T_i=T-R_i$, and $R_i=\sum_{j=1}^{i} r_j$ is the relevance that has been accumulated so far during the users' inspection of the SERP. Hence, T_i represents the remaining relevance expected still to be gained beyond position i. Given the definition of $C(i)$ provided by Equation 1, the INST

ADCS'15 December 8–9, 2015, Parramatta, NSW, Australia

ACM 978-1-4503-4040-3/15/12 … $15.00.
http://dx.doi.org/10.1145/2838931.2838938.



(a) $C(i)$ for $T = 3$

(b) $C(i)$ for $T = 10$

(c) $W(i)$ for $T = 3$

(d) $W(i)$ for $T = 10$

Figure 1: The conditional continuation probability $C(i)$ (upper) and weighting function $W(i)$ (lower, logarithmic vertical scale) for INST when $T = 3$ (left) and $T = 10$ (right), for two extreme rankings in which $r_i = 0$ throughout (dashed green) and $r_i = 1$ throughout (solid red).

position weightings are calculated as:

$$W(i) = W(1) \times \prod_{j=1}^{i-1} C(j) \;\approx\; \frac{1}{(i+T+T_i)^2}\,, \tag{2}$$

that is, $W(i)$ is inversely proportional to all of i, T, and T_i, in a sequence that converges to a finite sum as i becomes large. Increasing any of i, T, or T_i *decreases* the relative weight of the item at depth i in the ranking, and *increases* the expected search depth.

The value of $W(1)$ that is required in Equation 2 is set so as to ensure that the weights form a probability distribution:

$$W(1) = \left(\sum_{i=1}^{\infty} \prod_{j=1}^{i-1} C(j) \right)^{-1}. \tag{3}$$

A third vector of weights can be computed – also a probability distribution, and equally capable of describing the user model – the probabilities $L(i)$ that the document at rank i is the *last* one inspected by the user before they abandon the SERP [6]:

$$L(i) = \frac{W(i) - W(i+1)}{W(1)}\,. \tag{4}$$

Another useful quantity that helps understand the behavior of a user model is the *expected search depth*, defined as

$$E(W) = \sum_{i=1}^{\infty} i \times L(i) = 1/W(1)\,. \tag{5}$$

The fraction of searches that go beyond some depth i is then

$$\sum_{j=i+1}^{\infty} L(i) = \frac{W(i+1)}{W(1)}\,. \tag{6}$$

Properties of INST The intuition behind Equation 1 is that any particular item is less likely to be the focus of the user's attention as, other things being equal, any of these occurs:

- the user initially anticipates needing a larger number of useful documents; or
- the user's attention shifts further down the ranking; or
- the user has less success in identifying relevant documents.

In particular, for INST the expected search depth falls between approximately $T + 0.25$, when all documents encountered in the SERP are relevant, and $2T + 0.5$, when none of the documents viewed are relevant [1]. Figure 1 plots the continuation and weight functions $C(i)$ and $W(i)$ respectively for INST using two different values of T, and in each case for two extreme situations: rankings in which every item encountered is relevant ($r_i = 1$), and rankings in which every item is non-relevant ($r_i = 0$). As expected, the weights are less top-focused when T is larger, when the depth i is greater, and when non-relevant documents are being encountered in the ranking.

On any actual ranking, the weights and continuation probabilities range between these upper and lower extremes, depending on the number of relevant documents in the ranking, and the depths at which they arise. Even so, there will almost always be a score band admitted by the calculation, because the rankings being scored are finite prefixes of a complete permutation, and because even within that prefix, there may be unjudged documents which have not been assigned gain scores. That is, like RBP and other infinite weighted-precision metrics, a finite ranking gives rise to a score band, defined by a lower bound computed by assuming that all unjudged and/or unknown documents generate no utility, and an upper bound, computed by assuming that they all give rise to the maximum utility possible, usually 1.0.

Table 1 provides a numeric example, showing the computed lower- and upper-bound $W(i)$ values for a ranking containing ten known gain values. In this case, the terseness of the ranking means that even for a small value of T there is non-trivial ambiguity in the final score, and the best that can be said is that $0.306 \le \text{INST} \le 0.406$, and that the expected search depth is between 3.24 and 3.48. The situation

i	r_i	$r_{11+}=0$			$r_{11+}=1$		
		$C(i)$	$W(i)$	$L(i)$	$C(i)$	$W(i)$	$L(i)$
1	0	0.640	0.287	0.360	0.640	0.309	0.360
2	1.0	0.640	0.184	0.230	0.640	0.198	0.230
3	0.5	0.669	0.118	0.135	0.669	0.127	0.135
4	0	0.716	0.079	0.078	0.716	0.085	0.078
5	0	0.751	0.056	0.049	0.751	0.061	0.049
6	1.0	0.751	0.042	0.037	0.751	0.046	0.037
7	0	0.779	0.032	0.025	0.779	0.034	0.025
8	0.2	0.797	0.025	0.018	0.797	0.027	0.018
9	0	0.815	0.020	0.013	0.815	0.021	0.013
10	1.0	0.815	0.016	0.010	0.815	0.017	0.010
11	*undef*	0.831	0.013	0.008	0.815	0.014	0.008
12	*undef*	0.844	0.011	0.006	0.815	0.011	0.007
etc							
$\sum_{i=1}^{\infty} W(i)\cdot r_i$			0.306			0.406	

Table 1: Computing INST, using $T=2$, a SERP of $n=10$ gain values r_i, and two different scenarios beyond $n=10$: that all documents are *not* relevant, and that all documents are *fully* relevant. In this example INST has the value 0.306, with a residual of 0.100.

worsens with larger values of T: when $T=10$, $12.4 \le E(W) \le 18.0$, and $0.139 \le \text{INST} \le 0.513$. That is, the larger the value of T, the longer the prefix of judged documents that is required in order to provide reasonable tolerances on the resulting measurement.

Another key property of INST is that $C(i) < 1$ throughout the range; that is, at all ranks i there is a non-zero probability that the user will end their perusal of the SERP, and either reformulate their query and continue their search, or end their whole search session. Similarly, for all depths i, $W(i) > 0$, and the user might conceivably view the ranking through to any arbitrary depth, albeit with vanishingly small probability. That is, every item in the ranking, regardless of its depth, either contributes in a small way to the final score, or to the uncertainty embedded in that final score. In combination, these properties match well with observed user behavior [6].

3. IMPLEMENTING INST

We now extend the work of Bailey et al. [1] by discussing a number of key issues that affect how INST scores (and scores for some other metrics) are computed in practical situations.

Choosing Gain Values The gain values used in INST can be assigned through any chosen transformation, but with an assumption that the maximum relevance utility gained corresponds to $r_i = 1.0$, and the complete absence of utility corresponds to $r_i = 0.0$. Binary relevance maps trivially to these two values; multi-level relevance labels can be scaled to the interval $[0.0 \ldots 1.0]$.

From a user model perspective, we take the user's utility estimate T to be relative to the sum of the gain values r_i. For binary relevance labels this is again trivial, and corresponds to computing R_i as the number of relevant documents encountered at or before depth i. For multi-level relevance, one would expect to need to encounter more partially relevant documents than fully useful documents as captured by the estimate T. This expectation corresponds to requiring more partial gain values (where $r_i < 1$) to get to a certain total level of relevance T. That is, we add the partial gains to get R_i, rather than applying thresholding to get binary addends.

Performing the Calculation Algorithm 1 describes the process of computing an INST base score and residual, given a value for T, and a document ranking of length n, possibly including unjudged documents. Two cycles of computation are performed, the first to calculate $score_0$, the lower bound on the possible score range that arises when all unjudged documents are assumed to be non-relevant and generate zero gain; and then a second to calculate $score_1$, the upper bound on the score range assuming that all unjudged documents generate maximum gain, including in a trailing tail of documents beyond depth n in the ranking.

Algorithm 1 Computing INST for a document ranking.

Require: A ranking of documents $\langle d_i \mid 1 \le i \le n \rangle$; a set of relevance judgments J defined by $J : d \to \{undef\} \cup [0 \ldots 1]$, where d is a document number; a value T being the relevance total estimated as being required.

Ensure: *INST* is the base score for the document ranking; *residual* is the sum of the score uncertainty caused by unjudged documents within and beyond the end of the document ranking.

default $\leftarrow 0$
2: $N \leftarrow limit(T)$ ▷ needed to ensure *sumW* is accurate
score $\leftarrow 0$
4: $W(1) \leftarrow 1$ ▷ will eventually be scaled to correct value
sumW $\leftarrow 0$ ▷ the eventual scaling factor
6: $T_0 = T$
for $i \leftarrow 1$ **to** $\max(n, N)$ **do**
8: **if** $i > n$ **then**
$r_i \leftarrow default$ ▷ document not provided
10: **else if** $J[d_i] = undef$ **then**
$r_i \leftarrow default$ ▷ document provided, but not judged
12: **else**
$r_i \leftarrow J[d_i]$ ▷ document provided and judged
14: **end if**
$T_i \leftarrow T_{i-1} - r_i$
16: $score \leftarrow score + r_i \times W(i)$
$sumW \leftarrow sumW + W(i)$
18: use Equation 1 to compute $C(i)$ from i, T, and T_i
$W(i+1) \leftarrow W(i) \times C(i)$ ▷ prepare to iterate
20: **end for**
$score_0 \leftarrow score/sumW$ ▷ scale to get lower bound on score
22: *default* $\leftarrow 1$
repeat steps 3 to 20 to compute *score* and *sumW* again
24: $score_1 \leftarrow score/sumW$ ▷ upper bound on final score
INST $\leftarrow score_0$
26: *residual* $\leftarrow score_1 - score_0$
return $(INST, residual)$

Residuals and Infinite Tails An issue that is specific to INST arises at step 2. For RBP, the non-adaptive way in which $C(i) = p$ is defined means that the weights $W(i)$ can be determined formulaically, and that the *tail sum*, the values of $W(i)$ *not* included in any finite-depth computation, can also be directly calculated [5].

With INST, the situation is more complex, both because of the different underlying weighting regime, and also because the weights are altered as relevance is encountered. As a general rule, the greater the expected depth of evaluation, the greater the tail residual of any particular finite ranking, and with INST, the expected depth is maximized for a ranking in which $r_i = 0$ throughout. Since $\sum_{i=1}^{\infty}(1/i^2) = \pi^2/6$, the tail-sum weighting ρ_N for INST beyond depth N on an all-zero ranking can be computed as:

$$\rho_N = \sum_{i=N+1}^{\infty} W(i) = 1.0 - \frac{1}{S_{2T-1}} \cdot \sum_{i=1}^{N} \frac{1}{(i+2T-1)^2}\,, \qquad (7)$$

where $S_m = \pi^2/6 - \sum_{j=1}^{m}(1/j^2)$. If the objective is to determine an evaluation depth N for which the residual is less than some defined

	Expected depth		
	2.58	6.53	20.51
INST	$T = 1$	$T = 3$	$T = 10$
Eval. depth n, $\delta = 0.05$	30	105	371
– prob. going beyond	0.39%	0.29%	0.26%
Eval. depth n, $\delta = 0.01$	154	547	1931
– prob. going beyond	0.02%	0.01%	0.01%
RBP	$p = 0.612$	$p = 0.847$	$p = 0.951$
Eval. depth n, $\delta = 0.05$	7	19	60
– prob. going beyond	3.22%	4.26%	4.91%
Eval. depth n, $\delta = 0.01$	10	28	92
– prob. going beyond	0.74%	0.96%	0.98%

Table 2: Evaluation depth required to meet upper bound limits δ on residuals. The three RBP parameters are chosen to match the expected depth of INST with three values of T.

value δ, then a linear search can be used to identify the smallest N for which $\rho_n < \delta$. There are two places where this calculation is required. In the first, at step 2, a value *limit* is determined as a function of T, so that the numerical calculation will not be compromised by a too-shallow computation of *sumW* in Algorithm 1. If scores and residuals are to be reported to three decimal places, $\delta < 5 \times 10^{-4}$ is appropriate; hence, when $T \le 5$, $N = 2 \times 10^4$ suffices, and for $T \le 50$, $N = 2 \times 10^5$ is necessary.

Equation 7 also guides the ranking depth n required to reach a certain resolution in the computed INST scores. For example, we might require that n, the length of the document ranking, be such that the tail residual be guaranteed to be smaller than 0.05, or even 0.01. Table 2 lists the minimum values n for which the INST residual is certainly less than these two values of δ, for three different values of T. For example, if the INST residual is to be less than 0.05, and if $T = 3$, then documents might need to be judged to a depth of as much as $n = 105$. The second half of the table shows the equivalent computation for RBP, with three values of p chosen so as to result in the same expected search length as INST with $T = 1$, $T = 3$, and $T = 10$. The judging depths that arise are much smaller than for INST, because INST has a longer tail in its probability distribution $W(i)$ than does RBP. However, in the case of INST, the presence of relevant documents in the ranking also decreases the residual, an effect that does not apply to RBP. For example, a ranking of ten $r_i = 0$ values with $T = 2$ gives a residual of 0.150; the example shown in Table 1, which is also to depth $n = 10$, has a residual of 0.100; and a ranking of ten $r_i = 1$ values with $T = 2$ has a residual of just 0.006. The dramatic change is a consequence of the presence of relevant documents shortening the expected search length.

The second line in each pair in Table 2 shows the fraction of users that are expected to surpass the indicated value of n if presented with an all-zero ranking, based on Equation 6. As already noted, for a given guaranteed level of residual fidelity, INST requires much deeper relevance judgments than does RBP, because it is less strongly top-weighted. But in a probabilistic sense, a much smaller fraction of search sessions will reach that deeper level, especially if relevant documents occur near the head of them; and hence more tolerant values of "pessimal δ" may be appropriate, at least partially negating the need for the extra judgments. That is, with INST, the value of the residual for a run can be bounded in advance, but not exactly computed in advance, because the run itself affects the value of the residual, not just the depth of judged documents it contains.

Handling Ties Ties in the ranking order also pose a challenge in INST, because of the adaptive nature of the metric. One approach to ties is to assert that ties cannot exist in current SERPs, because an ordering decision must always be made, and hence it suffices to simply assess each run in the order the documents are presented, without regard to the document scores. But this head-in-the-sand approach avoids the problem rather than deals with it, and SERPs may well emerge in which ties are genuinely permissible. A second option is to adopt the methodology used in William Webber's `rbp_eval` implementation,[1] which treats each set of equal-rank items as a single combined document, sums the corresponding $W(i)$ weights, and applies the average weight $\overline{W(i)}$ equally across the group; the drawback of this mechanism is that in INST the calculation of $W(i)$ is on a per-rank basis, and depends both on what comes before and what comes after the i th document, with a feedback loop that makes all of the $W(i)$ values dependent on all of the r_i values. A third alternative is to consider all permutations of the tied documents and somehow average the set of final scores that emerge; but this would greatly (perhaps fatally) expand the execution time of an implementation if a long group of tied document scores is presented. A fourth option is to choose at random a single permutation of the tied group, and then apply the metric; but this means that scores are non-deterministic, and potentially non-repeatable.

In the case of INST, our preferred implementation option is a fifth mechanism – we suggest that the gain values r_i for any tied groups be averaged to get a single $\overline{r_i}$ value used for each and every document in the group, and that Algorithm 1 be applied without further modification. That is, once past the group of tied documents, the user's R_i will be correct. This approach has the benefits of being deterministic, being linear-time in terms of execution, and of being "fair" across the tied elements. We note in passing that one of the quirks of the widely-used `trec_eval` program[2] is that ties are handled by sorting each tied group into reverse document identifier order; while deterministic, this approach is less defensible than the ones we have canvassed above in terms of treating documents fairly.

Acknowledgment This work was supported by the Australian Research Council's *Discovery Projects* Scheme (project DP140102655).

References

[1] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. User variability and IR system evaluation. In *Proc. SIGIR*, pages 625–634, 2015.

[2] B. Carterette. System effectiveness, user models, and user utility: A conceptual framework for investigation. In *Proc. SIGIR*, pages 903–912, 2011.

[3] O. Chapelle, D. Metzler, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proc. CIKM*, pages 621–630, 2009.

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

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

[6] A. Moffat, P. Thomas, and F. Scholer. Users versus models: What observation tells us about effectiveness metrics. In *Proc. CIKM*, pages 659–668, 2013.

[7] S. Robertson. A new interpretation of average precision. In *Proc. SIGIR*, pages 689–690, 2008.

[8] M. D. Smucker and C. L. A. Clarke. Time-based calibration of effectiveness measures. In *Proc. SIGIR*, pages 95–104, 2012.

[9] E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Proc. CIKM*, pages 102–111, 2006.

[1] `www.williamwebber.com/research/downloads/rbo-0.2.1.tar.gz`

[2] `trec.nist.gov/trec_eval/`