The Curious Incidence of Bias Corrections in the Pool*

Aldo Lipani, Mihai Lupu, and Allan Hanbury

Institute of Software Technology and Interactive Systems (ISIS)
Vienna University of Technology, Austria
{surname}@ifs.tuwien.ac.at

Abstract. Recently, it has been discovered that it is possible to mitigate the Pool Bias of Precision at cut-off ($P@n$) when used with the fixed-depth pooling strategy, by measuring the effect of the tested run against the pooled runs. In this paper we extend this analysis and test the existing methods on different pooling strategies, simulated on a selection of 12 TREC test collections. We observe how the different methodologies to correct the pool bias behave, and provide guidelines about which pooling strategy should be chosen.

1 Introduction

An important issue in Information Retrieval (IR) is the offline evaluation of IR systems. Since the first Cranfield experiments in the 60s, the evaluation has been performed with the support of test collections. A test collection is composed of: a collection of documents, a set of topics, and a set of relevance assessments for each topic. Ideally, for each topic all the documents of the test collection should be judged, but due to the dimension of the collection of documents, and their exponential growth over the years, this praxis soon became impractical. Therefore, already early in the IR history, this problem has been addressed through the use of the pooling method [11]. The pooling method requires a set of runs provided by a set of IR systems having as input the collection of documents and the set of topics. Given these runs, the original pooling method consists, per topic, of: 1) collecting all the top n retrieved documents from each selected run in a so-called *pool* 2) generating relevance judgments for each document in the pool. The benefit of this method is a drastic reduction of the number of documents to be judged, quantity regulated via the number d of documents selected. The aim of the pooling method, as pointed out by Spärck Jones, is to find an unbiased sample of relevant documents [6]. The bias can be minimized via increasing either the number of topics, or the number of pooled documents, or the number and variety of IR systems involved in the process. But albeit the first two are controllable parameters that largely depend on the budget invested in the creation of the test collection, the third, the number and variety of the involved IR systems depends on the interest and participation of the IR community in the issued challenge.

In IR the need for more understandable metrics for practitioners has already been pointed out [7,4]. This led, on the one hand, to the development of new

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

evaluation measures that 'make sense' and, on the other hand, to step back and focus on simple metrics such as Precision. Additionally Precision at cut-off ($P@n$) is a cornerstone for more complex and sophisticated evaluation measures in IR. This is why this study focuses exclusively on $P@n$.

Herein, we study how the reduced pool and the two pool bias correctors [7,16] behave when used on different pool strategies and configurations. We measure the bias using the Mean Absolute Error (MAE) on three pooling strategies, fixed-depth pool, uniformly sampled pool, and stratified pool for various parameter values. We provide insights about the two pool bias corrector approaches.

The remainder of the paper is structured as follows: in Section 2 we provide a summary of the related work on pooling strategies and on pooling correction. In Section 3 we generalize the pooling strategies, look at the existing pool bias correction approaches, and analyze their properties relating them to the studied pooling strategies. Section 4 confirms the theoretical observations experimentally. Results are discussed in Section 5. We conclude in Section 6.

2 Related Work

This section is divided into two parts. First we consider the work done in correcting the pool bias for the evaluation measure $P@n$. Second we consider the work conducted on the pooling strategies themselves. We will not cover the extensive effort in creating new metrics that are less sensitive to the pool bias (the work done for Bpref [2], followed by the work done by Sakai on the condensed lists [9] or by Yilmaz et al. on the inferred metrics [18,17]).

2.1 Pool Bias Estimators

Webber and Park [16] attempted to correct the bias by computing the Mean Absolute Error (MAE) of each run when pooled and not pooled, for a given evaluation measure and test collection, to be added as correction. Their method follows the assumption that the scores produced by the runs are normally distributed, a probably incorrect but common assumption. Although the method was presented only on Rank-Biased Precision, they pointed out that similar results were obtained also with $P@n$.

We [7] attempted to correct the pool bias with a more complex algorithm that estimates the correction by measuring the effect of a tested run against the pooled runs. Our method makes use of information that comes from both non-relevant and non-judged documents. The method works under the assumption that, if the correction is triggered, the adjustment needed is proportional to the average gain of non-judged documents on the affected pooled runs.

2.2 Pooling Strategies

Pooling was already used in the first TREC, in 1992, 17 years after it was introduced by Spärck Jones and van Rijsbergen [11], on the discussion of building an 'ideal' test collection that would allow reusability. The algorithm [5] is described as follows: 1) divide each set of results into results for a given topic; then, for each topic: 2) select the top 200 (subsequently generalized to d) ranked documents of each run, for input to the pool; 3) merge results from all runs; 4) sort

results on document identifiers; 5) remove duplicate documents. This strategy is known as *fixed-depth pool*.

With the aim of further reducing the cost of building a test collection, Buckley and Voorhees [2] explored the uniformly sampled pool. At the time they observed that $P@n$ had the most rapid deterioration compared to a fully judged pool. The poor behavior of this strategy for top-heavy metrics was confirmed recently in Voorhees's [14] short comparison on pooling methods.

Another strategy is the stratified pool [18], a generalization of both the fixed-depth pool and the uniformly sampled pool. The stratified pool consists in layering the pool in different strata based on the highest rank obtained by a document in any of the given runs.

A comparison of the various pooling strategies has been recently reported by Voorhees [14]. We complement that report in several directions: First, and most importantly, we focus on bias correction methods and the effects of the pooling strategies on them rather than on the metrics themselves. Second, we generalize the stratified sampling method. Third, we expand the observations from 2 to 12 test collections. We also observe that the previous study does not distinguish between the effect of the number of documents evaluated with the effect of the different strategies (see Tab. 1 in [14]). In our generalization of the stratified pooling strategy we will ensure that the expected[1] number of judged documents is constant across different strategies.

3 Background Analysis

Here, the pooling method and its strategies are explained. Then the work conducted on the pool bias correction for the evaluation measure $P@n$ is analyzed. In this section, to simplify the notation, the average $P@n$ over the topics is denoted by g.

3.1 The Pooling Method

Three common strategies are used in the pooling method, listed in increasing order of generality: fixed-depth pool ($Depth@d$), uniformly sampled pool ($SampledDepth@d\&r$) and stratified ($Stratified$).

The simplest pooling strategy is $Depth@d$, which has been already described above. $SampledDepth@d\&r$ uses the $Depth@d$ algorithm as an intermediary step. It produces a new pool by sampling without replacement from the resulting set at a given rate r. Obviously, if $r = 1$ the two strategies are equivalent. The $Stratified$ further generalizes the pooling strategy, introducing the concepts of stratification and stratum. A stratification is a list of n strata, with sizes s_i and sample rates r_i: $z^n = [(s_0, r_0), ..., (s_n, r_n)]$. A stratum is a set of documents retrieved by a set of runs on a given range of rank positions. Which rank range ρ of the stratum j is: if $j = 1$ then $1 \leq \rho \leq s_1$ else if $j > 1$ then $\sum_{i=1}^{j-1} s_i < \rho \leq \sum_{i=1}^{j} s_i$. In this strategy, given a stratification z^n, we distinguish three phases: 1) pre-pooling: each document of each run is collected in a stratum based on its

[1] Obviously, a guarantee on the actual number of judged documents cannot be provided without an a posteriori change in the sampling rates.

rank; 2) purification: for each stratum all the documents found on a higher rank stratum get removed; 3) sampling: each stratum is sampled without replacement based on its sample rate. Obviously, when the stratification is composed by only one stratum, it boils down to *SampledDepth@d&r*.

Which strategy to choose is not clear and sometimes it depends on the domain of study. Generally, the *Depth@d* is preferred because of its widespread use in the IR community, but for recall oriented domains the $Stratified$ is preferred because of its ability to go deeper in the pool without explosively increasing the number of documents to be judged. The *SampledDepth@d&r* is generally neglected due to its lack in ability to confidently compare the performance of two systems, especially when used with top-heavy evaluation measures.

The main factor under the control of the test collection builder is the number of judged documents. This number depends both on the number of pooled runs and on the minimum number of judged documents per run. The following inequality shows the relation between these two components:

$$g(r, Q_{d+1}^{R_p}) - g(r, Q_d^{R_p}) \geq g(r, Q_{d+1}^{R_p \setminus \{r_p\}}) - g(r, Q_d^{R_p \setminus \{r_p\}}) \tag{1}$$

where r is a run, R_p is the set of runs used on the construction of the pool Q, $r_p \in R_p$, d is the minimum number of documents judged per run, and $g(r, Q)$ is the score of the run r evaluated on the pool Q. The proof is evident if we observe that: $Q_d^{R_p} \subseteq Q_{d+1}^{R_p}$, $Q_{d+1}^{R_p \setminus \{r_p\}} \subseteq Q_{d+1}^{R_p}$, $Q_d^{R_p \setminus \{r_p\}} \subseteq Q_{d+1}^{R_p \setminus \{r_p\}}$ and $Q_d^{R_p \setminus \{r_p\}} \subseteq Q_d^{R_p}$. When $r_p = r$, the inequality (Eq. 1) defines the *reduced pool* bias. In general however it shows that the bias is influenced by d, the minimum number of judged documents per run, and by $|R_p|$ the number of runs.

3.2 Pool Bias Correctors

Herein, we analyze the two pool bias correctors. Both attempt to calculate a coefficient of correction that is added to the biased score.

Webber and Park [16] present a method for the correction that computes the error introduced by the pooling method when one of the pooled runs is removed. This value is computed for each pooled run using a leave-one-out approach and then averaged and used as correction coefficient. Their correction coefficient for a run $r_s \notin R_p$ is the expectation:

$$\mathop{\mathrm{E}}_{r_p \in R_p} \left[g(r_p, Q_d^{R_p}) - g(r_p, Q_d^{R_p \setminus \{r_p\}}) \right] \tag{2}$$

where R_p is the set of pooled runs, $r_p \in R_p$ and $Q_d^{R_p}$ is a pool constructed with d documents per each run in R_p. As done in a previous study [7] we evaluate the method using the mean absolute error (MAE). Eq. 2 is simple enough that we can attempt to analytically observe how the method behaves with respect to the reduced pool, in the context of a *Depth@d* pool at varying d. We identify analytically a theoretical limitation of the Webber approach when used with a *Depth@d*. The maximum benefit, in expectation, is obtained when the cut-off value of the precision (n) is less or equal to d. After this threshold the benefit is lost.

We start analyzing the absolute error (AE) of the Webber approach for a run r_s:

$$\left| g(r_s, G) - \left[g(r_s, Q_d^{R_p}) + \mathop{\mathrm{E}}_{r_p \in R_p} \left[g(r_p, Q_d^{R_p}) - g(r_p, Q_d^{R_p \setminus \{r_p\}}) \right] \right] \right|$$

where G is ground truth[2], $Q_d^{R_p}$ is the pool constructed using a *Depth@d* strategy where d is its depth and R_p is the set of pooled runs. We compare it to the absolute error of the reduced pool:

$$\left| g(r_s, G) - g(r_s, Q_d^{R_p}) \right| \tag{3}$$

We observe that when the depth of the pool d becomes greater or equal than n, $g(r_p, Q_d^{R_p})$ becomes constant. For the sake of clarity we substitute it with C_n. We substitute $g(r_s, G)$, which is also a constant, with C_G. Finally, we also rename the components $a(d) = g(r_s, Q_d^{R_p})$, $b(d) = \mathrm{E}_{r_p \in R_p}[g(r_p, Q_d^{R_p \setminus \{r_p\}})]$, and call $f(d)$ the AE of the Webber method, and $h(d)$ the AE of the reduced pool:

$$f(d) = |C_G - [a(d) + C_n - b(d)]| \quad \text{and} \quad h(d) = |C_G - a(d)| \tag{4}$$

To study the behavior at varying of d, we define $\dot{g}$ as the finite difference of g with respect to d:

$$\dot{g}(r, Q_d^R) = g(r, Q_{d+1}^R) - g(r, Q_d^R) \tag{5}$$

We finitely differentiate the previous two equations, and since both are decreasing functions of d, to see where the margin between the two functions shrinks (the benefit decreases), it is sufficient to study when the inequality $\dot{f}(d) \geq \dot{g}(d)$ holds.

$$\dot{f}(d) = \begin{cases} -\dot{a}(d) + \dot{b}(d), & \text{if } C_G - [a(d) + C_n - b(d)] \geq 0 \\ \dot{a}(d) - \dot{b}(d), & \text{if } C_G - [a(d) + C_n - b(d)] < 0 \end{cases} \quad \text{and} \quad \dot{h}(d) = -\dot{a}(d)$$

Therefore,

$$\dot{f}(d) \geq \dot{h}(d) \text{ iff } \begin{cases} \dot{b}(d) \geq 0, & \text{if } C_G - [a(d) + C_n - b(d)] \geq 0 \\ 2\dot{a}(d) \geq \dot{b}(d), & \text{if } C_G - [a(d) + C_n - b(d)] < 0 \end{cases}$$

While the first condition is always verified ($\dot{b}(d)$ is an average of positive quantities), the second tells us that if $\dot{b}(d)$ is less or equal to $2\dot{a}(d)$ the Webber method decreases more slowly than the reduced pool. This inequality, as a function of r_s does not say anything about its behavior as it can be different for each r_s. Therefore we study the MAE using its expectation. We define R_G as the set of runs of the ground truth G, in which $R_p \subset R_G$. Using the law of total expectation we can write:

$$\begin{aligned} \mathop{\mathrm{E}}_{r_s \in R_G}[\dot{b}(d)] &= \mathop{\mathrm{E}}_{r_s \in R_G}[\mathop{\mathrm{E}}_{r_p \in R_G \setminus \{r_s\}}[g(r_p, Q_{d+1}^{R_G \setminus \{r_s, r_p\}}) - g(r_p, Q_d^{R_G \setminus \{r_s, r_p\}})]] = \\ &= \mathop{\mathrm{E}}_{r_{s_1}, r_{s_2} \in R_G : r_{s_1} \neq r_{s_2}}[g(r_{s_1}, Q_{d+1}^{R_G \setminus \{r_{s_1}, r_{s_2}\}}) - g(r_{s_1}, Q_d^{R_G \setminus \{r_{s_1}, r_{s_2}\}})] \end{aligned} \tag{6}$$

Using the pool inequality in Eq. 1:

[2] The ground truth is the pool using the maximum depth available in the test collection

$$\underset{r_{s_1}, r_{s_2} \in R_G : r_{s_1} \neq r_{s_2}}{\mathrm{E}} [g(r_{s_1}, Q_{d+1}^{R_G \setminus \{r_{s_1}, r_{s_2}\}}) - g(r_{s_1}, Q_d^{R_G \setminus \{r_{s_1}, r_{s_2}\}})] \leq$$
$$\leq \underset{r_{s_1} \in R_G}{\mathrm{E}} [g(r_{s_1}, Q_{d+1}^{R_G \setminus \{r_{s_1}\}}) - g(r_{s_1}, Q_d^{R_G \setminus \{r_{s_1}\}})] = \underset{r_s \in R_G}{\mathrm{E}} [\dot{a}(d)] \leq \underset{r_s \in R_G}{\mathrm{E}} [2\dot{a}(d)] \tag{7}$$

Therefore, in expectation, at increasing of depth of the pool d, for $P@n$ with $n \geq d$, the MAE of the Webber approach decreases more slowly than the MAE of the reduced pool.

We [7] introduced a method that attempts to correct the bias by measuring the effect of a run on the pooled runs, in terms of difference between the number of relevant, non-judged, and non-relevant documents. This information, averaged among all the pooled runs, in combination with the measurements made on the run itself, is used, first as a trigger to perform the correction and second as correction. Each pooled run r_p is effected using a merging function that averages the rank of all the shared documents between r_p itself and the selected run r_s, then uses the resulting average as a score to create a new reordered run r'_p. The trigger function is as follows, where ΔP_{r_s} and $\Delta \overline{P}_{r_s}$ is the average gain in precision and anti-precision (ratio of non-relevant documents) of the affected pooled runs.

$$\lambda = \bar{k}_{r_s} (\Delta P_{r_s} \overline{P}_{r_s} - \Delta \overline{P}_{r_s} P_{r_s})$$

For $\lambda > 0$ the correction is triggered, and the following correction added, where $\Delta \bar{k}_{r_s}$ is the gain on ratio of non-judged documents over n of the modified pooled runs, and $\bar{k}_{r_s}$ is the ratio of non-judged documents over n in the run to correct:

$$\bar{k}_{r_s} \cdot \max(\Delta \bar{k}_{r_s}, 0)$$

We observe that there exists a confounding factor that is the proportion of judged relevant to non-relevant documents. Assuming that all runs are ranked by some probability of relevance, i.e. that there is a higher probability to find relevant documents at the top than at the bottom of the runs, our approach (Lipani) is sensitive to the depth of the pool because at any one moment it compares one run, that is a set of d probably relevant documents and $|r_s| - d$ probably non-relevant documents with all the existing runs, i.e. a set of $d|R_p|$ probably relevant documents and $(\mathrm{E}[|r_p|] - d)|R_p|$ probably non-relevant documents. The effects of this aggregation are difficult to formalize in terms of the proportion of relevant and non-relevant documents, and we explore them experimentally in the next section.

4 Experiments

To observe how the pool and the two pool bias correctors work in different contexts we used a set of 12 TREC test collections, sampled from different tracks: 8 from Ad Hoc, 2 from Web, 1 from Robust and 1 from Genomics. In order to make possible the simulation of the different pooling strategies, the test collections needed to have been built using a *Depth*@d strategy with depth $d \geq 50$.

For *Depth*@d and *SampledDepth*@d&r, all the possible combinations of parameters with a step size of 10 have been explored. Fig. 1 shows the MAE of the

different methods, for $Depth@d$ at varying d. Fig. 2 shows the MAE of the different methods for the $SampledDepth@d\&r$, with fixed depth $d = 50$, at varying sample rate r from 10% to 90% in steps of 10.

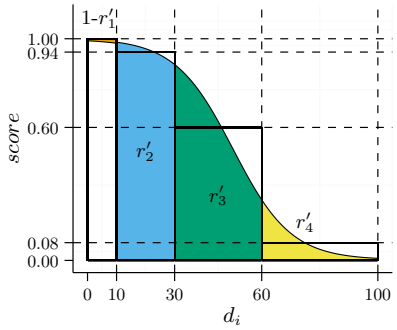
For $Stratified$, due to its more flexible nature, we constrained the generation of the stratifications. We should note that there are practically no guidelines in the literature on how to define the strata. First, we defined the sizes of the strata for each possible stratification and then for each stratification we defined the sample rates of each stratum.

Given n, the number of strata to generate, and $s \in S$, a possible stratum's size, we find all the vectors of size n, $s^n = (s_0\, s_1 \ldots s_n)$ such that $\sum_{i=1}^{n} s_i = D$, where D is the maximum depth of the pool available, with sizes s_i chosen in increasing order, except for the last stratum, which may be a residual smaller than the second-last. For each $n \in \mathbb{N}^+$, and constraining the set of stratum sizes S to multiples of 10, when $D = 100$, we find only ten possible solutions.

To find the sample rates r_i to associate to each stratum s_i, we followed a more elaborated procedure. As pointed out by Voorhees [14], the best results are obtained fixing the sample rate of the first stratum to 100%. From the second to the last stratum, when available, we sample keeping the expected minimal number of pooled documents for each run constant. This is done in order to allow a cross-comparison among the stratifications. However, for stratifications composed by 3 or more strata some other constraint is required. The TREC practice has shown that the sampling rate decreases fast, but so far decisions in this sense are very ad-hoc. Trying to understand how fast the rate should drop, we are led back to studies relating retrieval status values (RSV), i.e. scores, with probabilities of relevance. Intuitively, we would want our sampling rate to be related to the latter. Nottelmann and Fuhr [8] pointed out that mapping the RSV to the probability of relevance using a logistic function outperforms the mapping when a linear function is used. Therefore, to create the sampling rates, we define a logistic function with parameters $b_1 = 10/D$, $b_0 = D/2$ where D is the depth of the original pool (i.e. of the ground truth). b_1 defines the slope of the logistic function and is in this case arbitrary. b_0 is the minimal number of documents we want, on expectation, to assess per run. The sample rates are then the areas under the logistic curve for each stratum (Eq. 9). However, since, to keep in line with practice, we always force the first strata to sample at 100%, we correct the remaining sampling rates proportionally (Eq. 10). To verify that the expected minimal number of sampled documents is b_0, it is enough to observe that the sum of the areas that define the sampling rates is b_0 (Eq. 8). The resulting stratifications are listed in Table 1 and the corresponding MAEs for the different methods are shown in Fig. 3.

To measure the bias of the reduced pool and the two correcting approaches, we run a simulation[3] using only the pooled runs and a leave-one-organization-out approach, as done in previous studies [7,3]. The leave-one-organization-out approach consists in rebuilding the pool removing in sequence all the runs submitted by an organization. Finally as performed in previous studies [7,13,15,10,1,12],

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



$$\int_0^D \frac{1}{1+e^{b_1(x-b_0)}}dx = b_0 \quad (8)$$

$$r'_n = \int_{\sum_{i=1}^{n-1} s_i}^{\sum_{i=1}^{n} s_i} \frac{1}{1+e^{b_1(x-b_0)}}dx \quad (9)$$

$$r_n = \frac{1}{s_n}\left[r'_n - (1-r'_1)\frac{r'_n}{\sum_{i=2}^{N} r'_i}\right] \quad (10)$$

Table 1. List of the used stratifications z_i^n, where n is the size of the stratification, and i is the index of the solution found given the fixed constraints. E[d] is the mean number of judged documents per run for all the test collections with respect to z_1^1. † indicates when the difference with respect to the previous stratification is statistically significant (t-test, $p < 0.05$).

n	i	z_i^n	$(s_1, r_1)...(s_n, r_n)$	E[d]
1	1	z_1^1	(100, 50)	50.00
2	1	z_1^2	(10, 100) (90, 44)	50.75†
	2	z_2^2	(20, 100) (80, 38)	52.11†
	3	z_3^2	(30, 100) (70, 29)	52.29†
	4	z_4^2	(40, 100) (60, 17)	52.45

n	i	z_i^n	$(s_1, r_1)...(s_n, r_n)$	E[d]
3	1	z_1^3	(10, 100) (20, 94) (70, 30)	51.71†
	2	z_2^3	(10, 100) (30, 90) (60, 22)	52.29†
	3	z_3^3	(10, 100) (40, 83) (50, 14)	52.32
	4	z_4^3	(20, 100) (30, 77) (50, 14)	52.37
4	1	z_1^4	(10, 100) (20, 94) (30, 60) (40, 8)	52.25

to avoid buggy implementations of systems, the bottom 25% of poorly performing runs are removed for each test collection.

5 Discussion

In case of *Depth@d* in Fig. 1 we observe that, as expected given the analytical observations of Section 3.2, the Webber approach slows down its correction with increasing depth d. The ratio between the error produced by the reduced pool and the method decreases systematically after d becomes greater than the cut-off value n of *P@n*. This trend sometimes leads to an inversion, as for Ad Hoc 2, 3, 6, 7 and 8, Web 9 and 10 and Robust 14. The Lipani approach, as expected, is less reliable when the depth d of the pool is less than the cut-off value of the precision. We see that very clearly in Web 9 and Web 11. It generally reaches a peak when d and n are equal, and then improves again. Comparing the two approaches, we see that in the majority of the cases the Lipani approach does better than the Webber approach.

The *SampledDepth@d&r*, shown in Fig. 2, does not display the same effects observed for the *Depth@d*. Both corrections do better than the reduced pool. The effect in Webber's method disappears because in this case (and also in the *Stratified* pool later) the constant C_n in Eq. 4 is no longer a constant here, even for $n > d$. The effect observed in Lipani's method is removed by the more non-relevant documents introduced on the top of the pooled runs, which reduce the effect of the selected run. The Lipani approach does generally better, sometimes with high margin as in: Ad Hoc 5 and 8, Web 9, 10 and 11, and Genomics 14.

Finally, in the *Stratified* case, Fig. 3, the effects observed on the *Depth@d* are also not visible. For *P*@10, the corrections perform much better if we sample more from the top, most notably for the stratifications of size 3, but the correc-

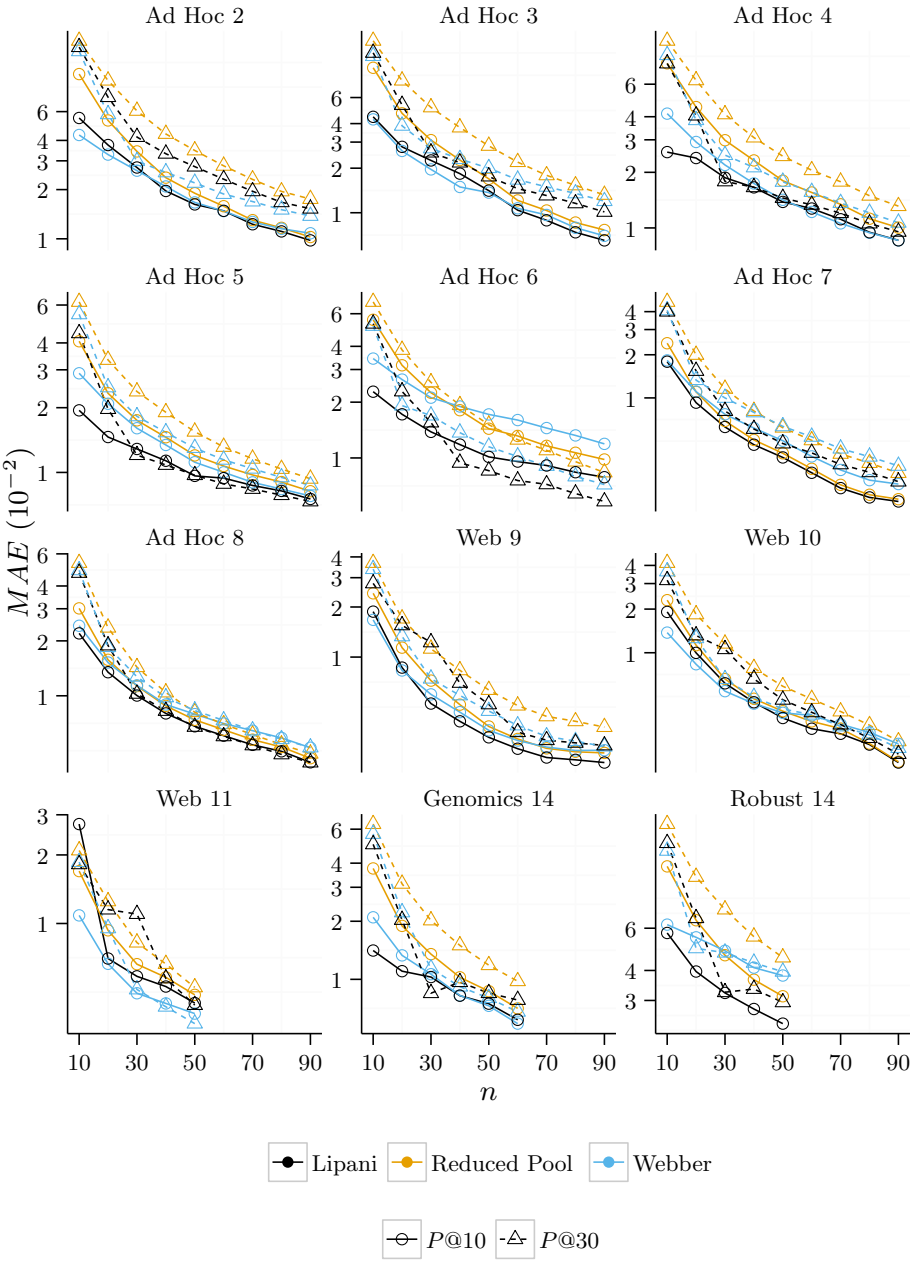


Fig. 1. MAE in logarithm scale of the ground truth (Depth@M, where M is the maximum depth of the test collection) against the *Depth@d* pool at varying of the depth n, for the evaluation measures, P@10 and P@30. MAE computed using the leave-one-organization out approach of pooled runs and removing the bottom 25% poorly performing runs.

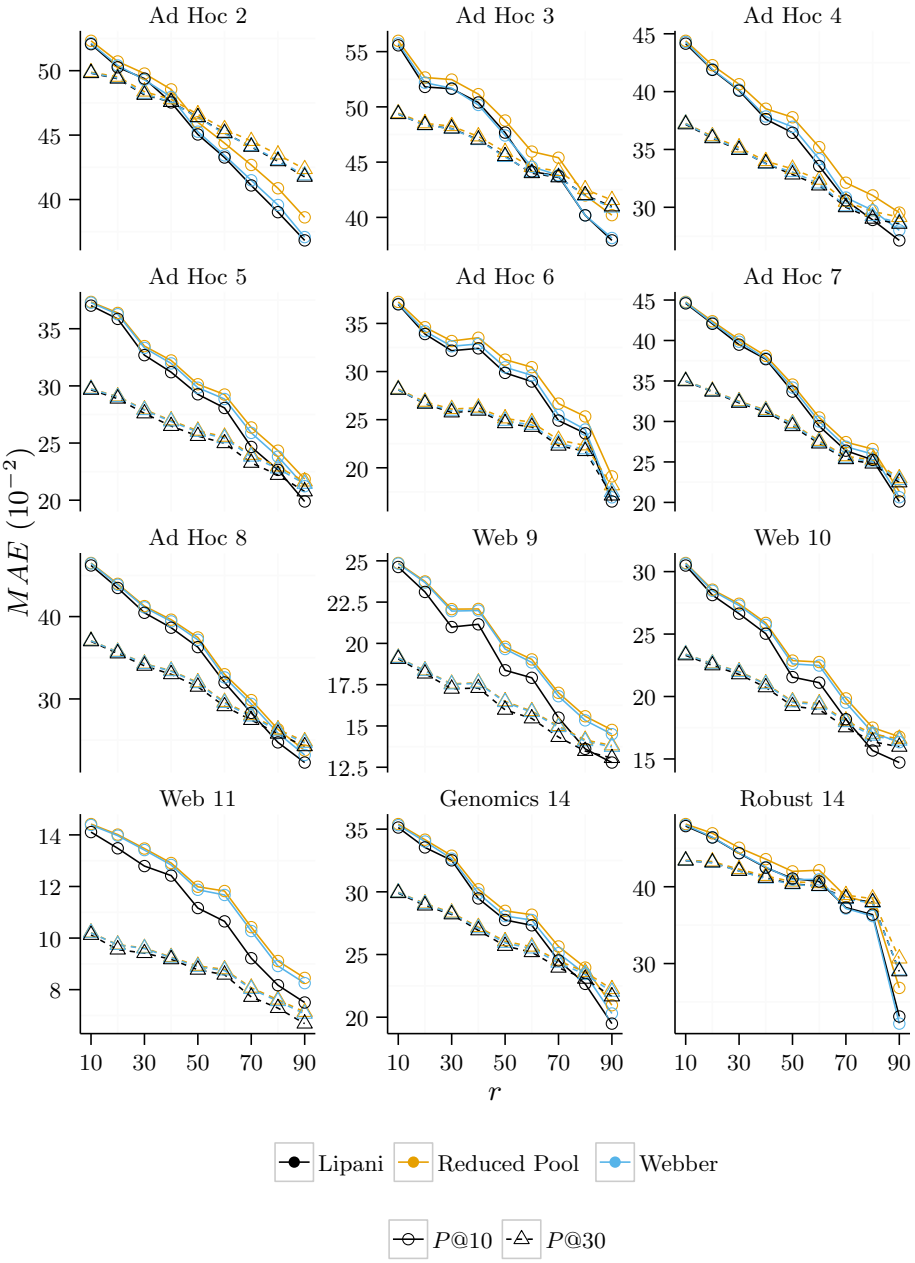


Fig. 2. MAE of the ground truth (Depth@M, where M is the maximum depth of the test collection) against the *SampledDepth*@d&r with fixed $d = 50$ at varying of the sample rate r, for the evaluation measures, P@10 and P@30. MAE computed using the leave-one-organization out approach of pooled runs and removing the bottom 25% poorly performing runs.

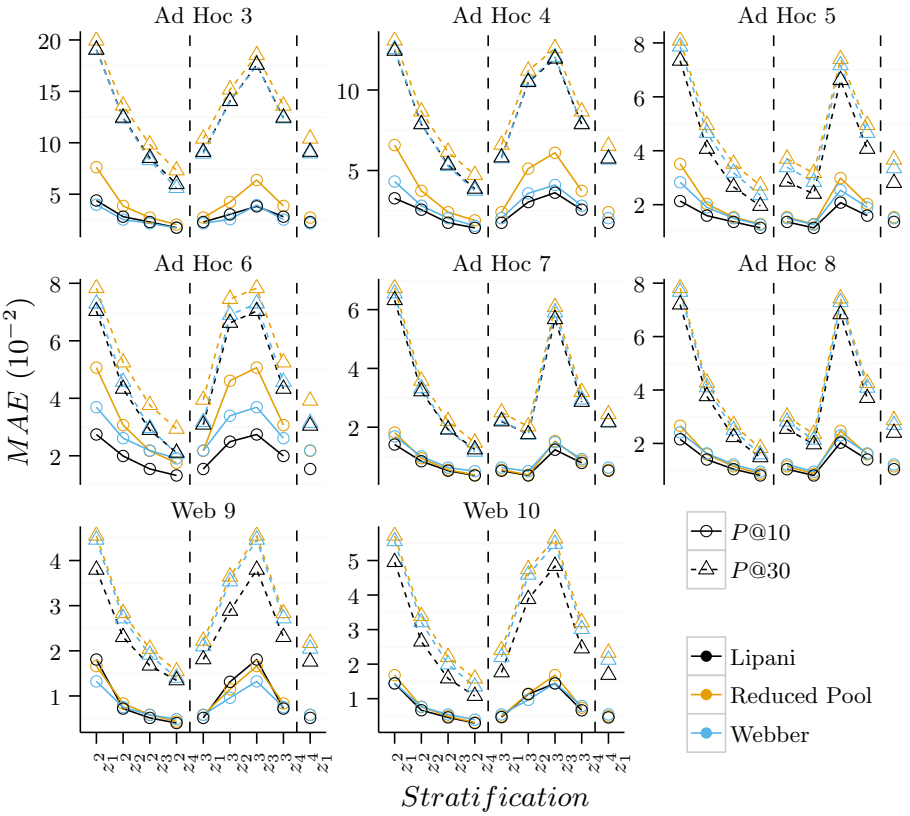


Fig. 3. MAE of the ground truth (*Depth*@M, where M is the maximum depth of the test collection) against the *Stratified* pool at varying of the different stratifications, for the evaluation measures, P@10 and P@30, only on test collections originally built using the *Depth*@100 pooling strategy. MAE computed using the leave-one-organization out approach of pooled runs and removing the bottom 25% poorly performing runs.

tion degrades when using P@30. This is particularly visible when comparing z_3^3 and z_4^3. Although they have essentially the same number of judged documents (with difference non statistically significant, Tab. 1), the stratification with a deeper first stratum makes a big difference in performance. Comparing the best stratification of size 2 (z_4^2) and the best stratification of size 3 (z_1^3) we observe that there is only a small difference in performance between them that could be justified by the smaller number of judged documents (with difference statistically significant, Tab. 1). The z_4^2 is the best overall stratification, confirming also the conclusion of Voorhees [14]. However a cheaper solution is the z_1^3, which, as shown in Tab. 1, evaluates fewer documents, but obtains a comparably low MAE.

Cross-comparing the three pooling strategies (observe the ranges on the y-scales), we see that the best performing strategy, fixing the number of judged documents, is *Depth*@d, then the *Stratified* and *SampledDepth*@d&r.

6 Conclusion

We have confirmed a previous study [7] that the Lipani approach to pool bias correction outperforms the Webber approach. We have further expanded these observations to various pooling strategies. We have also partially confirmed an-

other previous study indicating that *Stratified* pooling with a heavy top is preferable [14]. We have extended this by showing that, in terms of pool bias, the pooling strategies are, in order of performance, *Depth@d*, *Stratified*, and *SampledDepth@d&r*. Additionally, we made two significant observations on the two existing pool bias correction methods. We have shown, analytically and experimentally, that the Webber approach reduces its ability to correct the runs at increasing pool depth, when this is greater than the cut-off of the measured precision. Conversely, the Lipani approach sometimes manifests an instability when the depth of the pool is smaller the cut-off of the measured precision. These opposite behaviors would make the Lipani estimator a better choice since it improves with increasing number of judged documents. Both of these side-effects are reduced when a sampled strategy is used.

References

1. David Bodoff and Pu Li. Test theory for assessing ir test collections. In *Proc. of SIGIR*, 2007.
2. Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *Proc. of SIGIR*, 2004.
3. Stefan Büttcher, Charles L. A. Clarke, Peter C. K. Yeung, and Ian Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. In *Proc. of SIGIR*, 2007.
4. Charles L. A. Clarke and Mark D. Smucker. Time well spent. In *Proc. of IIiX*, 2014.
5. Donna Harman. Overview of the first trec conference. In *Proc. of SIGIR*, 1993.
6. Karen Sparck Jones. Letter to the editor. *Information Processing & Management*, 39(1), 2003.
7. Aldo Lipani, Mihai Lupu, and Allan Hanbury. Splitting water: Precision and anti-precision to reduce pool bias. In *Proc. of SIGIR*, 2015.
8. Henrik Nottelmann and Norbert Fuhr. From retrieval status values to probabilities of relevance for advanced ir applications. *Information Retrieval*, 6(3-4), 2003.
9. Tetsuya Sakai. Alternatives to bpref. In *Proc. of SIGIR*, 2007.
10. Mark Sanderson and Justin Zobel. Information retrieval system evaluation: Effort, sensitivity, and reliability. In *Proc. of SIGIR*, 2005.
11. K. Spärck Jones and C. J. van Rijsbergen. Report on the need for and provision of an" ideal" information retrieval test collection. *British Library Research and Development Report No. 5266*, 1975.
12. Julián Urbano, Mónica Marrero, and Diego Martín. On the measurement of test collection reliability. In *Proc. of SIGIR*, 2013.
13. Ellen M. Voorhees. Topic set size redux. In *Proc. of SIGIR*, 2009.
14. Ellen M. Voorhees. The effect of sampling strategy on inferred measures. In *Proc. of SIGIR*, 2014.
15. Ellen M. Voorhees and Chris Buckley. The effect of topic set size on retrieval experiment error. In *Proc. of SIGIR*, 2002.
16. William Webber and Laurence A. F. Park. Score adjustment for correction of pooling bias. In *Proc. SIGIR*, 2009.
17. Emine Yilmaz and Javed A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Proc. of CIKM*, 2006.
18. Emine Yilmaz, Evangelos Kanoulas, and Javed A. Aslam. A simple and efficient sampling method for estimating ap and ndcg. In *Proc. of SIGIR*, 2008.