Splitting Water: Precision and Anti-Precision to Reduce Pool Bias

Aldo Lipani
Inst. of Software Technology & Interactive Systems
Vienna University of Technology
Vienna, Austria
lipani@ifs.tuwien.ac.at

Mihai Lupu
Inst. of Software Technology & Interactive Systems
Vienna University of Technology
Vienna, Austria
lupu@ifs.tuwien.ac.at

Allan Hanbury
Inst. of Software Technology & Interactive Systems
Vienna University of Technology
Vienna, Austria
hanbury@ifs.tuwien.ac.at

ABSTRACT

For many tasks in evaluation campaigns, especially those modeling narrow domain-specific challenges, lack of participation leads to a potential pooling bias due to the scarce number of pooled runs. It is well known that the reliability of a test collection is proportional to the number of topics and relevance assessments provided for each topic, but also to same extent to the diversity in participation in the challenges. Hence, in this paper we present a new perspective in reducing the pool bias by studying the effect of merging an unpooled run with the pooled runs. We also introduce an indicator used by the bias correction method to decide whether the correction needs to be applied or not. This indicator gives strong clues about the potential of a "good" run tested on an "unfriendly" test collection (i.e. a collection where the pool was contributed to by runs very different from the one at hand). We demonstrate the correctness of our method on a set of fifteen test collections from the Text REtrieval Conference (TREC). We observe a reduction in system ranking error and absolute score difference error.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation*

General Terms

Experimentation, measurement, performance

Keywords

Evaluation, bias, pool, test collection, TREC

1. INTRODUCTION

A test collection is a valuable resource for Information Retrieval (IR) researchers because it gives the IR community a common ground to facilitate the development of search models. Numerous test collections have been developed in the field since the first Cranfield experiments in the 1960s. Since the start of TREC in the 1990s, this creation happens at a rate of approximately 25 test collections per year. A test collection is composed of: a set of documents, a set of topics and a set of relevance assessments for each topic, derived from the collection of documents. The number of documents in the collection generally makes the full judgment of the document set for every topic infeasible. Therefore, the relevance assessment process is generally optimized by pooling the top N documents for each run. The pool is constructed from systems taking part in the challenge for which the collection was made, at a specific point in time, after which the collection is generally frozen in terms of relevance judgments. The pooling technique aims to identify an unbiased sample of relevant documents. Nevertheless, pool bias negatively affects the score of unpooled runs—those of systems not present at the time of test collection creation. This is a drawback that ultimately affects the reliability of the test collection. The variables controlling this reliability are [14]: the number of topics and their representativeness of the information needs of the target user, the number of documents assessed per run, and, last but not least, the diversity of the pooled systems (often however only assessed as the cardinality of the set of runs).

In the last decades the IR community has branched out significantly in a variety of domains and applications, with the creation of specific IR test collections focusing on specific problems. At the same time, benchmarking techniques developed in the IR community are being implemented in industry. Information aware companies request measures to quantify the quality of their information access systems in general, and search systems in particular. With a narrower focus however, the effort to successfully solve the challenges facing the creators of test collections takes on new significance. Most notably, it is often difficult to acquire a sufficient number of participants and diverse systems in order to fulfill the required run diversity to guarantee a reliable test collection.

In this paper, we estimate the pool bias by studying the effect of an unpooled run on the set of pooled runs, when a fixed-depth pooling strategy is used. We do this through the estimation of an *average unjudged rate*, which we then normalize with its potential growth interval, in order to adjust the pool bias. Additionally, we introduce an indicator that

SIGIR'15, August 09 - 13, 2015, Santiago, Chile.

ACM 978-1-4503-3621-5/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2766462.2767749 .

provides strong clues about the quality of a new, unpooled run.

We do this based for Precision at cut-off $P@n$. There are two reasons to consider such a "simple" metric. First, it is a cornerstone for many other metrics developed for the most popular of user models these days: the web user [12]. Second, it is easy to understand by all users. This "understandability" of the IR metrics has drawn moderate attention from our community recently [10]. Our own experience in the industry leads us to believe that, when results are not presented as simply precision and recall, any numbers are just *assumed* to be precision or recall. Decision makers at lower or higher levels, trying to make sense of MAP, or any other commonly used metric in our community, will most often read 0.12 as 12% and simply assume that either 12% of documents are relevant or 12% of relevant documents have been returned on average. Of course, we do not forget why all the other metrics have been invented to replace, or complement, precision at cut-off: 1) for an ideal run, if the topic has less relevant documents then n, $P@n$ does not reach 1; it is not normalized by the number of relevant documents, therefore it is difficult to average over topics, 2) it partially neglects the position of the documents. Nevertheless, there are many cases where $P@n$ is useful (most often, but not only, for the user modeled as considering blocks of 10 documents at a time on the web). This is also demonstrated by its continued use and reporting throughout a majority of evaluation tracks at TREC, CLEF, or NTCIR.

We propose a new bias correction method and demonstrate its effectiveness through leave-one-out experiments, at different levels and combinations, organizations and systems, all the pooled runs, or only the 75% of the top runs as done in previous papers [22, 19, 2, 21, 20]. We then evaluate the results using the mean absolute error (MAE) and the system rank error (SRE), comparing it against the results obtained with the reduced pool, and with the method of correcting pool bias introduced by Webber and Park [23]. We do this on fifteen test collections from TREC, five of which are domain specific test collections.

In short, the contributions of this study are as follows:

1. a new perspective on $P@n$, based on the effect of a new run on the set of runs which contributed to the creation of the pool;
2. an indicator to trigger bias correction only when it is indeed necessary;
3. a bias correction method for $P@n$, including extensive experimental results to show that it outperforms existing bias correction methods,

The remainder of the paper is structured as follows: in Section 2 we provide a brief summary of the extensive work already done to assess and correct pool bias. Section 3 provides the intuition of our method and introduces the required concepts, followed in Section 4 by the method itself. In Section 5 we present and discuss our experimental results. We conclude in Section 6.

2. RELATED WORK

Work related to pool bias can be grouped in three categories: first, that aiming to fundamentally change the way assessment is done, by, instead of pooling, choosing assessment documents in order to maximize some evaluation goal. For instance, Cormack et al. [11] suggest to boost the proportional of relevant documents, Moffat et al. [15] to focus on the score accuracy of the best-performing systems, Carterette et al [7, 6] to maximize confidence that one system has or has not a better score then another one, using different models of probability of relevancy. While important, this is not the focus of the current paper, which instead addresses the problem of evaluating against existing test collections built using pooling.

Second, there are those studies aiming to assess the reliability of a test collection. This reliability (or lack thereof) can be often traced back to the pooling procedure. Most recently Urbano et al. [20] proposed an estimation of reliability of a test collection using Generalization Theory. We shall use this in our study to better understand the observations made in our experiments.

Finally, and most related to this study, are those works that address the problem of pool bias directly. Here, three different strategies have been studied: removing the bias from the onset, at test collection creation time; creating new metrics to better handle unjudged documents; or estimating the score error to make an adjustment in the metric.

The first strategy is the most desirable: enforcing a diverse set of runs through the efforts of the test collection creators themselves. Especially early test collections have, for instance, created manual runs to increase the likelihood of relevant documents appearing in the pool. The benefit of such efforts has been demonstrated, among others by Kuriyama et al. [13]. Nevertheless, not all test collections have this advantage, and adding such runs a posteriori, after an initial set of systems have been evaluated, is not done because it breaks the comparability of the runs evaluated across the years.

A lot more can be done and has been done for the second strategy: new metrics. In 2004, Buckley and Voorhees [4] introduced BPref as a metric specifically designed to handle incomplete information, which, as pointed out by Sakai in 2007 [17], is a restricted form of Average Precision (AP) on a so called 'condensed list'. These are condensed versions of the runs where unjudged documents are filtered out. Sakai introduces a new metric (the Q-measure) and shows that it is possible to obtain better performance then BPref even applying already well-known metrics to the condensed list. The concept of condensed list, first denoted as such by Sakai, was however already explored in relation to AP with the measure Induced AP, introduced by Yilmaz and Aslam [24] the year before, in 2006. Induced AP is Average Precision calculated on condensed lists. The methods explored by these three contributions do not simulate the effect of shallow pooling or of comparing unpooled runs against pooled ones, because they remove the effect of bias sampling from the query relevance (qrel) set, ending up with an unrealistic use case. This was later addressed by Sakai [18], who demonstrated that the condensed list approach leads in favor of new systems, the effect of these metrics instead creating incomplete relevance information by playing with the depth of the pool. Also in their 2006 report, Yilmaz and Aslam [24] introduce the Inferred AP, a more complex metric which is a closer approximation of AP but requires knowledge about the documents down to depth 100. Inferred AP adjusts the score sampling uniformly from the pooled documents and then estimates the true mean of the sample to adjust AP.

Later, Aslam and colleagues address the issue of the uniform sampling used in the 2006 version of Inferred AP [25]. In this later version they use a stratified sampling scheme.

However, their finding that the method is not subject to pooling bias is not confirmed in practice by Carterette et al. in 2008 [8]. As they write, this is possibly because aggregating probability of inclusion across multiple runs by taking the mean of the per run probabilities may not properly account for reinforcement by similar systems.

The problem of incomplete judgments leads to the definition of a completely new metric, defined by Moffat and Zobel [16]—called Rank-Biased Precision—expressed by a single value and a residual. The Residual quantifies the uncertainty introduced by the unjudged documents. Its value is computable thanks to the fact that it is not normalized by the number of relevant documents. This implies that the computation of the metric defines a lower bound for the given run. Moffat and Zobel attempted to make a measure that is naturally convergent, where the contribution of each rank has a fixed weight. This would have both benefits of a normalized metric and those of a metric averageable over topics with different numbers of relevant documents. This attempt was unsuccessful, as pointed out by Sakai [18], who proved this to be inferior with respect to the condensed list.

At this point we have the transition to the third category of approaches to solve the pool bias: metric error estimation and correction.

In their presentation of Rank-Biased Precision (RBP), Moffat and Zobel had already introduced the discussion concerning the fact that the residual can be used to estimate and correct pool bias. Webber and Park [23] continue their work on RBP by adding to the score the average residual calculated against the pool proceeding with a leave-one-run-out approach. To estimate it they span two dimensions: the topics and the systems. They used Rank-Biased Precision at ten (RBP@10) and Precision at ten (P@10) although the results for this last metric were not reported in the 2009 paper, the authors only mentioned that they were similar to RBP. In the present study we return to precision at cut-off and look not only at coefficients to correct pool bias, but also at whether there is something to correct in the first place.

3. PRECISION AND ANTI-PRECISION

The intuition at the base of the proposed method is that we can observe how a new, unpooled run impacts the existing, pooled runs. Given such an existing run, we can imagine reranking it based on the ranks of its documents in the unpooled run. A "bad" new run will tend to bring down known relevant documents and push up non-relevant ones. Quantifying these changes we create a measure of the potential quality of the new run.

In the following we describe theoretically the measures later used to reduce the pool bias. In evaluating IR systems, Precision (P) is one of the two fundamental measures. We recall its definition: given D a set of documents, D_r a subset of D (the documents in a run r), q a topic, and σ a function of relevancy returning the level of relevancy of the document d for the topic q, P is defined as:

$$P = \frac{|d \in D_r : \sigma(d, q) > 0|}{|D_r|}$$

Precision represents the proportion of relevant and retrieved documents against the retrieved ones. From P we derive the definition of Precision at cut-off n ($P@n$), used to better handle ranked retrieval systems: given ρ a function that returns the rank of a document d in a run r, we have:

$$P@n = \frac{|d \in D_r : \sigma(d, q) > 0, \rho(d, r) \leq n|}{n}$$

The measure takes into account only the relevant documents because it is supposed to be used when there is a complete knowledge of the relevance function over the documents in the run. When we consider the problem of missing relevance assessments this assumption is not true, ending up considering unjudged documents as non-relevant. To overcome this problem and take into account the missing information about the run, we define the complement of Precision, called Anti-Precision ($\overline{P}$). Anti-Precision measures the proportion of non-relevant and retrieved documents against the retrieved documents. In statistics, a similarly defined quantity is referred to as the False Discovery Rate (FDR) [1]. It is used in quantifying the results of multiple hypothesis testing experiments. However, given the very different use of it here, we continue to refer to it as Anti-Precision in this study, and define it as:

$$\overline{P} = \frac{|d \in D_r : \sigma(d, q) = 0|}{|D_r|}$$

As well as for Precision, we define also the cut-off version ($\overline{P}@n$):

$$\overline{P}@n = \frac{|d \in D_r : \sigma(d, q) = 0, \rho(d, r) \leq n|}{n}$$

Indeed, when a run is fully judged the following equation holds:

$$P + \overline{P} = 1$$

When it is not, and unjudged documents are present in the run, the sum of P and $\overline{P}$ is lower than 1, reduced by a quantity that represents the proportion of retrieved and unjudged documents against the retrieved documents. We refer to this as k bar ($\bar{k}$).

$$P + \overline{P} = 1 - \bar{k}$$

This quantity represents the uncertainty of the measurement. Just as P and $\overline{P}$, $\bar{k}$ can be also defined at cut-off ($\bar{k}$@n).

3.1 Analysis of a run shuffle

Before going on to the details of our proposed method, let us perform an imagination exercise in order to better understand the information content of a partially judged run. We want to analyze which kind of information precision and anti-precision expose if a given run r gets shuffled. As in a deck of cards a shuffling changes the order of the documents of a run and produces a new run that we will indicate as r'. This run has the same set of documents as before. We want to observe the variation in score the run obtains in the two states, original and shuffled. If we would use P, since there is no information about the position of the documents in the formula, we would measure a change of 0. Therefore, let us observe $P@n$. Given a run r and its shuffled version r' we define:

$$\delta P@n(r') = P@n(r') - P@n(r)$$

$\delta P@n$ has domain $[-1, 1]$ and is the variation in precision of the run after a shuffle. Its increase in value is the result of the combination of the following two related effects: the shuffle

moved up relevant documents, placing them in the top n, or moved down non-relevant or unjudged documents with the consequential moving up of potential relevant documents in the run. It decreases if the opposite happens. We also define $\delta\overline{P}@n$ as following:

$$\delta\overline{P}@n(r') = \overline{P}@n(r') - \overline{P}@n(r)$$

$\delta\overline{P}@n$ has domain $[-1, 1]$ and is the variation in anti-precision of the run after a shuffle. Its increase in value is the result of the combination of the following two related effects: the shuffle moved up non-relevant documents, placing them in the top n, or moved down relevant or unjudged documents with the consequential moving up of potential non-relevant documents in the run. It decreases if the opposite happens.

Finally, $\delta\bar{k}@n$ that is derived as following:

$$\begin{aligned}\delta\bar{k}@n(r') &= \bar{k}@n(r') - \bar{k}@n(r) \\ &= 1 - (P@n(r') + \overline{P}@n(r')) \\ &\quad - [1 - (P@n(r) + \overline{P}@n(r))] \\ &= -\delta P@n(r') - \delta P@n(r)\end{aligned} \tag{1}$$

$\delta\bar{k}@n$ has domain $[-1, 1]$ and is the variation of *unjudged* documents on a given run. Its increase in value is the result of the combination of the following effects: the shuffle moved up unjudged documents or moved down relevant and non-relevant documents with the consequential moving up of potential unjudged documents in the run. An interesting property of this function, which is possible to prove, is that if r has been judged to depth $d : d \geq n$, then the domain of the function $\delta\bar{k}@n$ is $[0, 1]$. This property always holds for pooled runs because they verify the condition (provided of course that no mistakes occurred in the pooling process).

In summary, when a run changes the order of its documents, δP, $\delta\overline{P}$, and $\delta\bar{k}$ are indicators of the direction of the judged relevant, judged non-relevant, and unjudged documents in the list.

3.2 Effect of a run on a pooled run

Now let us make a step further and consider not the relationship between a run and a random shuffle of itself, but between a run and another run. In the particular case where each run ranks completely the entire collection, this is the same as above. In general however, the systems only provide runs down to a certain limit (say 1000). To study this effect, we need to define a merging function between the two runs. The unpooled run will have an effect on the pooled run, measured by the quantities described above.

Such a merging function can simply be based on the rank of the documents in the run. The aim here is not to add or remove documents from a run, so although the word "merging" could imply the transfer of documents between the two runs to make a new one, we must keep in mind that all we need to do here is transfer only the information about the rank of the documents. We do this by linearly combining the ranks if the two runs share the same document.

In the following formula, by r_u we denote the new, previously unseen and unpooled run, whose effect on r_p an existing run, we want to study. This effect we represent as a new, synthetic run r', which consists exclusively of documents present in r_p, potentially re-ordered.

$$r' = r_p \circ r_u = \{d \in r_p : \rho(d, r') = \mu(d, r_p, r_u)\} \tag{2}$$

where

$$\mu(d, r_p, r_u) = \begin{cases} \rho(d, r_p)(1-\alpha) + \rho(d, r_u)\alpha & \text{if } d \in r_u \\ \rho(d, r_p) & \text{if } else \end{cases}$$

μ is the weighted arithmetic mean between the rank of the document in r_p and the rank of the document in r_u, with $0 \leq \alpha \leq 1$. When the same rank is assigned by μ to two different documents, which can happen in some cases for a pair of documents of which one is also in r_u and the other one is not, the common document is inserted after the r_p-exclusive document. In other words, the original run rank has priority.

As any functional composition operator, our merging operator $\circ$ is not commutative and always represents the effect of its right member on its left member.

In this context, $\delta P@n$ and $\delta\overline{P}@n$ can be used to analyze the quality of an unpooled run against the pooled one. An increase in $\delta P@n$ is the result of two forces, one direct and one indirect: 1) direct, if the relevant documents in the top n of r_u are the same documents found at the bottom of r_p, they will be pushed up; 2) indirect, if the r_u has non-relevant or unjudged documents in the bottom that are in the top n documents of r_p, they will be pushed down. The contribution decreases if the contrary happens. As well for $\delta\overline{P}@n$ the contribution is: 1) direct, if the non-relevant documents in the top n of r_u are shared with documents in the bottom of r_p; 2) indirect, if the r_u has relevant or unjudged documents in the bottom that are in the top n documents of r_p. If the run r_p would be judged in its totality, these two effects would be perfectly correlated and it would be possible to calculate one just knowing the other from the following equation:

$$\delta P@n + \delta\overline{P}@n = 0$$

However, when r_p contains unjudged documents at ranks below n, their sum becomes $-\delta\bar{k}@n$, as shown in Eq. 1.

As explained above, $\delta\bar{k}@n$ represents the ratio of unjudged documents brought to the top n of the run r_p by the run r_u. Moreover, it is possible to prove that $\delta P = 0$ and $\delta\overline{P} = 0$ if and only if one of the following two conditions occurs: 1) the two runs r_p and r_u do not share any documents with each other in their top n documents, or 2) the two runs are identical in the top n. These are the two cases where our method will not say anything about the new run r_u just by using the existing run r_p (but we might based on other pooled runs).

Let us now take an example to illustrate how this indicator could be useful to understand the behavior of a run and predict its quality. We use the test collection Robust 2005 and in particular we focus our attention on a special run that presents an unusual effect, the routing run `sab05ror1`. It has the peculiarity of being strongly discounted when it is not in the pool. Buckley et al. [3] studied it at length, pointing out that the reason for its behavior was related to the size of the test collection. For this run let us calculate $P@10$ and $\bar{k}@10$. Let us also consider the average of $\delta P@10$ and $\delta\overline{P}@10$, which we denote as follows:

$$\Delta P@10 = \frac{1}{|R_p|} \sum_{r \in R_p} \delta P@10(r_p \circ r_u)$$
$$\Delta\overline{P}@10 = \frac{1}{|R_p|} \sum_{r \in R_p} \delta\overline{P}@10(r_p \circ r_u)$$

where R_p is the set of runs used in the creation of the test collection.

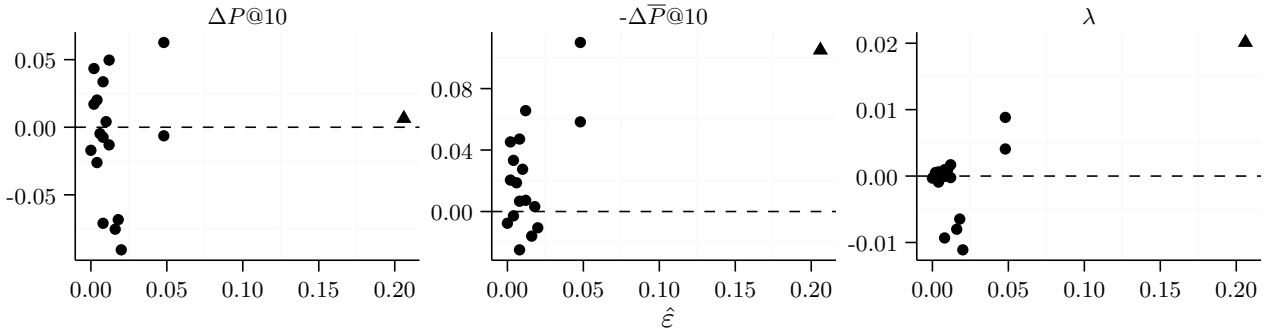

Figure 1: Plot of $\Delta P@10$, $\Delta\overline{P}@10$ and λ against the residual ($\hat{\varepsilon}$) in a leave-one-organization-out experiment, for the Robust 2005 test collection. The run indicated as ▲ is the unusual run `sab05ror1`.

Table 1: Measures computed for the run `sab05ror1` when it is not part of the pool

$P@10$	$\bar{k}@10$	$\Delta P@10$	$\Delta\overline{P}@10$
0.4220	0.444	0.0065	-0.1053

Table 1 shows these values for this particular run.

When the run is not part of the pool, $P@10$ assigns it the 11th position in 18th runs. $\bar{k}@10$ says that there are many documents that are unjudged and that therefore there is a high potential to grow. $\Delta P@10$ indicates a low average positive contribution to the pooled runs, and shows that among the relevant documents there is little intersection. $\Delta\overline{P}@10$ instead is negative which suggests that many non-relevant documents have been ranked lower than before, therefore suggesting a good ability of this special run to discriminate relevant documents from non-relevant ones.

In Figure 1 we show the resulted $\Delta P@10$, $\Delta\overline{P}@10$ against the residual error ($\hat{\varepsilon}$, the difference between the true score and the unpooled score), generated with a leave-one-organization-out approach. Here we can observe that just using $\Delta P@10$ is not enough because it takes into account only one of the two positive contributions of the run, the other one being the reduction in $\Delta\overline{P}@10$.

Let us now return to the general case. When the average negative contribution of the unpooled run to other runs is reduced (i.e. $\Delta\overline{P} < 0$) and the run has a positive contribution (i.e. $\Delta P > 0$), the run suffers from pool bias and its score should be adjusted. More problematic is the case when $\Delta\overline{P}$ and ΔP have the same sign (i.e. the run has both a negative and a positive contribution, on average). Indeed, if we have $\Delta\overline{P} > 0$ and $\Delta P > 0$ we would improve the $P@n$ score of the run only if their ratio is greater then the ratio of P to $\overline{P}$, because it means that there is a chance to improve the existing score. On the other hand, if we have $\Delta\overline{P} < 0$ and $\Delta P < 0$ we would improve only if their ratio is lower then the ratio of P to $\overline{P}$ because it means that the contribution of the run is more able to discriminate the non-relevant documents.

From these observations we derive a single value indicator that merges the information of all the indicators defined:

$$\lambda = \bar{k}@n(\Delta P@n \cdot \overline{P}@n - \Delta\overline{P}@n \cdot P@n) \qquad (3)$$

For all runs where $\lambda > 0$ we apply our correction method.

The sign of the difference in the brackets is equivalent to the ratios discussion above. The $\bar{k}$ factor has no impact on the sign (as $\bar{k} \geq 0$), but removes those special cases where $\bar{k} = 0$, since in these cases there is no possibility to improve the score of the run (i.e. to get it closer to what we would have obtained if the run had been contributing to the pool).

Returning briefly to the example of the `sab05ror1` run, we can now see in Figure 1 that λ clearly distinguishes this run from the rest.

Algorithm 1 Adjustment based on pooled runs

$r_u \leftarrow$ unpooled run
$R_p \leftarrow$ set of pooled runs
$T \leftarrow$ set of topics
$Q \leftarrow$ qrels on T derived from R_p
$s_{r_u} \leftarrow P@n(r_u)$
$\bar{s}_{r_u} \leftarrow \overline{P}@n(r_u)$
$\bar{k}_{r_u} \leftarrow 1 - (s_{r_u} + \bar{s}_{r_u})$
for all $r_p \in R_p$ **do**
  $r'_p \leftarrow r_p \circ r_u$
  $\delta P_{r_p} \leftarrow (P@n(r'_p)$ - $P@n(r_p))$
  $\delta\overline{P}_{r_p} \leftarrow (\overline{P}@n(r'_p)$ - $\overline{P}@n(r_p))$
  $\delta\bar{k}_{r_p} \leftarrow (-\delta P r_p - \delta\overline{P} r_p)$
end for
$\Delta P_{r_u} \leftarrow \frac{1}{|R_p|} \sum_{r_p \in R_p} \delta P_{r_p}$
$\Delta\overline{P}_{r_u} \leftarrow \frac{1}{|R_p|} \sum_{r_p \in R_p} \delta\overline{P}_{r_p}$
$\lambda \leftarrow \bar{k}_{r_u}(\Delta P_{r_u} \bar{s}_{r_u} - \Delta\overline{P}_{r_u} s_{r_u})$
if $\lambda > 0$ **then**
  $\Delta\bar{k}_{r_u} \leftarrow \frac{1}{|R_p|} \sum_{r_p \in R_p} \delta\bar{k}_{r_p}$
  $a \leftarrow \bar{k}_{r_u} \max(\Delta\bar{k}_{r_u}, 0)$
else
  $a \leftarrow 0$
end if
return $s_{r_u} + a$

4. ADJUSTING SCORE FOR BIAS

Now that we have an understanding of which runs are suffering from pool bias, with respect to precision at cut-off, we proceed by presenting our method to adjust the score (Algorithm 1). As hinted at before, the method is to adjust the pool bias suffered by a system that has not been pooled by measuring the effect of the system on the pooled runs. The only information that is needed is the relevance assessments for each topic and the pooled runs, normally available for most existing test collections. As we presented earlier, $P@n$ as calculated with the incomplete pool is a lower bound for the score of r_u. To correct the pool bias we want to add a

quantity that stays within its uncertainty limit $\bar{k}_{r_u}$. In other words, our growth potential in terms of $P@n$ is bounded by $\bar{k}_{r_u}$. We are interested in estimating the missing precision of the unjudged documents in the run r_u.

The question is then: Where in this interval do we find our correction value? In the absence of any other external information, we will take the average effect of this run r_u on the existing runs, in terms of $\bar{k}$.

We do this by computing the $\delta\bar{k}_{r_p}$ produced by r_u on a pooled run r_p via the run composition function defined in Eq. 2. This measures the aggregated change in precision and anti-precision, as described by Eq. 1. We do this for each run in the pool, and average these values. This average, denoted $\Delta\bar{k}_{r_u}$, when positive, acts as a maximum likelihood estimator for our position in $[0,\bar{k}_{r_u}]$. Therefore, the correction quantity is the product between $\Delta\bar{k}_{r_u}$ and $\bar{k}_{r_u}$. The following section will therefore add $\bar{k}_{r_u}\max(\Delta\bar{k}_{r_u},0)$ to the $P@n$ for those runs with $\lambda > 0$, as shown in the last seven lines of Algorithm 1.

5. EXPERIMENTS

To test the pool bias adjustment developed in the previous section we used 15 test collections sampled from TREC: 7 test collections from the Ad Hoc track, 3 from the Web track, and 5 from more domain specific IR tracks: Genomics, Robust, Legal, Medical and Microblog. We tested the algorithm[1] through a leave-one-out approach comparing our method with that of Webber and Park [23]. As the baseline we consider the traditional evaluation against the reduced pool. We call this the *reduced* pool to distinguish it from the ground truth pool—the one also containing documents exclusively contributed by the removed runs or organizations. We performed the leave-one-out at two different levels: 1) *leave-one-run-out*: as firstly described by Zobel [26], one run at a time is exited from the pool. This is done by removing all the documents uniquely introduced by it from the relevance assessments; 2) *leave-one-organization-out*: as introduced by Büttcher [5], it is similar to the *leave-one-run-out*, with the difference that not only is one run removed from the pool, but also all the runs generated by the same organization. This is done by removing all the documents uniquely introduced by the organization's runs from the relevance assessments. This second approach simulates better the testing of a new run, since in most cases it has been observed that the runs produced by the same organization come from the same system, with only some parameter variation. Therefore, they often bring to the pool the same relevant documents. Finally, as in previous studies [2, 19, 20, 21, 22], to avoid buggy implementations of some of the systems that took part in the challenges, we tested again with only the top 75% of runs of each test collection.

The results for the *leave-one-run-out*, in addition to being less realistic as a model of real life, are also more conservative than those for *leave-one-organization-out*, such that in the following we shall discuss only the latter.

5.1 Correction results

In these settings, we explored the different value of the weight of the merging function α. In which we observe that as expected, as α goes from 1 to 0, the role of the new run on the runs in the pool decreases, to the point where, for $\alpha = 0$, the method no longer makes any change to the existing runs. This degree of change in α also affects the variability of ΔP and $\Delta\overline{P}$ which decreases as well due to the lower variation between the synthetic and the pooled runs. This effect grows linearly with α and in the following we shall report the maximum effects, obtained for $\alpha = 1$. Moreover, we tested the algorithm with different bias indicators (i.e. replacing λ with ΔP or $\Delta\overline{P}$, as discussed in Section 3.2), thus testing the presence and absence of information about relevant or non-relevant documents in the relevance assessment as potential flags to trigger bias correction. We consistently observe lower performance compared with λ.

Figure 3 shows the comparison, per test collection, of the three different approaches in the *leave-one-organization-out* experiment as a function of the Mean Absolute Error (MAE). As defined before in [23], the Mean Absolute Error is computed as the absolute difference between the scores of two runs, averaged over the set of topics. In addition to observing the error in the scores, it is also of interest to see how many rank reversals occur. The System Rank Error (SRE) is the sum of all the variation on rank of the system given the true rank. Figure 4 shows the SRE. For the experiments with the 75% top runs, actual values are reported in Table 2. In addition to these two measures, in Table 2, we also reported the SRE*, which only counts the variation on system rank when the difference among them is statistically significant in the ground truth (Tukey's test, $p < 0.05$ [9]).

In the table and plots we observe that our method, in a majority of cases outperforms the reduced pool and the Webber method. The last lines of Table 2 show how often each method outperformed both other methods (ties are not counted). It also shows how often it obtained the absolute worst score. It can be observed that, of the 675 tests summarized in Table 2, the proposed method obtains the worst performer mark exactly three times. On the other hand, the competing method is significantly more aggressive in its bias correction. In the majority of the cases it obtains worse system scores and rankings when compared to the simpler method of not doing anything (i.e. the reduced pool). This happens in particular in Ad Hoc 6, 7, 8 and Robust 2005. The proposed method is shown to be stable. Particularly important, it gets worse scores exactly once on SRE*, the metric measuring reversals among systems identified to be statistically significantly different. And, it happens with Medical 2011 and $P@100$, increasing the SRE* from 0 (for reduced pool) to 10, which reason should be found in the used shallow pool depth of 10.

5.2 Relation to test collection stability

The results observed in Figures 3 and 4, as well as in Table 2 lead us to question whether or not there is a connection between the effect of our method and the quality of the test collection. We therefore compare the percentage MAE change for each test collection and for each n of $P@n$, with the two coefficients of stability recently adapted by Urbano et al. [20] from Generalizability Theory: the Generalizability Coefficient ($E\rho^2$) and Dependability (Φ). $E\rho^2$ measures the stability based on system variance and the relative differences between systems; Φ measures the stability based on system variance and the absolute effectiveness scores. To infer that a test collection is reliable, both measures must tend to 1. Figure 2 shows the relation between these two factors and the change in MAE for $P@10$ and $P@100$ for

[1]The software is available on the website of the first author.

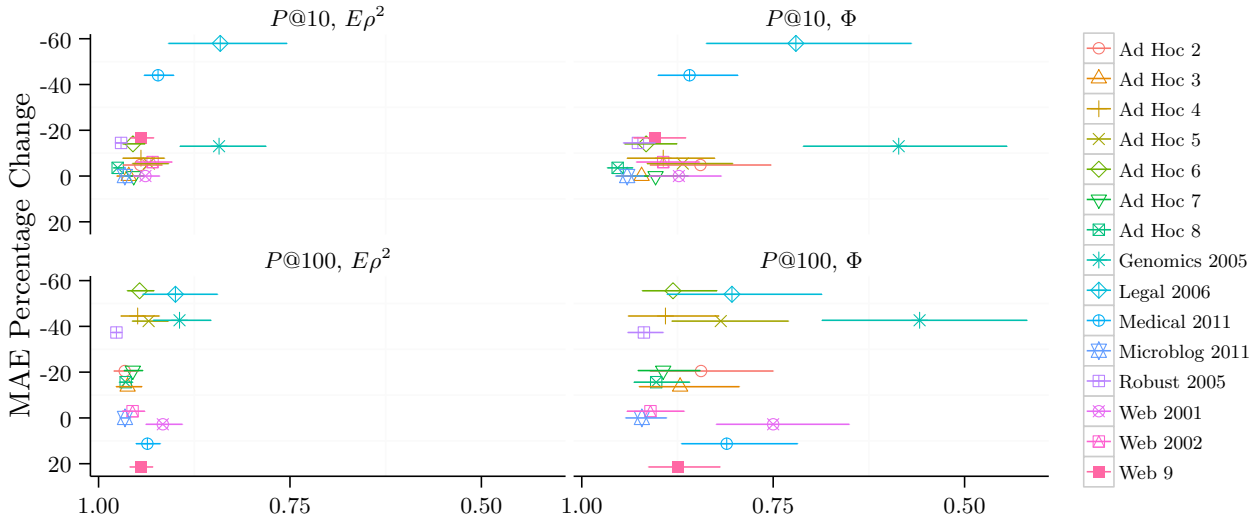

Figure 2: Plots of the percentage change of Mean Absolute Error for $P@10$ and $P@100$ against the coefficients of stability, Generalizability Coefficient ($E\rho^2$) and Dependability (Φ). Spearman's rank correlation for P@10: $E\rho^2$ is 0.48 (p>0.07) and for Φ is 0.36 (p>0.18). For P@100: $E\rho^2$ is 0.17 (p>0.54) and for Φ is 0.09 (p>0.74).

the 15 test collections studied here. This change in MAE is calculated between our method and the traditional, reduced pool method. In general, we observe a weak correlation with $E\rho^2$ and Φ (i.e. less error for more unstable test collections). With $P@10$ our method has a stronger effect with more unstable test collections. An interesting case happens at $P@100$, where for some test collection the MAE percentage change is positive, that is resulting in a lack of correlation, with which we get more ambiguous results.

To understand this, it is needed to understand that when the cut-off of $P@n$ is greater then the depth of the pool, we are essentially comparing with an uncertain ground truth, since also the large pool (the one with the runs of the organization we removed for testing) is affected by the presence of unjudged documents. This uncertainty in comparing the result when the depth of the pool is less then the considered $P@n$ needs to be considered when looking at these results, as well as those of all other proposed methods. The *depth* of each test collection is available for reference in Table 2.

6. CONCLUSION

The primary focus of this paper is an insight that information about the quality of an unpooled run can be obtained by observing its effect on existing, pooled runs. Such an effect is modeled by the creation of a synthetic run, obtained by merging the two runs—the pooled and the unpooled—in a very simple way, by linearly combining the ranks of each document in the old run (i.e. we do not want to add new documents to the old run, just observed how its own documents shift as a function of the information provided by the new run). The effect is measured with essentially three quantities: the change in the position of the judged relevant documents (measured via precision), the change in the position of the judged non-relevant documents (measured via anti-precision), and the change in the position of the unjudged documents (measured via a measure $\bar{k}$ we define). Observing these changes across the set of pooled runs—the effect of the new run on the existing runs—we identify a coefficient λ whose sign allows us to decide whether a bias correction should be made or not. We then proceed with the provision of a bias correction procedure based on the above three quantities, which we show to be conservative in the sense that it never damages significant rank orders, and only very rarely affects changes in system rankings. This is opposed to previous methods which are too aggressive in the bias correction, and in so being, add another level of uncertainty to the system rankings.

The proposed method addresses a significant concern coming from research but also from practice: the necessity to have a reliable, yet understandable metric, which we can communicate to partners outside of our community. This last condition significantly restricts our possible choices. Precision at cut-off is by far the most easily understood quantity to communicate and with this study we have shown that we can correct pool bias when considering a run that has not participated in the creation of the pool.

Acknowledgements

This research was supported by the Austrian Science Fund (FWF) project number P25905-N23 (ADmIRE).

7. REFERENCES

[1] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, B57(1), 1995.

[2] D. Bodoff and P. Li. Test theory for assessing ir test collections. In *Proc. of SIGIR*, 2007.

[3] C. Buckley, D. Dimmick, I. Soboroff, and E. Voorhees. Bias and the limits of pooling for large collections. *Inf. Ret.*, 10(6), 2007.

[4] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *Proc. of SIGIR*, 2004.

[5] S. Büttcher, C. L. A. Clarke, P. C. K. Yeung, and I. Soboroff. Reliable information retrieval evaluation

Table 2: Summary of the results per test collection generate trough a leave-one-organization-out using the top 75% of the pooled runs. With: $|R|$ number of runs submitted, $|O|$ number of organizations involved, $|R_p|$ number of pooled runs, d depth of the pool and $|T|$ number of topics.

Track		P@n	Ours			Webber			Reduced Pool		
			MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Ad Hoc 2	$\|R\|$: 38	5	0.0063	19	0	0.0069	18	0	0.0065	19	0
	$\|O\|$: 22	10	0.0082	32	0	0.0084	27	0	0.0085	32	0
	$\|R_p\|$: 36	20	0.0132	50	0	0.0119	48	0	0.0138	52	0
	d : 100	30	0.0149	51	0	0.0129	42	0	0.0164	56	0
	$\|T\|$: 50	100	0.0216	88	2	0.0169	61	2	0.0293	122	4
Ad Hoc 3	$\|R\|$: 40	5	0.0025	3	0	0.0038	3	0	0.0025	3	0
	$\|O\|$: 22	10	0.0024	5	0	0.0029	0	0	0.0025	5	0
	$\|R_p\|$: 26	20	0.0042	5	0	0.0041	6	0	0.0042	5	0
	d : 200	30	0.0049	13	0	0.0051	12	0	0.0051	13	0
	$\|T\|$: 50	100	0.0067	21	0	0.0071	28	0	0.0085	25	0
Ad Hoc 4	$\|R\|$: 33	5	0.0065	17	0	0.0069	20	0	0.0073	18	0
	$\|O\|$: 19	10	0.0081	23	0	0.0081	22	0	0.0093	25	0
	$\|R_p\|$: 32	20	0.0089	29	0	0.0096	33	0	0.0107	33	0
	d : 100	30	0.0089	27	0	0.0098	31	0	0.0117	32	0
	$\|T\|$: 50	100	0.0084	26	0	0.0115	35	0	0.0161	52	0
Ad Hoc 5	$\|R\|$: 61	5	0.0067	39	0	0.0073	39	0	0.0069	39	0
	$\|O\|$: 21	10	0.0070	50	0	0.0072	50	0	0.0074	50	0
	$\|R_p\|$: 61	20	0.0071	59	0	0.0077	61	0	0.0080	66	0
	d : 100	30	0.0068	61	0	0.0080	73	0	0.0085	77	0
	$\|T\|$: 50	100	0.0048	66	0	0.0086	116	0	0.0104	136	0
Ad Hoc 6	$\|R\|$: 74	5	0.0179	16	3	0.0308	28	5	0.0234	20	5
	$\|O\|$: 17	10	0.0224	15	6	0.0354	29	8	0.0283	22	11
	$\|R_p\|$: 19	20	0.0253	18	6	0.0355	35	12	0.0336	31	12
	d : 55	30	0.0263	20	6	0.0365	38	11	0.0389	31	11
	$\|T\|$: 50	100	0.0136	25	0	0.0186	34	4	0.0288	41	4
Ad Hoc 7	$\|R\|$: 103	5	0.0011	4	0	0.0018	4	0	0.0012	4	0
	$\|O\|$: 42	10	0.0017	8	0	0.0022	8	0	0.0017	8	0
	$\|R_p\|$: 79	20	0.0021	13	0	0.0027	18	0	0.0022	13	0
	d : 100	30	0.0022	24	0	0.0029	28	0	0.0025	27	0
	$\|T\|$: 50	100	0.0025	40	0	0.0038	58	0	0.0038	53	0
Ad Hoc 8	$\|R\|$: 129	5	0.0036	11	8	0.0042	11	8	0.0038	11	8
	$\|O\|$: 41	10	0.0036	7	5	0.0043	9	7	0.0038	9	7
	$\|R_p\|$: 80	20	0.0035	6	1	0.0042	14	2	0.0038	7	2
	d : 100	30	0.0035	7	1	0.0042	10	1	0.0038	8	2
	$\|T\|$: 50	100	0.0030	26	6	0.0049	42	8	0.0043	38	7
Web 9	$\|R\|$: 104	5	0.0023	15	0	0.0031	15	0	0.0023	15	0
	$\|O\|$: 23	10	0.0022	17	0	0.0027	19	0	0.0025	19	0
	$\|R_p\|$: 64	20	0.0027	25	0	0.0027	21	0	0.0030	26	0
	d : 100	30	0.0027	41	0	0.0026	34	0	0.0034	43	0
	$\|T\|$: 50	100	0.0043	105	0	0.0036	109	1	0.0052	147	3
Web 2001	$\|R\|$: 97	5	0.0010	5	0	0.0015	5	0	0.0010	5	0
	$\|O\|$: 29	10	0.0016	9	0	0.0021	9	0	0.0016	9	0
	$\|R_p\|$: 61	20	0.0017	14	0	0.0021	17	0	0.0019	14	0
	d : 100	30	0.0018	26	0	0.0021	15	0	0.0022	27	0
	$\|T\|$: 50	100	0.0040	96	0	0.0026	63	0	0.0041	87	0
Web 2002	$\|R\|$: 69	5	0.0038	54	0	0.0038	54	0	0.0038	54	0
	$\|O\|$: 16	10	0.0045	76	0	0.0040	80	0	0.0049	80	0
	$\|R_p\|$: 69	20	0.0041	78	0	0.0039	78	0	0.0051	95	0
	d : 50	30	0.0044	106	0	0.0036	87	0	0.0052	120	0
	$\|T\|$: 50	100	0.0039	138	0	0.0024	92	0	0.0038	136	0
Genomics 2005	$\|R\|$: 62	5	0.0052	64	0	0.0054	69	0	0.0057	69	0
	$\|O\|$: 32	10	0.0063	111	0	0.0061	110	0	0.0073	117	0
	$\|R_p\|$: 58	20	0.0077	89	0	0.0066	80	0	0.0088	106	0
	d : 60	30	0.0084	106	0	0.0069	81	0	0.0100	139	0
	$\|T\|$: 49	100	0.0055	93	0	0.0049	93	0	0.0091	197	0
Robust 2005	$\|R\|$: 74	5	0.0179	16	3	0.0308	28	5	0.0234	20	5
	$\|O\|$: 17	10	0.0224	15	6	0.0354	29	8	0.0283	22	11
	$\|R_p\|$: 19	20	0.0253	18	6	0.0355	35	12	0.0336	31	12
	d : 55	30	0.0263	20	6	0.0365	38	11	0.0389	31	11
	$\|T\|$: 50	100	0.0136	25	0	0.0186	34	4	0.0288	41	4
Legal 2006	$\|R\|$: 34	5	0.0593	80	0	0.1146	135	11	0.1327	136	11
	$\|O\|$: 8	10	0.0572	94	15	0.1097	138	25	0.1440	139	25
	$\|R_p\|$: 23	20	0.0306	64	12	0.0813	128	24	0.0984	139	33
	d : 10	30	0.0219	63	1	0.0636	117	17	0.0750	130	30
	$\|T\|$: 39	100	0.0063	54	17	0.0224	101	33	0.0250	107	39
Medical 2011	$\|R\|$: 127	5	0.0267	142	0	0.0309	159	0	0.0464	261	0
	$\|O\|$: 29	10	0.0265	157	0	0.0377	219	1	0.0586	336	5
	$\|R_p\|$: 56	20	0.0224	152	0	0.0209	142	0	0.0326	206	2
	d : 10	30	0.0201	153	0	0.0153	121	0	0.0229	174	0
	$\|T\|$: 34	100	0.0092	176	8	0.0054	97	0	0.0078	149	0
Microblog 2011	$\|R\|$: 184	5	0.0053	101	14	0.0056	101	14	0.0053	101	14
	$\|O\|$: 58	10	0.0060	183	57	0.0063	183	57	0.0062	183	57
	$\|R_p\|$: 119	20	0.0065	217	68	0.0065	217	64	0.0067	217	68
	d : 30	30	0.0069	272	77	0.0070	256	76	0.0076	294	95
	$\|T\|$: 49	100	0.0022	156	12	0.0022	143	13	0.0024	159	14
top performer			46	35	20	18	23	2	0	0	0
worst performer			2	3	1	26	18	1	44	33	12

with incomplete and biased judgements. In *Proc. of SIGIR*, 2007.

[6] B. Carterette. Robust test collections for retrieval evaluation. In *Proc. of SIGIR*, 2007.

[7] B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In *Proc. of SIGIR*, 2006.

[8] B. Carterette, V. Pavlu, E. Kanoulas, J. A. Aslam, and J. Allan. Evaluation over thousands of queries. In *Proc. of SIGIR*, 2008.

[9] B. A. Carterette. Multiple testing in statistical analysis of systems-based information retrieval experiments. *ACM Trans. Inf. Syst.*, 30(1), Mar. 2012.

[10] C. L. A. Clarke and M. D. Smucker. Time well spent. In *Proc. of IIiX*, 2014.

[11] G. V. Cormack, C. R. Palmer, and C. L. A. Clarke. Efficient construction of large test collections. In *Proc. of SIGIR*, 1998.

[12] C. Hauff and F. de Jong. Retrieval system evaluation: Automatic evaluation versus incomplete judgments. In *Proc. of SIGIR*, 2010.

[13] K. Kuriyama, N. Kando, T. Nozue, and K. Eguchi. Pooling for a large-scale test collection: An analysis of the search results from the first NTCIR workshop. *Inf. Ret.*, 5(1), 2002.

[14] W.-H. Lin and A. Hauptmann. Revisiting the Effect of Topic Set Size on Retrieval Error. In *Proc. of SIGIR*, 2005.

[15] A. Moffat, W. Webber, and J. Zobel. Strategic system comparisons via targeted relevance judgments. In *Proc. of SIGIR*, 2007.

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

[17] T. Sakai. Alternatives to bpref. In *Proc. of SIGIR*, 2007.

[18] T. Sakai and N. Kando. On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Inf. Ret.*, 11(5), 2008.

[19] M. Sanderson and J. Zobel. Information retrieval system evaluation: Effort, sensitivity, and reliability. In *Proc. of SIGIR*, 2005.

[20] J. Urbano, M. Marrero, and D. Martín. On the measurement of test collection reliability. In *Proc. of SIGIR*, 2013.

[21] E. M. Voorhees. Topic set size redux. In *Proc. of SIGIR*, 2009.

[22] E. M. Voorhees and C. Buckley. The effect of topic set size on retrieval experiment error. In *Proc. of SIGIR*, 2002.

[23] W. Webber and L. A. F. Park. Score adjustment for correction of pooling bias. In *Proc. of SIGIR*, 2009.

[24] E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Proc. of SIGIR*, 2006.

[25] E. Yilmaz, E. Kanoulas, and J. A. Aslam. A simple and efficient sampling method for estimating AP and NDCG. In *Proc. of SIGIR*, 2008.

[26] J. Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proc. of SIGIR*, 1998.

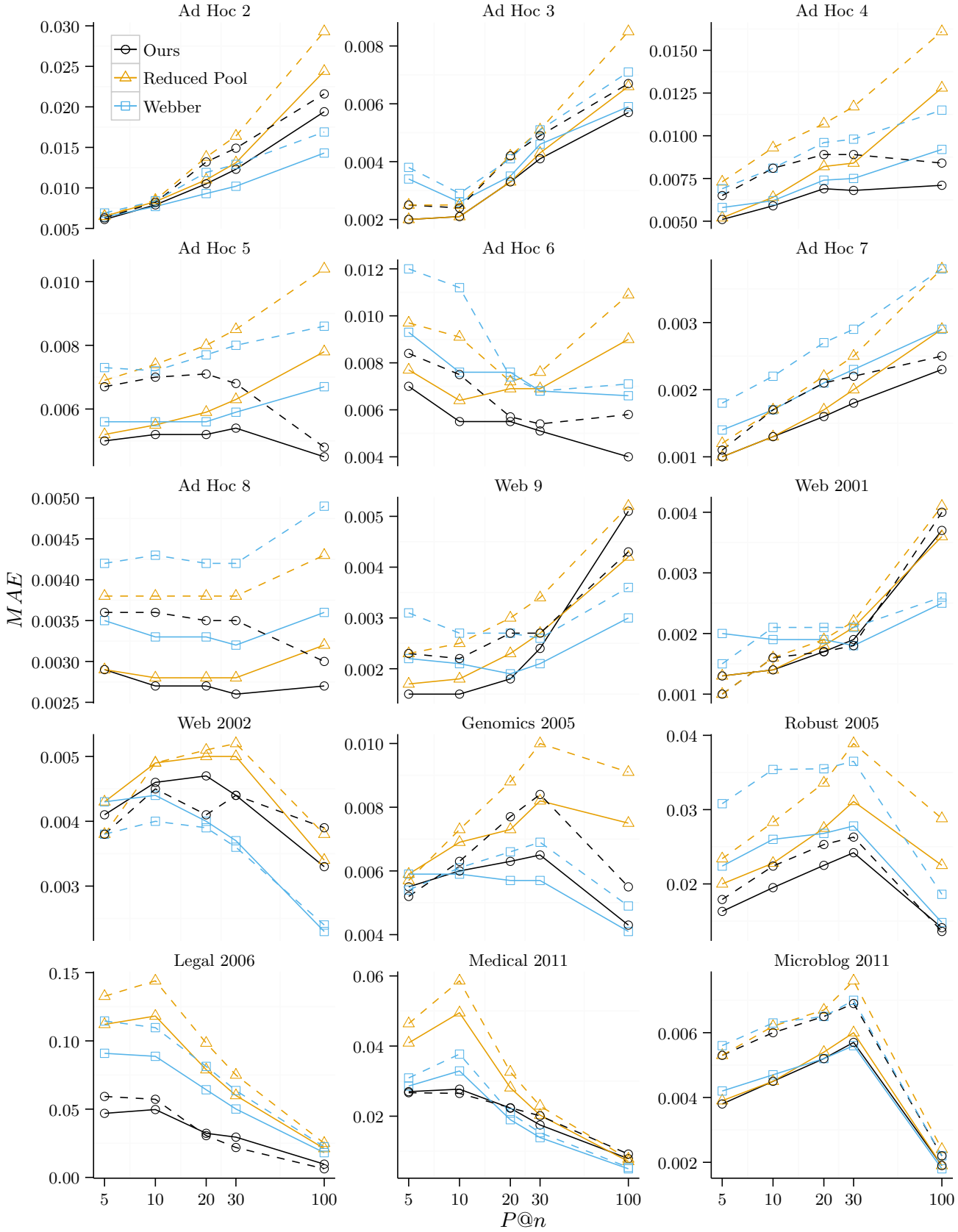

Figure 3: Plots per test collection of the Mean Absolute Error against the $P@n$ of the Reduced Pool and the two approaches, Ours and Webber, to correct pool bias. Generated using a leave-one-organization-out, using all the runs for the continuous lines and only the top 75% for the dashed lines. Our approach uses as indicator λ and $\alpha = 1$.

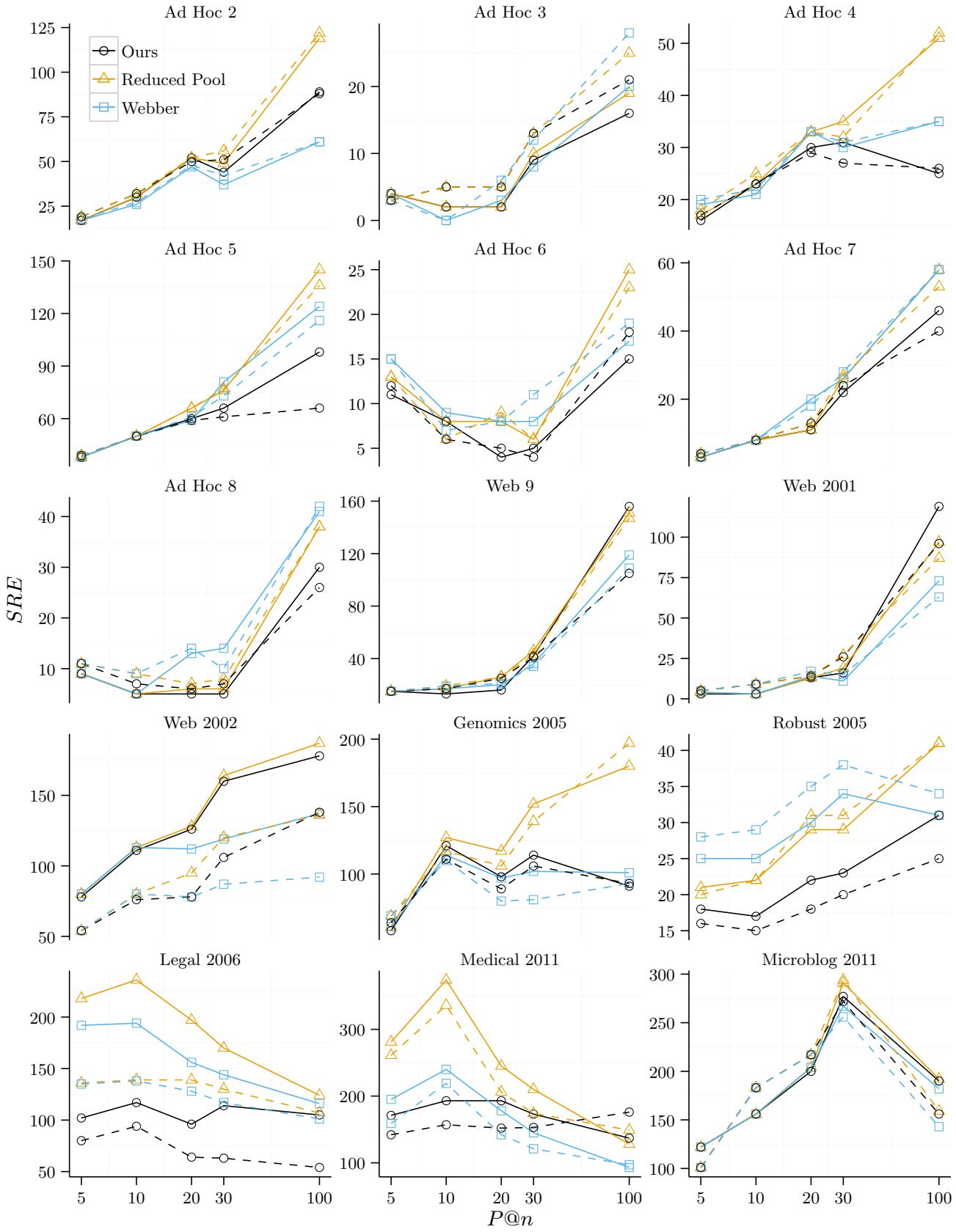

Figure 4: Plots per test collection of the System Rank Error against the $P@n$ of the Reduced Pool and the two approaches, Ours and Webber, to correct pool bias. Generated using a leave-one-organization-out, using all the runs for the continuous lines and only the top 75% for the dashed lines. Our approach uses as indicator λ and $\alpha = 1$.