Fixed-Cost Pooling Strategies

Aldo Lipani, David E. Losada, Guido Zuccon, Mihai Lupu

Abstract—The empirical nature of Information Retrieval (IR) mandates strong experimental practices. A keystone of such experimental practices is the Cranfield evaluation paradigm. Within this paradigm, the collection of relevance judgments has been the subject of intense scientific investigation. This is because, on one hand, consistent, precise, and numerous judgements are keys to reducing evaluation uncertainty and test collection bias; on the other hand, however, relevance judgements are costly to collect. The selection of which documents to judge for relevance, known as *pooling method*, has therefore a great impact on IR evaluation. In this paper we focus on the bias introduced by the pooling method, known as *pool bias*, which affects the reusability of test collections, in particular when building test collections with a limited budget. In this paper we formalize and evaluate a set of 22 pooling strategies based on: traditional strategies, voting systems, retrieval fusion methods, evaluation measures, and multi-armed bandit models. To do this we run a large-scale evaluation by considering a set of 9 standard TREC test collections, in which we show that the choice of the pooling strategy has significant effects on the cost needed to obtain an unbiased test collection. We also identify the least biased pooling strategy in terms of pool bias according to three IR evaluation measures: AP, NDCG, and P@10.

Index Terms—Pooling Method, Test Collections, Pool Bias.

1 Introduction

The effectiveness of an IR system is evaluated with the use of test collections. A test collection consists of a collection of documents, a set of topics, and a set of relevance judgments, which express the relevance relationship between topics and documents.

This set of relevance judgments is, in the vast majority of cases, by necessity a very small subset of the Cartesian product between the set of documents and the set of topics. If we were to consider even a relatively small test collection, with 500,000 documents and 50 topics (this is approximately the size of the Ad Hoc 8 test collection [1]), the total relevance judgments to be made would be 5×10^6. At a rate of 120 seconds/judgment, this represents the equivalent of 95 years of work for one person [2]. Therefore, since the very beginning of standardized IR benchmarking at the Text REtrieval Conference (TREC) in the early 1990s, *"pooling"* has been used to reduce the number of judgments, while still preserving the ability of the benchmark to distinguish between two or more retrieval systems [3].

The typical organizational process of an evaluation exercise (in Figure 1) goes as follows: After having identified a retrieval issue, the organizers of this exercise define a collection of documents and a set of topics (c and $\mathcal{Q}$). These two sets are then given to a number of participating organizations ($\mathcal{O}$), which, after having developed their IR systems, return to the organizers a series of search results (rs). Out of the union set of rs over all organizations ($\mathcal{R}$), a subset is selected, the pooled set ($\mathcal{R}_p$). This pooled set is then used as input of the pooling method (J), which together with human assessors, generates a set of relevance judgments ($\mathcal{J}$). In this process many decisions need to be taken when defining c, $\mathcal{Q}$, $\mathcal{R}_p$ and J, which give rise to important research questions [4]. In this paper, we focus on the decision of the pooling method J.

Pooling fundamentally relies on the assumption that if many and sufficiently diverse systems participate in a pool (i.e., having provided lists of documents they consider to be relevant for each topic), a set of judgements can be identified that, once adjudicated, can provide, using an effectiveness measure, a score that is predictive of the future relative performance of two or more systems. The original pooling method, now referred to as Depth@K, was first proposed in 1975 by Spärck Jones and van Rijsbergen [5], and first used when TREC started in 1991 [6].

The Depth@K strategy aggregates, for every topic, the top K documents returned by each system, and presents only this set to the human assessor(s) for evaluation. While the pooling method was introduced with the objective of finding as many relevant documents as possible (under the assumption that if a document is not retrieved by any system, it is probably irrelevant for the topic), the realistic objective is in fact to produce an *unbiased sample of the set of relevant documents* [7].

Since the proposal of the Depth@K pooling strategy, substantial research effort has gone into improving the evaluation procedures, reducing the associated costs, mitigating the effect of biases, and devising alternative pooling strategies (e.g. [8], [9], [10], [11], [12], [13], [14], [15], [16]).

Since the early days of pooling, it has been observed that, in the absence of sufficiently numerous and diverse systems, there is a risk that the identified set of relevant documents will be so limited that future systems, retrieving a new set of relevant (but at this point unjudged) documents, will be considered ineffective because they do not primarily find the set of relevant documents found by the systems that were originally pooled [17]. Incomplete judgments, i.e., the presence among the retrieved results of unjudged documents, have little impact on the small newswire collections used in early TREC years; however, they do lead to uncertainty in the evaluation quality on larger, web-size collections, thus rendering evaluation on these collections biased [18], [19].

This bias, named *pool bias*, manifests with *the effect that documents that were not selected in the pool created from the original retrieval systems will never be considered relevant* [20]. In the following we provide a formal definition of pool bias:

Definition 1. The pool bias $b(r, J_{\mathcal{R}_p})$ is a systematic error we observe when performing a measurement with an evaluation measure f on a run r using the pooled documents resulting from a pooling strategy J with input a set of pooled runs $\mathcal{R}_p$, which may or may not contain information about the run r:

$$b(r, J_{\mathcal{R}_p}) = f(r, J_{\mathcal{R}_p}) - f(r, \mathcal{I}) \quad (1)$$

where $\mathcal{I}$ is the *ideal set of judgments* one would obtain when evaluating the entire collection of documents. We say that $f(r, \mathcal{I})$ is the *true measurement* and $f(r, J_{\mathcal{R}_p})$ is the *biased measurement*.

The bias measured using f on the set of judgments ($\mathcal{J}$) is inversely proportional to the cost of $\mathcal{J}$, which is equal to the number of judgments times a unit cost of judgment (u):

$$b(r, J_{\mathcal{R}_p}) \propto C(J_{\mathcal{R}_p})^{-1}, \quad C(J_{\mathcal{R}_p}) = |J_{\mathcal{R}_p}| \cdot u. \quad (2)$$

To minimize this bias, we can either increase the number of judgments, hence the cost of $\mathcal{J}$, or improve J.

The research effort in this area has channeled in two main directions. On one hand, prior work has attempted to reduce bias at test collection build time by considering different pooling strategies [9], [10], [21]. On the other hand, for already existing test collections, some studies have adopted measures that reduce the effect of the bias [11], [12], [13]. Sometimes, the two directions intertwine, and a new pooling strategy is proposed together with a matching evaluation measure [22], but that significantly restricts the future use of the collection to specific measures.

In this paper, we focus on reducing this pool bias at test collection build time, exploring different pooling strategies to identify the most efficient way to create the pool, while controlling the bias. We focus on a specific case of pooling: when the pool has to respect a financial constraint (budget) that limits the number of documents to be pooled to a fixed value (N documents). We call this *fixed-cost pooling*. Moreover, these N documents to be judged should be distributed fairly across topics (equally divided when possible). Both are typical constraints in most IR evaluation exercises like TREC, CLEF and NTCIR. While a number of isolated studies have analyzed and proposed a number of pooling strategies, a complete picture of their effectiveness and bias is still lacking, and little has been analyzed about these strategies in the context of fixed-cost pooling. This article extends and complements the body of evidence regarding pooling by providing: a synthesis of a substantial line of research done on the pooling method; a coherent mathematical framework to describe pooling strategies; the identification of theoretical similarities between the analyzed strategies; and, a large-scale evaluation using 9 test collections. Based on this, we provide guidelines for building more stable test collections. In addition to the traditional Depth@K pooling strategy, we analyze the pool bias of a set of 22 previously identified pooling strategies.

The remainder of this article is structured as follows. Section 2 defines the notation used throughout the paper. Then, Section 3 presents the pooling strategies. Experiments are reported in Section 4, and discussed in Section 5. Finally, we conclude in Section 6.

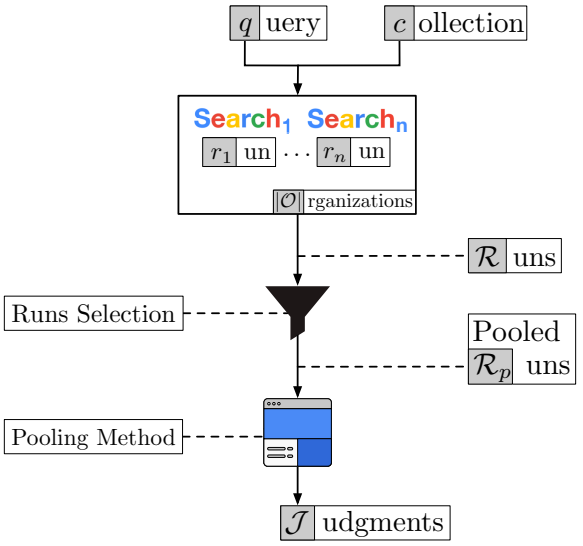

Fig. 1. The typical organizational process of an evaluation exercise.

2 Notation

In the following table we present the notation used throughout the paper. The table includes a set of symbols, functions and operators used to express operations in a compact way.

Symbols	
$\mathcal{Q}$	Set of topics.
q	A topic $q \in \mathcal{Q}$.
$\mathcal{R}$	Set of runs.
$\mathcal{R}_p$	Set of pooled runs $\mathcal{R}_p \subseteq \mathcal{R}$.
r	A run $r \in \mathcal{R}$.
$\mathcal{O}$	Set of organizations.
o	An organization $o \in \mathcal{O}$.
$\mathcal{R}_o$	Set of runs submitted by o.
$\mathcal{D}$	Collection of documents.
d	A document $d \in \mathcal{D}$.
$\mathcal{D}_r$	Set of documents retrieved by r.
$\mathcal{J}$	Set of pooled documents ($\mathcal{J} = \mathcal{J}^+ \cup \mathcal{J}^-$ and mutually exclusive).
$\mathcal{J}^+$	Set of relevant documents $\mathcal{J}^+ \subseteq \mathcal{J}$.
$\mathcal{J}^-$	Set of irrelevant documents $\mathcal{J}^- \subseteq \mathcal{J}$.
ϵ	A small number $\ll \min_{r \in \mathcal{R}_p}(\lvert r \rvert^{-1})$.

Functions	
[condition]	Returns 1 if the condition within the brackets is verified, 0 otherwise (aka Iverson bracket).
$\tau@n(\mathcal{R}, s)$	Returns the union of the top N documents retrieved by the set of runs $\mathcal{R}$ ordered by the function s.
$\rho(d, r)$	Returns the *rank* at which d has been retrieved in r if $d \in \mathcal{D}_r$, the highest rank possible otherwise ($\lvert\mathcal{D}\rvert$).
$\sigma(d, r)$	Returns the *score* at which d has been retrieved in r if $d \in \mathcal{D}_r$, the lowest score returned by r otherwise.
$\mu(a, b)$	Returns a random number in $[a, b]$.
$\mathrm{id}(r)$	Returns a natural number $n \in \mathbb{N}$ unique for every $r \in \mathcal{R} : 1 \leq n \leq \lvert\mathcal{R}\rvert$.

Sequences	
$a\lvert_{n_0}^{n_1}$	Set of elements of the sequence a_n from n_0 to n_1, $\{a_i\}_{n_0 \leq i \leq n_1}$.
$\mathrm{Avg}(a\lvert_{n_0}^{n_1})$	Average of the sequence a_n for the values from n_0 to n_1.
$\mathrm{Var}(a\lvert_{n_0}^{n_1})$	Variance of the sequence a_n for the values from n_0 to n_1.

3 Pooling Strategies

We examine each of the pooling strategies that we empirically investigate in this article as alternative to the standard Depth@K strategy. As mentioned in the introduction, in this paper we are mainly concerned with pools formed by exactly N documents, but the strategies may be further generalized to variable-size pools (e.g., by implementing different stopping criteria; this is left for future work). Moreover, the fair distribution of the N documents across topics by equally dividing them, may be also further generalized to variable-size topic pools (e.g., by implementing different topic allocation strategies, this is also left for future work).

In what follows, we make the effort to unify all the pooling strategies under the same mathematical framework in order to be able to formally assess their similarities and differences. In this framework we define each pooling strategy as a set-building function (J) that outputs a set of pooled documents, given as input a set of runs and a scoring function (s) used to score all candidate documents. Each pooling strategy is identifiable by the properties of J and s.

These pooling strategies can be classified in different ways. In this paper we have chosen to do it in two ways, by their type and by their origin. The classification based on type arises naturally from the mathematical formalization of the pooling strategies. This classification by type has led to the definition of the following classes: *non-adaptive*, *adaptive with run allocation*, and *adaptive without run allocation*. Where, in general, by adaptive we refer to those pooling strategies that adapt their behavior based on knowledge acquired in the previous selection step(s), and by non-adaptive we refer to those pooling strategies that do not adapt. By *adaptive with run allocation* we refer to an adaptive strategy that, to select the next document, it first selects a run – allocates a judgment to be performed to this run – then selects a document from this run. *Adaptive without run allocation* refers to an adaptive strategy that selects a document by aggregating information across runs. About the classification by origin, the reason of this choice is twofold. First, because this classification allows us to distinguish between classic pooling strategies [9] and the more recent multi-armed bandit based strategies [15]. Second, because this classification, since some of the pooling strategies are relatively new to IR, allows us to recall their underlying intuitions that have been extensively investigated in IR, i.e., as retrieval fusion methods [23], [24], [25] or as IR evaluation measures [16]. This classification by origin has led to the definition of the following classes: *classic pooling*, *voting systems*, *retrieval fusion methods*, *IR evaluation measures*, and *multi-armed bandit models*.

3.1 Non-Adaptive Pooling Strategies

Non-adaptive pooling strategies do not modify their behavior based on the current pooled documents, regardless of whether these documents have been judged or not. The strategy Depth@K belongs to this category. The following subsections group the pooling strategies by their origin: *classic pooling*, *voting systems*, *retrieval fusion methods*, and *IR evaluation measures*.

3.1.1 Classic Strategies

Before analyzing the considered pooling strategies, we start presenting the formalization of the most common strategy: Depth@K. Then, we present some natural variants, Take@N and FairTake@N, which consider the number of required documents (N) as a parameter.

Depth@K (D). This strategy creates, for each topic, a global ranked list of documents where each document is scored based on its highest position across all $\mathcal{R}_p$ runs. Given this ranked list, the top ranked documents are selected to form the pool. The Depth@K strategy is specified by the following definitions of s, which scores every document d retrieved by the set of pooled runs $\mathcal{R}_p$:

$$s(d, \mathcal{R}_p) = \max_{r \in \mathcal{R}_p} (-\rho(d, r)) \tag{3}$$

and J, which determines the set of pooled documents:

$$J_{\mathcal{R}_p} = \{d \in r : r \in \mathcal{R}_p,\ s(d, \mathcal{R}_p) \geq -K\} \tag{4}$$

A primary feature of this pooling strategy is its *fairness* to the pooled runs. A strategy is *fair* when the probability of a document to be judged at a given position is constant across runs. This is guarantied by selecting the top K documents from every run. However, although this pooling strategy takes into consideration the contribution of all pooled runs, it has no control on the exact size of the final set of pooled documents ($|\mathcal{J}|$). It is therefore not a fixed-cost pooling strategy. Moreover, the number of documents selected per topic can vary depending on the size of the overlapping retrieved documents the pooled runs share on a per topic basis.

Lipani et al. [26] have introduced a natural extension of Depth@K that guarantees a given number N of pooled documents, called Take@N, effectively turning Depth@K into a fixed-cost pooling strategy. We now formalize this strategy, show its limitation, and introduce a new version that addresses it.

Take@N (T). This strategy creates, for each query, a global ranked list of documents using a new definition of s:

$$s(d, \mathcal{R}_p) = \max_{r \in \mathcal{R}_p} (-\rho(d, r) - \epsilon \cdot \mathrm{id}(r)) \tag{5}$$

This definition of s is similar to the definition in Eq. 3. However, it differs by the small deterministic contribution ($\epsilon \cdot \mathrm{id}(r)$) that is used to provide a unique score for every d in order to break ties. This contribution is small enough to not change the order defined by the document's ranks, and it is deterministic because it is based on the ids of the runs. The top n ranked documents, fraction of the size of the pool N, are selected to be pooled as follows:

$$J_{\mathcal{R}_p} = \tau@n(\mathcal{R}_p, s) \tag{6}$$

where $\tau@n$ is always well defined, i.e. there is no ambiguity on which documents to return first. Compared to Depth@K, this strategy presents a drawback: it does not guarantee fairness with the pooled runs. With Depth@K all runs contribute equally to the pool (first K documents). With Take@N, instead, not all runs may contribute the same. The contributions are however only slightly unbalanced: the maximum difference between the number of documents contributed by two runs is equal to one. This strategy

also compared with Depth@K behaves differently across topics, because while Depth@K can vary based on the size of the overlapping retrieved documents the pooled runs share, Take@N distributes the N documents to be judged uniformly. That is, to every topic is assigned, if possible, the same fraction of documents to be judged ($n \cdot |\mathcal{Q}| = N$).

FairTake@N (F). This strategy aims to address the lack of fairness of Take@N by introducing a non-deterministic selection of the documents to be judged. This strategy shares some of the characteristics of the *Stratified* pooling strategy [22]. The *Stratified* strategy defines multiple strata, each characterized by a depth and a sample rate. This strategy is defined in two steps. First, each document is assigned to a stratum based on its highest rank across the pooled runs. Then, documents are sampled based on the sample rate of the stratum. FairTake@N is akin to having a stratification composed of two strata, a stratification with sample rate 1 as deep as the number of documents to be judged $n_{q,0}$ does not exceed n_q, the fraction of documents to be judged assigned to the topic q, and a second stratification of depth 1 with sample rate equal to $(n_q - n_{q,0})/|\mathcal{R}_p|$, which guaranties eventually to have exactly n_q judged documents. By definition, this strategy is fair with the pooled runs because any document at a given position has the same probability to be judged. In this strategy s is defined as:

$$s(d, \mathcal{R}_p) = \max_{r \in \mathcal{R}_p} (-\rho(d, r) - \epsilon \cdot \mu(0, 1)) \tag{7}$$

J is defined as in Eq. 6. Fairness is achieved by introducing a small random component to the score s. This value breaks potential ties and is small enough to not influence the ranking. Its random nature ensures that any document has equal opportunity to be sampled from any run. In this way, the strategy selects n_q documents to be judged in a fair way because every run will have in expectation (across topics) the same number of judged documents.

3.1.2 Voting System-Based Strategies

These strategies are based on the intuitions underlying voting systems. In general, voting systems take one of two forms: (1) positional voting systems that rely on the rank at which a document is retrieved (e.g. to assign a voting score to that document), and (2) majority voting systems that assign document weights based on pairwise comparisons between candidate documents.

BordaTake@N (B). This strategy is a positional voting strategy in which candidate documents are ranked in order of preference. For this strategy, s is defined as:

$$s(d, \mathcal{R}_p) = \sum_{r \in \mathcal{R}_p} B(d, r) + \epsilon \cdot \mu(0, 1) \tag{8}$$

where B defines the particular implementation of the Borda count. In this case, because we are dealing with truncated ballots (i.e. not every document is ranked by each run), we follow the method also used by Aslam and Montague [23]: Therefore, for a document d, the strategy assigns a score equal to the size of the collection of documents ($|\mathcal{D}|$) minus the rank at which d has been retrieved in the r ($\rho(d, r)$) if d has been retrieved by r, or else, if d has not been retrieved by r, the average score the strategy would have assigned to the documents retrieved between the last ranked document (equal to the size of the run $|r|$) and the size of the collection of documents ($|\mathcal{D}|$). Formally, B is defined as follows:

$$B(d, r) = \begin{cases} |\mathcal{D}| - \rho(d, r) & \text{if } d \in \mathcal{D}_r \\ \text{Avg}_{|r| < n \leq |\mathcal{D}|} \left(|\mathcal{D}| - n\right) & \text{if } d \notin \mathcal{D}_r \end{cases} \simeq$$

$$\simeq - \begin{cases} \rho(d, r) & \text{if } d \in \mathcal{D}_r \\ \frac{|\mathcal{D}| + |r| + 1}{2} & \text{if } d \notin \mathcal{D}_r \end{cases} \tag{9}$$

where the symbol $\simeq$ indicates rank equivalence, and the expression on the right side of $\simeq$ is a simplified rank equivalent form of the same strategy. J is defined as in Eq. 6. Comparing this equation with Eq. 5 we observe that BordaTake@N is different from Take@N in that it considers the sum of all ranks at which a document has been retrieved, while Take@N only considers the highest rank (the earliest rank).

CondorcetTake@N (C). This majority voting strategy ensures that pooled documents are those that, when compared to not-pooled documents, have been retrieved at higher ranks by more systems. Strategies that fulfill this condition satisfy the *Condorcet criterion*, and it is easy to prove that Depth@K, Take@N, FairTake@N and BordaTake@N do not satisfy this condition. Specifically, this strategy starts by forming a list containing the set of all documents retrieved by the pooled systems. Then, it sorts the list according to the following procedure. Each document pair d_i and d_j is then compared as follows. We iterate through the document rankings of each system and increment a counter if d_i is ranked above d_j (or decrement the counter in the converse situation). When all systems have been considered, if the counter is positive, then d_i should be ranked above d_j; if it is negative, then the opposite ranking should be enforced. This leads to the definition of the following comparative function:

$$C(d_0, d_1, \mathcal{R}_p) = \sum_{r \in \mathcal{R}_p} \text{sign}(\rho(d_1, r) - \rho(d_0, r)) \tag{10}$$

This function does not define a total order, leading to the so-called Condorcet paradox. Imagine three documents, d_a, d_b, and d_c, such that d_a is preferred over d_b, d_b over d_c, and d_c over d_a. This cycle is a paradox because the conclusions are in conflict with each other. A solution is to adopt a method that still respects the Condorcet condition but that does not fall in this paradox. In our case we use what is known as Copeland's method, which counts the number of times a document beats the other documents. This leads to the following definition of s:

$$s(d, \mathcal{R}_p) = \sum_{d' \in \mathcal{D}} \begin{cases} 1 & C(d, d', \mathcal{R}_p) > 0 \\ 0 & \text{otherwise} \end{cases} + \epsilon \cdot \mu(0, 1) \simeq$$

$$\simeq \sum_{d' \in \bigcup_{r \in \mathcal{R}_p} \mathcal{D}_r} \begin{cases} 1 & C(d, d', \mathcal{R}_p) > 0 \\ 0 & \text{otherwise} \end{cases} + \epsilon \cdot \mu(0, 1) \tag{11}$$

where the expression on the right side is a simplified rank equivalent form of the same strategy. This strategy is related to the Borda voting system. It can be proven that a relaxation of the Condorcet criterion used in the Copeland method leads to the Borda strategy (see Electronic Appendix 1). This observation illustrates why this method is majority-based. It

only counts when a document in the majority of the cases, across runs, has a higher score than another document, rather than counting its contribution per each individual run, like in BordaTake@N.

3.1.3 Retrieval Fusion Method-Based Strategies

Another class of non-adaptive pooling strategies is based on retrieval fusion methods. The main difference with the other strategies is that these are based on the score each ranker gives to a document (rather than the rank). To allow the comparison of scores between runs, score normalization is required, otherwise the pooling strategy would be biased towards the runs that produce larger scores. Following existing practice in fusion for retrieval [23], [27], [28], [29], we apply the following feature scaling:

$$\overline{\sigma}(d,r) = \frac{\sigma(d,r) - \min_{d' \in \mathcal{D}_r}(\sigma(d',r))}{\max_{d' \in \mathcal{D}_r}(\sigma(d',r)) - \min_{d' \in \mathcal{D}_r}(\sigma(d',r))} \quad (12)$$

which normalizes all the values into the range $[0,1]$. To be noted that for any document not retrieved by the run r by the definition of σ, which returns the minimum value observed in the run, $\overline{\sigma}$ returns 0.

CombMAXTake@N (MAX). This strategy assigns to each document the maximum retrieval score that the document has across all systems. In general, a document may be retrieved by multiple systems, and this likely happens with different scores. s is therefore defined as:

$$s(d,\mathcal{R}_p) = \max_{r \in \mathcal{R}_p} (\overline{\sigma}(d,r)) + \epsilon \cdot \mu(0,1) \simeq \\ \simeq \max_{r \in \mathcal{R}_p : d \in \mathcal{D}_r} (\overline{\sigma}(d,r)) + \epsilon \cdot \mu(0,1) \quad (13)$$

where on the right side of $\simeq$ we can observe a simplified rank equivalent form of the same strategy. After constructing a new document ranking with the maximum scores, the pool is obtained as for FairTake@N, i.e., only the documents with the highest n_q scores are included in the pool $\mathcal{J}$, where n_q is the fraction of documents to be judged assigned to the topic, as in the definition of J in Eq. 6. The *CombMAX* retrieval fusion method, which shares the same underlying intuition of CombMAXTake@N, is a commonly used strong baseline in the literature of fusion methods for retrieval. This strategy minimizes the probability to discover relevant documents being poorly ranked. This definition of s and the definition in Eq. 5 are similar, while the former uses documents' ranks, the latter documents' scores.

CombMINTake@N (MIN). While the previous strategy minimizes the probability to discover relevant documents being poorly ranked, this strategy minimizes the probability to discover irrelevant documents ranked at early ranks. This strategy also combines the scores from different runs (by extracting the minimum score of each document across all runs). s is therefore defined as:

$$s(d,\mathcal{R}_p) = \min_{r \in \mathcal{R}_p} (\overline{\sigma}(d,r)) + \epsilon \cdot \mu(0,1) \quad (14)$$

J is defined as in Eq. 6.

CombMEDTake@N (MED). This strategy takes a middle-ground approach to the selection of pooling documents based on fusion, by selecting the median score (as opposed to the maximum or minimum score as in CombMAXTake@N and CombMINTake@N, respectively). s is defined as follows:

$$s(d,\mathcal{R}_p) = \operatorname*{Med}_{r \in \mathcal{R}_p} (\overline{\sigma}(d,r)) + \epsilon \cdot \mu(0,1) \quad (15)$$

J is defined as in Eq. 6.

CombSUMTake@N (SUM). Instead of selecting a single score as in CombMAXTake@N, CombMINTake@N, and CombMEDTake@N, CombSUMTake@N sums all the available document's scores. s is therefore defined as:

$$s(d,\mathcal{R}_p) = \sum_{r \in \mathcal{R}_p} \overline{\sigma}(d,r) + \epsilon \cdot \mu(0,1) \simeq \\ \simeq \operatorname*{Avg}_{r \in \mathcal{R}_p} (\overline{\sigma}(d,r)) + \epsilon \cdot \mu(0,1) \simeq \\ \simeq \sum_{r \in \mathcal{R}_p : d \in \mathcal{D}_r} \overline{\sigma}(d,r) + \epsilon \cdot \mu(0,1) \quad (16)$$

where we observe that the expression on the right side of the first $\simeq$ demonstrates its rank equivalence of this strategy with a strategy defined by the arithmetic mean across runs, because differing only by a constant ($1/|\mathcal{R}_p|$); and the expression on the right side of the second $\simeq$ presents a simplified rank equivalence form of the same strategy. Comparing this equation with Eq. 8, we observe that CombSUMTake@N is the counterpart of the Borda strategy, but for scores (Borda uses ranks). J is defined as in Eq. 6.

CombANZTake@N (ANZ). This strategy computes the average of the non-zero document scores. This strategy effectively eliminates the effect of a single run failing to retrieve a document (and thus assigning a zero score to that document). s is therefore defined as:

$$s(d,\mathcal{R}_p) = \frac{1}{|\{r \in \mathcal{R}_p : \overline{\sigma}(d,r) > 0\}|} \sum_{r \in \mathcal{R}_p} \overline{\sigma}(d,r) + \epsilon \cdot \mu(0,1) \quad (17)$$

J is defined as in Eq. 6.

CombMNZTake@N (MNZ). This strategy aims to give higher weights to documents retrieved by multiple systems. This is achieved by multiplying the sum of scores of a document by the number of runs that retrieved that document. s is defined as:

$$s(d,\mathcal{R}_p) = |\{r \in \mathcal{R}_p : \overline{\sigma}(d,r) > 0\}| \sum_{r \in \mathcal{R}_p} \overline{\sigma}(d,r) + \epsilon \cdot \mu(0,1) \quad (18)$$

J is defined as in Eq. 6.

3.1.4 IR Evaluation Measure-Based Strategies

This section presents several strategies inspired by IR evaluation measures. These pooling strategies accumulate evidence of the importance of a document d for a given topic based on both a) the rank $\rho(d,r)$ at which d has been retrieved in the pooled run $r \in \mathcal{R}_p$, and b) the specific characteristics of the considered IR evaluation measure.

All the pooling strategies below share the same generalization of s, in which the contribution from every rank is replaced by a gain function related to the evaluation measure. s is defined as follows:

$$s(d,\mathcal{R}_p) = \sum_{r \in \mathcal{R}_p : d \in \mathcal{D}_r} G(\rho(d,r)) + \epsilon \cdot \mu(0,1) \quad (19)$$

where G is the gain defined by the evaluation measure. To simplify the notation, in the following $\rho(d, r)$ will be simply denoted by ρ.

DCGTake@N (DCG). This strategy uses the gain function defined in the discounted cumulative gain (DCG) to rank candidate documents [30]. The gain is characterized by an inverse log_2 decay function, as follows:

$$G(\rho) = \frac{1}{\log_2(\rho + 1)} \tag{20}$$

Candidate documents are ranked in decreasing order of the sum of values computed by G in s. J is defined as in Eq. 6.

RRFTake@N (RRF). This strategy is rooted in the reciprocal rank (RR) evaluation measure, which is commonly used to assess system effectiveness in tasks such as known item search, question answering, or query auto completion [31]. A variant of RR, the reciprocal rank fusion (RRF), has been used as retrieval fusion method [32]. RRF makes use of an additional parameter, α, that controls the decay of the document contribution score as a function of the rank. In this pooling strategy we employ the same idea, with $\alpha = 60$ as in Cormack et al. [32]; other values will be investigated in future work. Its G is defined as follows:

$$G(\rho) = \frac{1}{\rho + \alpha} \tag{21}$$

Candidate documents are ranked in decreasing order of the sum of values computed by G in s. $J_{\mathcal{R}_p}$ is defined as in Eq. 6.

PPTake@N (PP). This strategy (*PP*, for *perfect precision*) is inspired by the family of measures that count the number of relevant documents found at rank ρ and divide it by the number of documents up to rank ρ. Average Precision [33] and Sakai's Q-Measure [34] are examples of metrics belonging to this family. To define the G function for this class of IR evaluation measures, we assume to compute these IR evaluation measures on a ranked list as if all documents up to rank ρ are relevant, therefore the rank score attributed to a document retrieved by runs in $\mathcal{R}_p$ is the number of runs that have retrieved that document:

$$G(\rho) = 1 \tag{22}$$

This leads to a set-based majority voting procedure to rank documents and select the top n_q. It is set-based because the order in which the documents are retrieved does not count. This can be seen as a relaxation of the Borda strategy. Candidate documents are ranked in decreasing order of the sum of values computed by G in s. J is defined as in Eq. 6.

RBPTake@N (RBP). This strategy computes document scores based on Rank Biased Precision (RBP) [35]. The RBP formula is characterized by a parameter p that models the user persistence, i.e. the likelihood that the user examines a document. The persistence parameter is effectively used to discount the contribution of a relevant document, similarly to other gain-discount based measures [36]. The gain function is defined as follows:

$$G(\rho) = (1 - p)p^{\rho - 1} \tag{23}$$

In our experiments we use $p = 0.8$; this is akin to previous work that relied on RBP for evaluation [37], [38] and for pooling [14], [21]. The use of RBP as a document discount factor in weighting the contribution of documents to the pool creates a family of 3 pooling strategies [21], one being RBPTake@N. We present the other two in the next subsection. Candidate documents are ranked in decreasing order of the sum of values computed by G in s. J is defined as in Eq. 6.

3.2 Adaptive Pooling Strategies

So far we have discussed the non-adaptive pooling strategies. These strategies are characterized by first computing a score for each candidate document, ranking the documents decreasingly, and selecting the top n_q, where n_q is the fraction of documents to be judged assigned to the topic. They are non-adaptive because the score of a document is not affected by the previously selected documents.

Another class of pooling strategies are adaptive. These recompute the scores used by the ranking function s based on the last document selected. This is formalized by having s taking as input the current set of pooled documents and iteratively changing the scores of the documents.

First, the definition of s is expanded to consider the documents that have already been pooled. The superscript $\mathcal{J}$ indicates that we now receive the pooled documents as an input. The new definition of s, which will be denoted as $s_+^{\mathcal{J}}$, ensures that documents that have been pooled in the previous iteration are not re-scored:

$$s_+^{\mathcal{J}}(d, \mathcal{R}_p) = \begin{cases} s^{\mathcal{J}}(d, \mathcal{R}_p) & d \notin \mathcal{J} \\ -\infty & d \in \mathcal{J} \end{cases} \tag{24}$$

Setting $s_+^{\mathcal{J}}(d, \mathcal{R}_p)$ to $-\infty$ ensures that already pooled documents do not get selected again. The specific definition of $s^{\mathcal{J}}(d, \mathcal{R}_p)$ will be determined by each pooling strategy.

The set $\mathcal{J}$ grows as documents are pooled. The pooled documents after the n-th iteration of judgments will be referred to as $\mathcal{J}_n$. The construction of the $\mathcal{J}_n$s is achieved recursively:

$$\begin{aligned} \mathcal{J}_1 &= \tau@1(\mathcal{R}_p, s_+^{\emptyset}) \\ \mathcal{J}_n &= \mathcal{J}_{n-1} \cup \tau@1(\mathcal{R}_p, s_+^{\mathcal{J}_{n-1}}) \\ J_{\mathcal{R}_p} &= \mathcal{J}_n \end{aligned} \tag{25}$$

$\mathcal{J}_1$ contains the top-ranked document (beginning of the assessment process), and $\mathcal{J}_n$ contains all previously judged documents ($\mathcal{J}_{n-1}$) together with a newly selected document that depends on how $s_+^{\mathcal{J}_{n-1}}$ re-scores the documents. This definition of a pooling strategy generalizes the non-adaptive definition previously presented in Eq. 6.

There exists another type of adaptive strategy: the adaptive with run allocation. These adaptive strategies also specify which runs should be pooled (e.g., by iteratively choosing documents from one run or another). This is formalized by a sequence r_n that determines from which run r the documents have to be pooled.

We have seen that in the adaptive pooling strategies without run allocation, s is defined by the pooling strategy using as input the previous pooled documents. In the adaptive pooling strategies with run allocation, s is defined as follows:

$$s^{\mathcal{J}_{n-1}}(d, \mathcal{R}_p) = -\rho(d, r_n) \tag{26}$$

where the effect of run pooling strategy is only observed in the run allocation sequence ($\{r_n\}_{n\in\mathbb{N}_1}$), which is different for every strategy. This definition of s scores every documents of the allocated run in order of their retrieved rank position, and by substituting it into Eq. (24), it allows s_+ to re-rank to the end of the list all the documents already pooled.

Adaptive pooling strategies modify their behavior based on the current pooled documents. These strategies can be further divided into two categories based on which kind of document information is required in the adaptive stage: *non relevance-based*, and *relevance-based*. All the pooling strategies listed below are relevance-based pooling strategies, except for RBPAdaptiveTake@N, which is a non relevance-based one. The adaptive pools are incrementally built using the recursive definition of $J_{\mathcal{R}_p}$ in Eq. 25. We now describe the pooling strategies that belong to this category, classified by their origin: *classic strategies*, *IR evaluation measures*, and *multi-armed bandit models*.

3.2.1 Classic Strategies

This category includes traditional strategies developed in IR. In this category, two strategies exhibit adaptive behavior, the Move-To-Front strategy (MTFTake@N), and the Hedge strategy (HedgeTake@N).

MTFTake@N (MTF). *MTF* is a heuristic developed by [9], which associates a priority to each run. Initially, all runs have maximum priority. At every iteration of $\mathcal{J}_n$, this strategy selects a random run among the maximum priority runs. Then, it takes the first document retrieved by this run and judges it for relevance. At the next iteration, if the document was relevant ($\mathcal{J}^+_{n-1} \setminus \mathcal{J}_{n-2} \neq \emptyset$) then MTFTake@$N$ will continue selecting and judging documents from the same run. Otherwise, the priority of the current run is decreased and the method randomly selects another maximum priority run. We first define the following function that returns the number of times a run r has been sampled:

$$\#(r, r|_1^n) = |\{i \in \{1, 2, 3, \ldots, n\} : r = r_i\}| \tag{27}$$

The run selection sequence is defined as follows:

$$r_1 = \arg\min_{r\in\mathcal{R}_p} (\mu(0,1))$$
$$r_n = \begin{cases} r_{n-1} & \text{if } \mathcal{J}^+_{n-1} \setminus \mathcal{J}_{n-2} \neq \emptyset \\ \arg\min_{r\in\mathcal{R}_p} \big(|\{d \in \mathcal{D}_r : \\ : \rho(d,r) \leq \#(r, r|_1^{n-1})\} \cap \\ \cap \mathcal{J}^-_{n-1}| + \epsilon \cdot \mu(0,1)\big) & \text{otherwise} \end{cases} \tag{28}$$

r_1 makes an initial random selection (all runs have the maximum priority), and r_n either continues on the current run because the last document was relevant (r_{n-1}), or jumps to another maximum priority run. s is as defined in Eq. 26 and J in Eq. 25.

HedgeTake@N (H). This strategy is an online learning algorithm proposed by Aslam et al. [39] for metasearch and pooling. It associates a set of losses to the contributing runs. These losses depend on the relevance outcomes and the positions in the runs of the judged documents. For example, a run's loss is increased (decreased) if the run retrieved a non-relevant (relevant) document at a high position. After each assessment, the run's losses are updated and the next pick (next assessed document) depends on the run's losses and the positions of the unjudged documents in the runs. For each document-run pair, the following function takes the document's position and estimates the loss we would obtain if the document is deemed non-relevant:

$$G(\rho) = \ln(|\mathcal{D}|/\rho) \tag{29}$$

This loss needs to be computed for all documents (including those that do not belong to the run). This is achieved by extending G as follows:

$$G^*(d, r) = \begin{cases} G(\rho(d, r)) & \text{if } d \in \mathcal{D}_r \\ \text{Avg}_{|r|<i\leq|\mathcal{D}|}\, G(i) & \text{otherwise} \end{cases} \tag{30}$$

If the document does not belong to the run then the loss is estimated as the average loss the document would get if retrieved in positions from $|r| + 1$ to $|\mathcal{D}|$. As we obtain relevance assessments, we iteratively accumulate the loss induced by each run ($L(r, \mathcal{J})$). These runs' losses depend on the relevance outcomes and the positions in the runs of the judged documents (as defined by G^*). The loss of run r is defined as follows:

$$L(r, \mathcal{J}_{n-1}) = \frac{1}{2} \sum_{d\in\mathcal{J}^-_{n-1}} G^*(d, r) - \frac{1}{2} \sum_{d\in\mathcal{J}^+_{n-1}} G^*(d, r) \tag{31}$$

Next, the loss is normalized by:

$$\overline{L}(r, \mathcal{J}_{n-1}) = \frac{\beta^{L(r,\mathcal{J}_{n-1})}}{\sum_{r'\in\mathcal{R}_p} \beta^{L(r',\mathcal{J}_{n-1})}} \tag{32}$$

This normalization has a parameter $\beta \in [0, +\infty[$ that controls the way in which new judgments change the weights. We set $\beta = 0.1$ as in Losada et al. [15]; other values will be investigated in future work. Finally, s is defined as the weighted average of the documents' losses across all runs:

$$s^{\mathcal{J}_{n-1}}(d, \mathcal{R}_p) = \sum_{r\in\mathcal{R}_p} \big(\overline{L}(r, \mathcal{J}_{n-1}) \cdot G^*(d, r)\big) + \epsilon \cdot \mu(0, 1) \tag{33}$$

J is defined as in Eq. 25. It is interesting to observe that this strategy takes into account also the non-relevant documents. Now, we make some observations about how this pooling strategy changes behavior as we vary β. In particular we analyze three special values of β, when $\beta \to 0$, $\beta = 1$, and $\beta \to +\infty$ (see Electronic Appendix 2). When $\beta = 1$ we observe that this strategy reduces to a non-adaptive evaluation measure-based strategy with G defined as follows:

$$G(\rho) = \log\left(\frac{1}{\rho}\right) + \frac{\log(|\mathcal{D}|!)}{|\mathcal{D}|} \tag{34}$$

When β tends to $+\infty$, we observe that this strategy reduces to a MTFTake@N like pooling strategy. This observation derives from the fact that when β tends to $+\infty$ the normalization in Eq. 32 will select the run that has the largest L score. From this run, due to Eq. 33, the document with the highest rank, not yet pooled, is selected. Now, if the document was relevant, a new document will be picked from the same run because it is still the run with the largest score; if the document is not relevant, the score of the run is reduced, and a new document will be picked potentially from a run with a larger score, like the MTFTake@N strategy. However

there is a main difference between these two strategies, for MTFTake@N the run is kept the same every time a picked document is judged relevant, in this case this is embedded in the definition of selection of the run by increasing the score for the run. When β tends to 0, we observe an opposite behavior than the one observed when β tends to $+\infty$: the score for a run is increased if the retrieved document is non-relevant and decreased if the document is relevant. This generates a pooling strategy that instead of keeping sampling from runs that retrieved relevant documents like MTFTake@N, it keeps sampling from runs that retrieved non-relevant documents.

3.2.2 IR Evaluation Measure-Based Strategies

The next two strategies are extensions of RBPTake@N. Thanks to the convergent behavior of RBP, Moffat et al. [21] have naturally extended RBPTake@N to include additional information into the scoring function s.

RBPAdaptiveTake@N (*RBP*$^{\mathbf{A}}$). This strategy is an adaptive version of *RBP*, which adds documents to the pool in an incremental way. For each run $r \in \mathcal{R}_p$, it computes its residual $e(r, \mathcal{J})$, i.e. a value proportional to the number of not judged documents in the run. The residual is defined as:

$$e(r, \mathcal{J}_{n-1}) = p^{|r|} + (1-p) \sum_{d \in \mathcal{D}_r : d \notin \mathcal{J}_{n-1}} p^{\rho(d,r)-1} \quad (35)$$

where the first term represents the residual obtained by counting the contribution of every non-retrieved and non-judged document by r (from its actual depth $|r|$ to the infinite depth), and the second term represents the residual obtained counting the contribution of every non-judged document within the run depth $|r|$. s is defined as follows:

$$s^{\mathcal{J}_{n-1}}(d, \mathcal{R}_p) = \\ = \sum_{r \in \mathcal{R}_p : d \in \mathcal{D}_r} (G_{\text{RBP}}(\rho(d,r)) \cdot e(r, \mathcal{J}_{n-1})) + \epsilon \cdot \mu(0,1) \quad (36)$$

With each new selection, the runs' residuals change and the score $s^{\mathcal{J}_{n-1}}(d, \mathcal{R}_p)$ needs to be recomputed (thus, the adaptive nature of RBPAdaptiveTake@N). J is defined as in Eq. 25.

RBPAdaptive*Take@N (*RBP*$^{\mathbf{A^*}}$). This pooling strategy is also an adaptive pooling strategy that uses both the RBP residuals, as RBPAdaptiveTake@N, and the actual RBP score $b(r, \mathcal{J})$ of a run r, computed using binary relevance:

$$b(r, \mathcal{J}_{n-1}) = \sum_{d \in \mathcal{D}_r : d \in \mathcal{J}^+_{n-1}} G_{\text{RBP}}(\rho(d,r)) \quad (37)$$

The candidate documents for pooling are ranked by decreasing:

$$s^{\mathcal{J}_{n-1}}(d, \mathcal{R}_p) = \\ = \sum_{r \in \mathcal{R}_p : d \in \mathcal{D}_r} \Bigg[G_{\text{RBP}}(\rho(d,r)) \cdot e(r, \mathcal{J}_{n-1}) \cdot \\ \cdot \left(b(r, \mathcal{J}_{n-1}) + \frac{e(r, \mathcal{J}_{n-1})}{2} \right)^3 \Bigg] + \epsilon \cdot \mu(0,1) \quad (38)$$

At each iteration n, this strategy uses the information about the relevance of the last selected document (observe the set of judged relevant documents $\mathcal{J}^+_{n-1}$ in Eq. 38). Being an adaptive strategy, J is defined as in Eq. 25.

3.2.3 Multi-Armed Bandit Models-Based Strategies

These strategies model pooling as a multi-armed bandit problem [15]. The bandit-based strategies are adaptive. As we select and judge documents, we gain knowledge on the quality of the contributing runs. Run selection is driven by the classical exploration versus exploitation dilemma. This works as follows: At any point, we can opt for *exploiting* our current knowledge (i.e. choose the run that has supplied the highest average number of relevant documents) or, alternatively, we can opt for *exploring* (i.e. choose a suboptimal run). Exploitation maximizes the expected reward on the next pick, but exploration may produce the greater total reward over a long period of time (the runs that are currently inferior can eventually become good suppliers of relevant documents). Every bandit-based strategy implements a specific bandit allocation method. A bandit allocation method chooses the next pick (next run) based on past actions and obtained rewards (relevance of judged documents).

MABGreedyTake@N (BG). This strategy is based on the ϵ-greedy bandit allocation method. A greedy approach consists of always selecting the run with the largest average of judged relevant documents. This greedy approach, which is similar to MTFTake@N, has been shown to be sub-optimal. A simple variant consists of behaving greedily most of the time and sometimes selecting a random (suboptimal) run. A simple strategy that implements this idea is ϵ_n-greedy [40]. At any point, ϵ_n-greedy plays with probability $1 - \epsilon_n$ the run with the highest average of judged relevant documents, and with probability ϵ_n a randomly chosen run. ϵ_n is known as the exploration probability. It is good practice setting ϵ_n such that it decreases with the number of picks (n). This is because estimates become more accurate as more evidence is encountered and, therefore, the exploration probability should decrease. We employ the following definition of $\epsilon_n = \min(1, c_0|\mathcal{R}_p|/(c_1^2(n-1)))$, where c_0 and c_1 are parameters. Following Losada et al. [15], we set c_0 to 0.01, and c_1 to 0.1. For each run, we first compute the proportion of the run's judged documents that were deemed as relevant:

$$P(r, r|_1^n, \mathcal{J}_n) = \begin{cases} 1/2 & \#(r, r|_1^n) = 0 \\ \frac{|\{d \in \mathcal{D}_r : \rho(d,r) \leq \#(r, r|_1^n)\} \cap \mathcal{J}^+_n|}{\#(r, r|_1^n)} & \text{otherwise} \end{cases} \quad (39)$$

following the run succession used by s as defined in Eq. 26:

$$r_n = \begin{cases} \arg\max_{r \in \mathcal{R}_p}(\mu(0,1)) & \mu(0,1) < \\ & < \min\left(1, \frac{c_0|\mathcal{R}_p|}{c_1^2(n-1)}\right) \\ \arg\max_{r \in \mathcal{R}_p}(& \\ P(r, r|_1^{n-1}, \mathcal{J}_{n-1}) + & \\ +\epsilon \cdot \mu(0,1)) & \text{otherwise} \end{cases} \quad (40)$$

The second line of the equation above encodes the greedy action, which selects the run with the highest average ($\epsilon \cdot \mu(0,1)$, again, is incorporated here to break the ties), while the first line encodes the exploration action (random run selection). J is defined as in Eq. 25.

MABUCBTake@N (UCB). This strategy implements the UCB1-Tuned method [41]. UCB associates an *upper confidence index* to each run. This index estimates the uncertainty about the quality of the run (average relevance of documents from the run). After n rounds of judgment, we

would like to sample from the *leading* run (the one with the largest proportion of judged relevant documents). However, we need to be sure that the other runs have been sampled enough. Otherwise, we cannot be sure that they are indeed inferior. MABUCBTake@N (UCB) computes upper confidence bounds for the proportions of relevant documents supplied by the runs and compares the upper confidence bounds of apparently inferior runs with the estimated mean of the leading run. The index of the UCB1 strategy is the sum of two components: the current estimated mean and a quantity related to the size of the one-sided confidence interval for the estimated mean. UCB1-Tuned is an evolution of UCB1 that takes into account the variance of each run. In this strategy we use the probability of extracting relevant documents as defined in Eq. 39, and we define its average by renaming the function P in Eq. (39) defined in the previous strategy as follows:

$$P_{\mu}(r, r|_1^n, \mathcal{J}_n) = P(r, r|_1^n, \mathcal{J}_n) \tag{41}$$

The definition of its variance is:

$$P_{\sigma^2}(r, r|_1^n, \mathcal{J}_n) = P(r, r|_1^n, \mathcal{J}_n)(1 - P(r, r|_1^n, \mathcal{J}_n)) \tag{42}$$

S defines the reward to maximize:

$$S(r, \mathcal{J}_{n-1}, r|_1^{n-1}) = P_{\mu}(r, r|_1^{n-1}, \mathcal{J}_{n-1}) + \sqrt{\frac{\ln(n-1)}{\#(r, r|_1^{n-1})}} \cdot \cdot \sqrt{\min\left(\frac{1}{4}, P_{\sigma^2}(r, r|_1^{n-1}, \mathcal{J}_{n-1}) + \sqrt{\frac{2\ln(n-1)}{\#(r, r|_1^{n-1})}}\right)} + + \epsilon \cdot \mu(0,1) \tag{43}$$

Here, we observe that, for the reward to be properly defined, $\#(r, r|_1^{n-1})$ must always be ≥ 1. To guarantee this, all the runs get the first document evaluated. Therefore, in the definition of P in Eq. (39) used to define P_{μ} and P_{σ^2}, we can ignore the first case when $\#(r, r|_1^n) = 0$. The initialization is achieved by defining F as follows:

$$F(r, \mathcal{J}) = \max_{d \in \mathcal{D}_r : d \notin \mathcal{J}}(-\rho(d, r)) + \epsilon \cdot \mu(0,1) \tag{44}$$

and the run allocation policy is defined as:

$$r_n = \begin{cases} \arg\max_{r \in \mathcal{R}_p}(F(r, \mathcal{J}_{n-1})) & \text{if } \exists\ r \in \mathcal{R}_p, \\ & \exists\ d \in \mathcal{D}_r : \rho(d, r) = 1 \\ \arg\max_{r \in \mathcal{R}_p}(& \\ S(r, \mathcal{J}_{n-1}, r|_1^{n-1})) & \text{otherwise} \end{cases} \tag{45}$$

J is defined as Eq. 25.

MABBetaTake@N (BB). This strategy is based on a heuristic called Thompson sampling [42]. It represents each run with a probability of supplying a relevant document, and each run's probability is associated with a probability distribution under a Bayesian framework. The process begins with no knowledge of these probabilities. This is encoded by applying a uniform prior for each run. This uniform initialization, which is equivalent to the Beta distribution when assigning its shape parameters $\alpha = 1$ and $\beta = 1$ (Beta$(1, 1)$), represents the lack of knowledge about the chances of extracting relevant documents from each run. Run selection is done by extracting a sample from each distribution ($|\mathcal{R}_p|$ samples, one from each Beta distribution) and the run yielding the largest sample is chosen. This selection approach tends to select runs that have a high mean (i.e. high likelihood of yielding relevant documents). Next, the top ranked unjudged document of the chosen run is judged for relevance, and the relevance outcome is used for updating the run's Beta distributions. With binary relevance, the relevance outcome can be modeled as a Bernoulli variable. This is a mathematical convenience because it guarantees that the update leads to posterior distributions (after incorporating the new evidence) that are also Beta distributed. So, we iteratively update the parameters of the Beta distributions based on the relevance of the judged documents. The run allocation sequence used by s in Eq. 26 is defined as follows:

$$r_n = \arg\max_{r \in \mathcal{R}_p}(\text{Beta}(1 + |r \cap \mathcal{J}_{n-1}^+|, 1 + |r \cap \mathcal{J}_{n-1}^-|)) \tag{46}$$

J is defined as Eq. 25. To be noted that here the small random component ($\epsilon \cdot \mu(0,1)$), useful to break the ties, is not necessary since it is already a stochastic process.

MABMaxMeanTake@N (MM). This is another Bayesian solution that represents the runs with Beta probabilities and updates the probability distributions based on the relevance assessments. The difference between MABBetaTake@N and MABMaxMeanTake@N is that MABMaxMeanTake@N does not make run selection by sampling from the Beta distributions. The run selected by MABMaxMeanTake@N is simply the one that has the maximum mean of the Beta distributions. The run allocation sequence, used in s as in Eq. 26, is defined as:

$$r_n = \arg\max_{r \in \mathcal{R}_p}\left(\frac{1 + |r \cap \mathcal{J}_{n-1}^+|}{2 + |r \cap \mathcal{J}_{n-1}^-|} + \epsilon \cdot \mu(0,1)\right) \tag{47}$$

J is defined as in Eq. 25.

Losada et al. [15] also describe a non-adaptive version of a multi-armed bandit based-strategy, which randomly allocates the runs from which to select the documents to be pooled. However, this strategy, as expected, performs similarly to FairTake@N, therefore it has not been considered in this article.

4 Experiments & Results

We do a large-scale evaluation in terms of pool bias of the 22 pooling strategies presented above on 9 test collections using 3 measures of bias and 3 IR evaluation measures. In this section we first present the experimental design. Next, we present the material and experiment setup. We then introduce the measures of bias; and finally, we present the results.

4.1 Experimental Design

In the introduction we have presented that the pooling method is used to build test collections in evaluation efforts like TREC. In these evaluation efforts, an evaluation challenge is instantiated and the set of topics $\mathcal{Q}$ to be evaluated defined. Next, participating organizations $\mathcal{O}$ are invited to submit a set of runs of a given size, of which a subset per $\mathcal{O}$ is then used to form the set of pooled runs $\mathcal{R}_p$. Next, a

pooling strategy is used to pool the documents to be judged by human relevance assessors. At the end of this building process, a test collection is released that is then used in laboratory experiments that, unavoidably, will suffer from pool bias.

In order to compare the effectiveness of the pooling strategies presented above in mitigating the effect of pool bias, we run a series of simulation experiments in which we simulate the process of building a test collection. One simulation consists in, given a set $\mathcal{O}$, building a test collection with the runs submitted by $|\mathcal{O}| - 1$ organizations and measure the bias on the runs submitted by the leftover organization. This can be formally expressed as follows: Given a set of organizations $\mathcal{O}$, a set of runs $\mathcal{R}_p$ submitted by $\mathcal{O}$, and an *ideal* set of judgments I that has a relevance value for each document, we can compute an ideal mean absolute error for a pooling strategy J as follows:

$$\frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} \left| f(r, J_{\mathcal{R}_p \setminus \{r' \in \mathcal{R}_p : o_{r'} = o_r\}}) - f(r, I) \right| = \\ = \frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} \left| b(r, J_{\mathcal{R}_p \setminus \{r' \in \mathcal{R}_p : o_{r'} = o_r\}}) \right| \quad (48)$$

where the right-hand side is obtained by substituting the definition of pool bias from Section 1. This is referred to in the literature as a *leave-one organization-out* approach. This approach is preferred to a *leave-one run-out* approach because it better simulates the case that the retrieval model used by the organization has not contributed to the pool. However, due to the presence (in Eq. (48)) of the ideal set of judgments I, which in reality does not exist, this error cannot be computed. Instead, in IR we usually dispose of an approximation of this set I, which in the following we indicate as the ground-truth G. The use of G in the measurement introduces a random error in the observed measurement $f(r, J_{\mathcal{R}_p})$, which we define as follows:

Definition 2. We define as the *random error*, the difference we would observe on a measure f applied on a run r between the actual measurement and the true measurement:

$$\varepsilon = f(r, G) - f(r, I) \quad (49)$$

where I is the ideal set of judgments, therefore making $f(r, I)$ the true measurement and G the actual ground-truth, therefore making $f(r, G)$ the *actual measurement*.

This difference is defined as the random error because we have no means of control over it. By its definition, the random error goes to zero, if $f(r, G)$ tends to $f(r, I)$ when the number of judged documents $|G|$ tends to $|I|$. By using this definition we can define the actual Mean Absolute Error (MAE) as:

$$\text{MAE}(J_{\mathcal{R}_p}) = \\ = \frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} \left| f(r, J_{\mathcal{R}_p \setminus \{r' \in \mathcal{R}_p : o_{r'} = o_r\}}) - f(r, G) \right| = \\ = \frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} \left| b(r, J_{\mathcal{R}_p \setminus \{r' \in \mathcal{R}_p : o_{r'} = o_r\}}) + \varepsilon \right| \quad (50)$$

Therefore, when using G in the simulations that calculate MAE, the absolute value we are measuring is a composition of the pool bias and random error. However, we claim that this random error is not an issue for our comparison because: (1) this is an error measured between G and I, which makes it independent and constant across the set of tested pooling strategies Js; (2) the presence of this error is in line with standard evaluation praxis in IR, because this is the same error we would observe every time we test a run on an existing test collection; (3) the random error is 0 for some combination of f and G, e.g. this happens when f is P@n and at least the first n documents retrieved by r are contained in G.

In order to measure the difference in pool bias we must have perfect knowledge of all the documents that appear in any of the runs. The objective of these experiments is to quantify the effect of missing information (introduced by the pooling strategy) — therefore, we cannot allow missing information to exist at the onset of the experimental process. In this context, the best test collections are those originally built with Depth@K, because this requirement is easily satisfiable by using the pooled runs $\mathcal{R}_p$ and resizing them to a depth equal to $|r| = K$.

This process of test collection transformation is depicted in Figure 2. Essentially, the newly created test collections are "clean" in the sense that no information is kept for any of the runs for ranks above K. This cleaning is essential in order to ensure the validity of the experiments with different pooling strategies. If we were not to do this cleaning, when using $f(r, G)$ to observe the pool bias resulting of the use of a particular pooling strategy we would be confounding it with the pool bias of the original test collection.

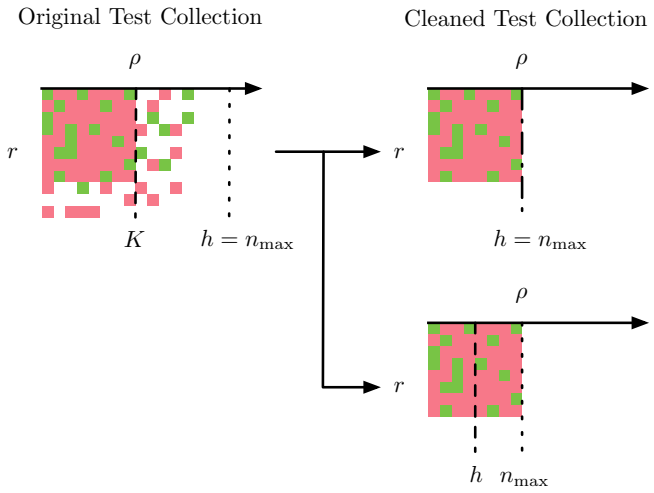

Fig. 2. In the top left corner we illustrate the shape and setup of the original test collection. The y-axis indicates the runs, the x-axis the rank, every block represents a pooled document, which color indicates its status: green if relevant, red if irrelevant, and white for unjudged. K indicates the depth of the pooling strategy used to build the original test collection; h indicates the horizon of the pooling strategy; and n_{max} the maximum evaluation depth available. In the right corner we present the shape and setup of the three experiments. At the top, the shape and setup used to compare the performance of the different pooling strategies and compare the expected number of judged documents. At the bottom, the shape and setup used to verify the consistency of the results of the first experiment varying h.

This experimental design raises two potential issues: 1) the selection of too few documents to be judged (low

N) may cause the reduction of the number of judged documents per topic at the level that makes any analysis based on this judgments inconclusive; 2) the resizing of the runs may have unexpected effects on the conclusion of the simulation experiments, i.e., would a certain pooling strategy be preferred for a lower runs' size and another one for a higher one?

To address these questions we design two additional experiments. In the first, for every pooling strategy J, we measure the average number of judged documents (AJ) for the pair run-topic, which we define as follows:

$$\text{AJ}(J_{\mathcal{R}_p}) = \frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} \left| \left\{ d \in \mathcal{D}_r : d \in J_{\mathcal{R}_p \setminus \{r' \in \mathcal{R}_p : o_{r'} = o_r\}} \right\} \right| \tag{51}$$

this is then average across topics. AJ measures the expected number of judged documents we would expect on a new run. In the second experiment we verify the consistency of the results when used in a real setting. To do this with the same test collections we test the same case but reducing what we call the horizon of the pooling strategies. The horizon (h) is defined as the depth of the runs available to the pooling strategy. If the results found are not consistent with the ones found by the designed experiment, we have to reconsider our previous conclusions, if they are consistent, it means that the horizon effect is a negligible effect in our comparison. To illustrate our methodology we provide a graphical representation of both experiments, the designed one and this new one in Figure 2.

4.2 Material

To test the effectiveness of the different pooling strategies we selected 9 test collections from TREC [1]: Ad Hoc 3 [43], Ad Hoc 8 [44], Web 9 [45], Web 23 [46], Robust 14 [47], Genomics 14 [48], Legal 15 [49], Blog 15 [50], and Microblog 20 [51]. These test collections are named by concatenating the name of the track and the edition of TREC in which they have been built. We selected these test collections because of: 1) the diverse origin – in fact they cover 6 different domains: News, Web, Genomics, Legal, Blog, and Microblog; 2) the large number of judged documents in the collections; 3) the large number of organizations that contributed to the pools – we assume that the number of participating organizations is directly proportional to the variety of the submitted runs, and 4) the pooling strategy used to build the collections, i.e., *fixed depth at cut-off K* pooling strategy (Depth@K). The last point makes the collections suitable for testing new pooling strategies. As explained in the sample design, these test collections require to be normalized to a clean Depth@K. In addition, due to the prototypical nature of the tracks organized to build the test collections, we filtered out the 25% of lowest performing runs from our experimentation. This filtering is done to remove those runs that are likely to contain bugs or very exploratory methods. This procedure is in line with standard practices in the IR field [52].

4.3 Measures of Pool Bias

The measures of pool bias take as input an IR evaluation measure f. We have already presented the first measure of bias in Eq. 50, the mean absolute error (MAE). This measure estimates the expected observed pool bias plus random error on the score of a non-pooled run. This is done by averaging the difference in score of the every $r \in \mathcal{R}_p$ when pooled with the ground truth G, and when non-pooled, together with the runs submitted by its same organization, with a fixed-cost pooling strategy (J). A low MAE means that the score obtained by a run with J strategy when not pooled is close to the score obtained by the run when evaluated with the ground-truth.

The second measure of bias we present is system rank error (SRE). This measure counts the number of rank positions lost or gained by runs in the system ranking with respect to when it is pooled with the ground truth G, defined by the test collection, and not pooled with a fixed-cost pooling strategy (J). We define SRE as:

$$\begin{aligned} \text{SRE}(J_{\mathcal{R}_p}) = \sum_{r \in \mathcal{R}_p} \Big| \Big\{ & r' \in \mathcal{R}_p \setminus \{r'' \in \mathcal{R}_p : o_{r''} = o_r\} : \\ & : f(r, J_{\mathcal{R}_p \setminus \{r'' \in \mathcal{R}_p : o_{r''} = o_r\}}) \leq f(r', G) < f(r, J_{\mathcal{R}_p}) \vee \\ & \vee f(r, J_{\mathcal{R}_p}) < f(r', G) \leq f(r, J_{\mathcal{R}_p \setminus \{r'' \in \mathcal{R}_p : o_{r''} = o_r\}}) \Big\} \Big| \end{aligned} \tag{52}$$

A low SRE means that the rank position of the runs when not pooled using J is close to the rank position of the runs when pooled with the ground-truth. In IR when comparing ranking of runs, it is common practice to evaluate their significance. We implemented this in the next measure named system rank error with statistical significance (SRE*). SRE* is similar to SRE but instead of counting all the position differences of a run against all the other runs, it counts only if significant according to a paired t-test with $p < 0.05$ calculated on the ground-truth. SRE* is defined as follows:

$$\begin{aligned} \text{SRE}^*(J_{\mathcal{R}_p}) = \sum_{r \in \mathcal{R}_p} \Big| \Big\{ & r' \in \mathcal{R}_p \setminus \{r'' \in \mathcal{R}_p : o_{r''} = o_r\} : \\ & : \Big(f(r, J_{\mathcal{R}_p \setminus \{r'' \in \mathcal{R}_p : o_{r''} = o_r\}}) \leq f(r', G) < f(r, J_{\mathcal{R}_p}) \vee \\ & \vee f(r, J_{\mathcal{R}_p}) < f(r', G) \leq f(r, J_{\mathcal{R}_p \setminus \{r'' \in \mathcal{R}_p : o_{r''} = o_r\}}) \Big) \wedge \\ & \wedge \text{t-test}_{\text{paired}}(r, r', G) < 0.05 \Big\} \Big| \end{aligned} \tag{53}$$

Juxtaposing these measures of bias we can observe that a zero MAE value implies that SRE and SRE* are also equal to zero. However, the contrary is not true. We can also observe that this is true between SRE and SRE*, where a zero SRE corresponds to a zero SRE*, but not vice versa.

4.4 Experimental Setup

In this paragraph we present the setup of the first two experiments, the first, designed to compare the pooling strategies, and the second, where we measure the expectation of the number of judged documents per run. For these two experiments, each pooling strategy takes as parameter the pool size, i.e., the number of judged documents. To test how the different strategies behave for different values of this parameter, we repeated the experiment varying the pool size from 5,000 in steps of 5,000 till all the judgments of the test collection were used. We did this for Ad Hoc 3, Ad Hoc 8, and Web 9. For Blog 15 we varied the pool size from 2,000 in steps of 2,000, and for Genomics 14, Legal 15, Microblog 20, Robust 14, and Web 23 we varied the pool size from

1,000 in steps of 1,000 due to the smaller size of these test collections.

In the third experiment, when we verify the stability of the first experiment, for each pooling strategy we fix $N = 10,000$ we then repeated the experiment varying the horizon h from 10, in steps of 10 till the size of the original test collection K. We did this for all the test collections.

In all three experiments, the pool size N, when possible, is equally divided across the topics. Due to an imbalance of documents judged in the original Depth@K strategy among the topics, for big Ns and for some topics we would not find enough documents to cover the number of allocated judgments for these topics, $N/|\mathcal{Q}|$, where $|\mathcal{Q}|$ is the number of topics. In this case the number of judged documents available per topic can vary. Therefore, the aggregated number of documents to be judged for a fixed-cost pooling strategy would not equal the desired pool size of N judged documents. To avoid this, we implement a heuristic that redistributes the remaining judgments, when needed, fairly across the rest of the topics that still have available documents. Given as input the set of pooled runs ($\mathcal{R}_p$) this heuristic does, in order to achieve the prefixed N pooled documents across topics, a search on the space of possible per-topic sizes. This search space is constrained by the fact that every per-topic size cannot be greater than the number of available judged documents per topic. The heuristic first starts by assigning to each topic q a per-topic size n_q equal to N divided by the number of topics ($N/|\mathcal{Q}|$). So for example, if we have a N of 10,000 documents for 50 topics the heuristic assigns to every topic an $n_q = 200, \forall q \in \mathcal{Q}$. Now, if for some topics the assigned n_qs are too large, for example there is a lack of documents to be judged for these topics the heuristic then reduces the n_qs of these topics to the maximum allowed (that is of course smaller than $N/|\mathcal{Q}|$) and reassigns the remaining judgments to the other topics for which there are still available documents. The reassignment is done by incrementing by 1 each topic n_q until one of the two conditions is verified: 1) the topic has been exhausted, that is no more documents are available, in this case the topic is excluded and the algorithm continues with the other topics, or 2) the sum of the n_qs has reached N ($n_1 + ... + n_{|\mathcal{Q}|} = N$), in this case the algorithm stops returning the found solution. However if this second condition is not verified before all the topics get exhausted the heuristic returns an error. This means that there are not enough documents already judged in the original pool to achieve a solution of size N.

The IR evaluation measures we selected for this study are AP, NDCG, and P@10. The reason for this selection is twofold: (a) these measures are widely used in IR, and (b) they encompass common features of most IR measures: top-heaviness, precision based, recall based, and utility based.

The software used in this article to evaluate the proposed pooling strategies is available on the website of the first author.

4.5 Results

In Figure 3 we show the bias evaluation obtained using the non-adaptive pooling strategies and in Figure 4 the bias evaluation obtained using the adaptive ones for the Ad Hoc 8 test collection. In the figures, each column is an IR evaluation measure while each row is a measure of bias. The x-axis in each of the plots is the number of judged documents, while the y-axis is the scale of the respective measure bias. Every line is a pooling strategy. In Figures 5 and 6 we show for 3 other test collections but measuring MAE on the evaluation measure AP. Similar patterns of behavior were observed for the other test collections.

TABLE 1
Performance comparison for the Ad Hoc 8 and $N = 10,000$.

J	$\lvert\mathcal{J}^+\rvert$	AP			NDCG			P@10		
		MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
F	1681	.0655	1104	423	.0961	1229	579	.0265	190	34
B	2193	.0541	974	309	.0858	1150	503	.0150	46	**9**
C	2193	.0542	976	311	.0860	1148	501	.0150	47	**9**
MAX	1939	.0456	905	238	.0694	1052	409	.0348	305	47
MIN	557	.1333	2066	1356	.1823	1987	1313	.3728	2481	1697
MED	1221	.0828	1203	520	.1126	1314	659	.0601	664	185
SUM	2328	.0475	912	252	.0726	1059	416	.0134	48	**9**
ANZ	675	.0716	1267	583	.0413	802	267	.1929	1834	1051
MNZ	2258	.0494	928	268	.0779	1103	458	**.0128**	**38**	**9**
DCG	2195	.0536	972	307	.0841	1142	495	.0140	40	**9**
RRF	2205	.0530	961	296	.0834	1136	490	.0140	41	**9**
PP	2188	.0545	976	311	.0864	1153	506	.0154	50	**9**
RBP	1782	.0628	1080	402	.0932	1206	556	.0219	120	22
RBP^A	1690	.0649	1097	417	.0954	1220	570	.0255	171	34
RBP^{A*}	2084	.0511	959	294	.0761	1100	453	.0182	91	10
H	2635	.0229	528	28	.0345	651	76	.0285	254	**9**
MTF	2464	.0386	819	168	.0590	957	317	.0162	87	9
BG	2053	.0515	974	305	.0776	1102	455	.0219	140	17
UCB	1903	.0576	1039	361	.0856	1171	521	.0236	157	23
BB	3019	.0210	503	19	.0323	622	55	.0197	157	**9**
MM	**3267**	**.0160**	**391**	**5**	**.0247**	**520**	**25**	.0179	147	**9**

In these figures we can observe that all lines converge to a pool bias value of the test collection for large pool size values. This is because all the pooling strategies are constrained to select documents for which we have relevance assessments. This is done as explained previously by only including in the analysis pooled runs and by resizing them to the same depth of the Depth@K pooling strategy used to build the test collection. Thereby, all alternative pooling strategies will reduce to the original Depth@K strategy with K defined by the original test collection.

In Tables 1 we show the performance of each pooling strategy for $N = 10,000$ for Ad Hoc 8. Similar results are observed for the other test collections.

In Figure 7, we show the expected number of judged documents across runs and topics (JD) for Ad Hoc 8. Similar patterns of behavior were observed for the other test collections. The JD values give us an estimate of how many documents we should expect to be judged for a non pooled run and for a single topic. Every line is a pooling strategy, and the x-axis in both plots is the total number of judged documents, while the y-axis is the scale of JD.

In Figure 8, we show the stability of the results when varying the horizon of the pooling strategies for a fixed pool size $N = 10,000$ for Ad Hoc 8. Every line is a pooling strategy, and the x-axis in both plots is the horizon, while the y-axis is MAE measured on AP. Similar patterns of behavior were observed for the other test collections, IR evaluation measures, measure of bias, and N values.

5 Discussion

In the following we discuss the results reported above. We consider the FairTake@N strategy as our baseline. While this strategy is slightly different from Depth@K (see Section 3), FairTake@N is the strategy closest to Depth@K that

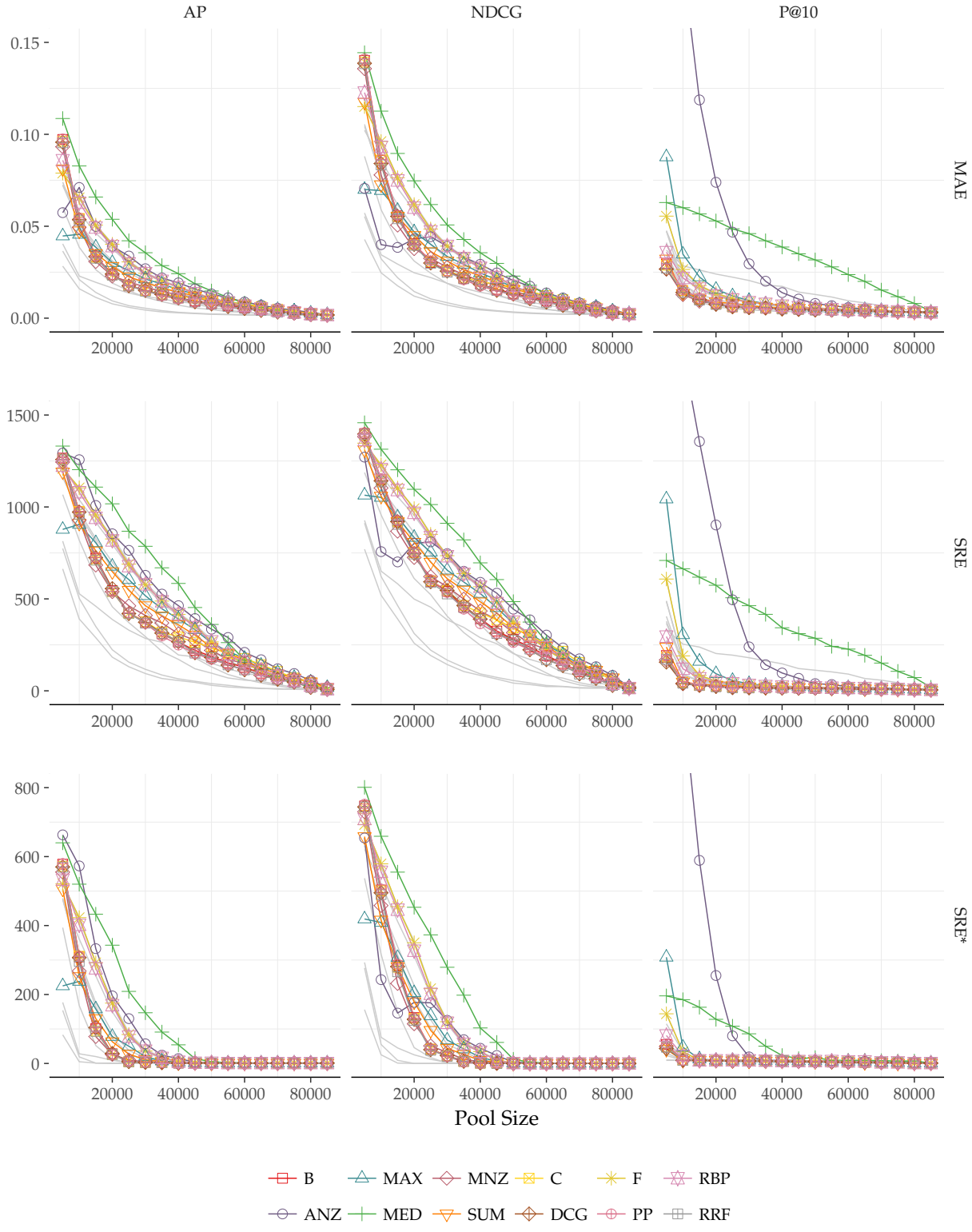

Fig. 3. Pool bias measured for the *non-adaptive* pooling strategies in terms of the measures of bias (row): MAE, SRE, and SRE*, and IR evaluation measures (columns): AP, NDCG, and P@10. This is plot by using the Ad Hoc 8 test collection, and for different pool sizes (i.e. aggregated number per topic of documents that require relevance judgment). The lines in gray are the *adaptive* pooling strategies (in Fig. 4) for comparison.

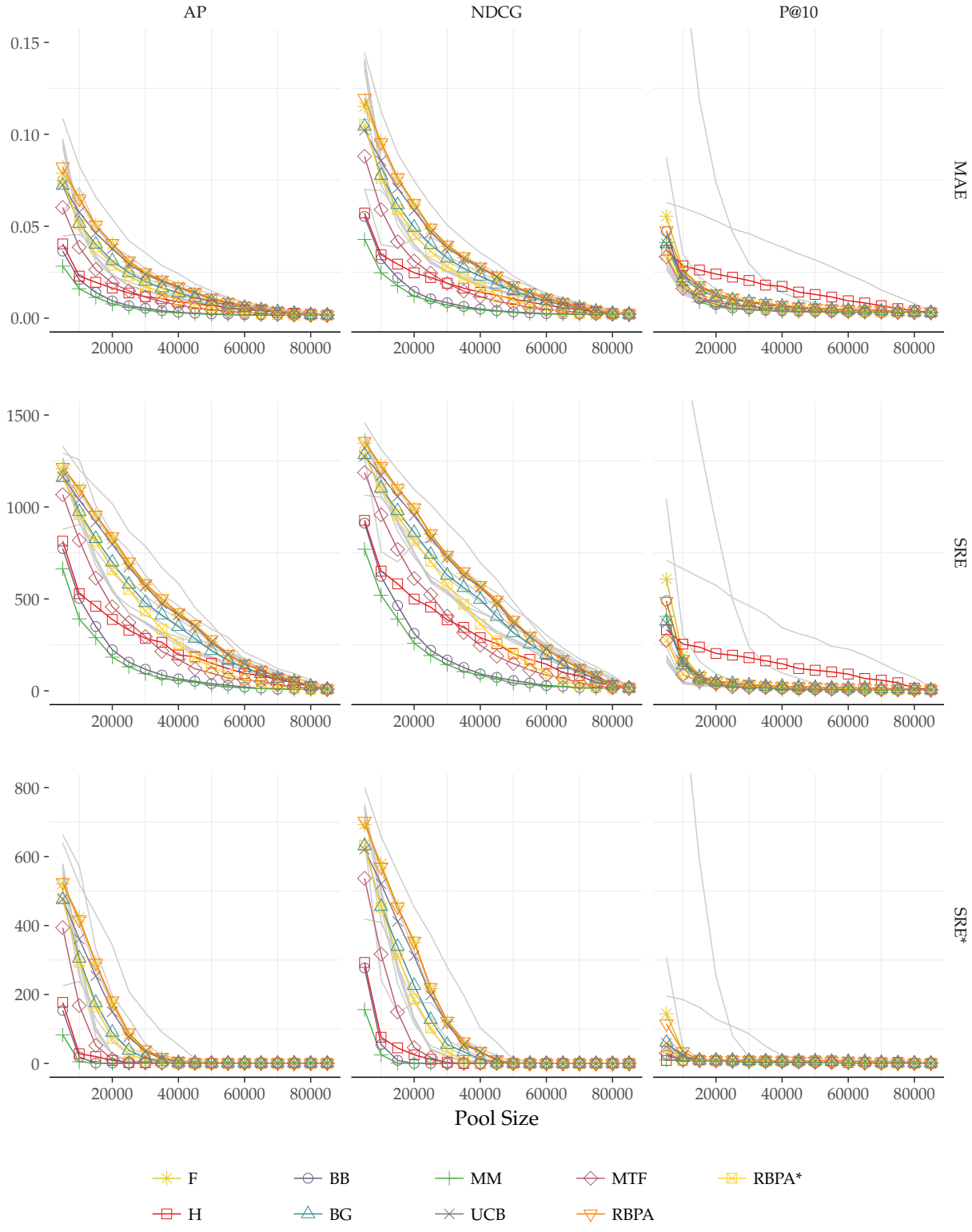

Fig. 4. Pool bias measured for the *adaptive* pooling strategies in terms of the measures of bias (row): MAE, SRE, and SRE*, and IR evaluation measures (columns): AP, NDCG, and P@10. This is plot by using the Ad Hoc 8 test collection, and for different pool sizes (i.e. aggregated number per topic of documents that require relevance judgment). The lines in gray are the *non-adaptive* pooling strategies (in Fig. 3) for comparison.

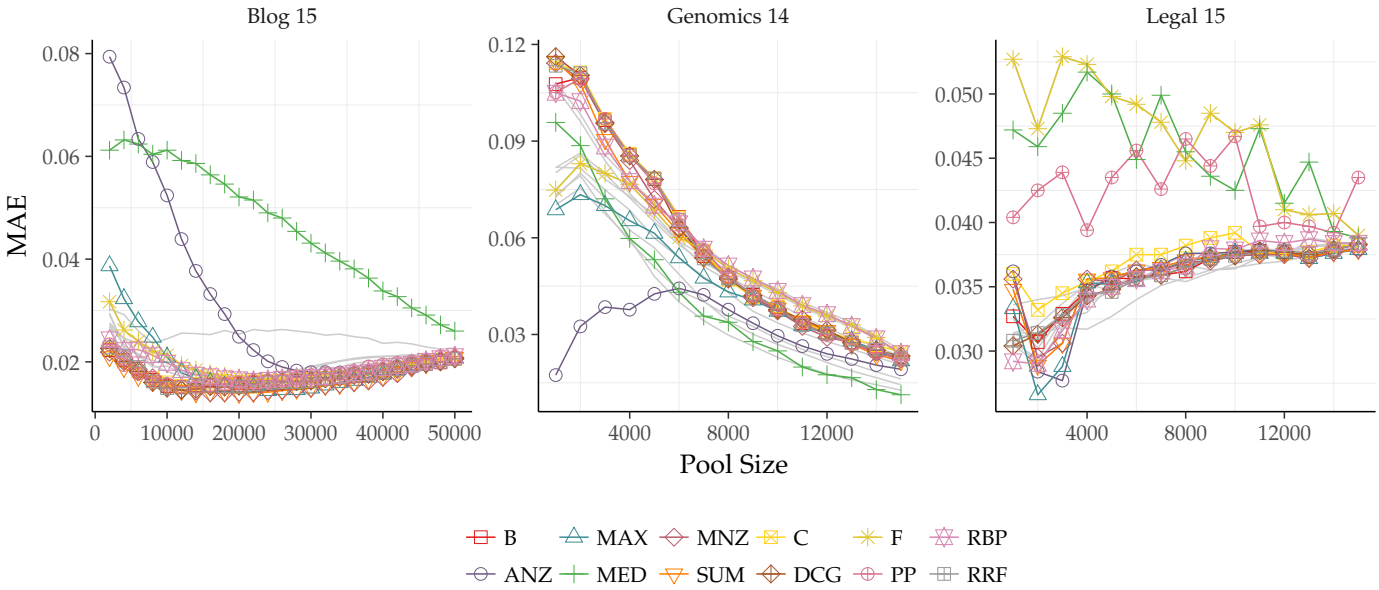

Fig. 5. Pool bias measured for the *non-adaptive* pooling strategies in terms of the measure of bias MAE and IR evaluation measure AP, and for different pool sizes (i.e. aggregated number per topic of documents that require relevance judgment). The lines in gray are the *adaptive* pooling strategies (in Fig. 6) for comparison.

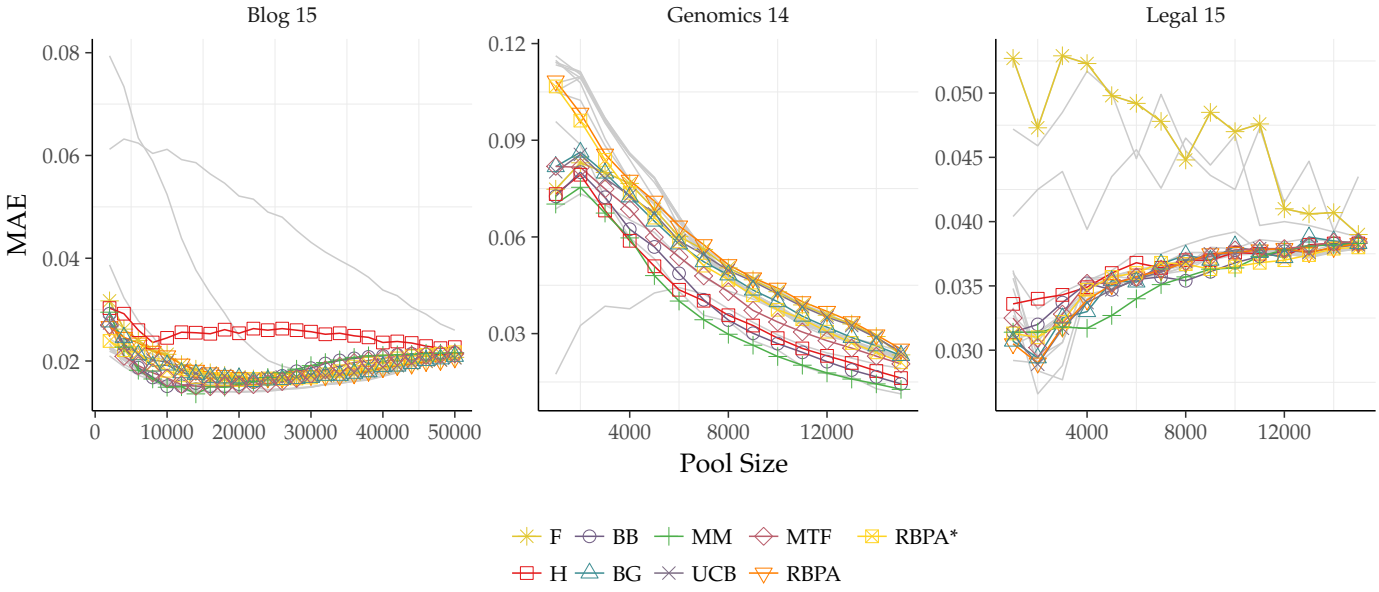

Fig. 6. Pool bias measured for the *adaptive* pooling strategies in terms of the measure of bias MAE and IR evaluation measure AP, and for different pool sizes (i.e. aggregated number per topic of documents that require relevance judgment). The lines in gray are the *non-adaptive* pooling strategies (in Fig. 5) for comparison.

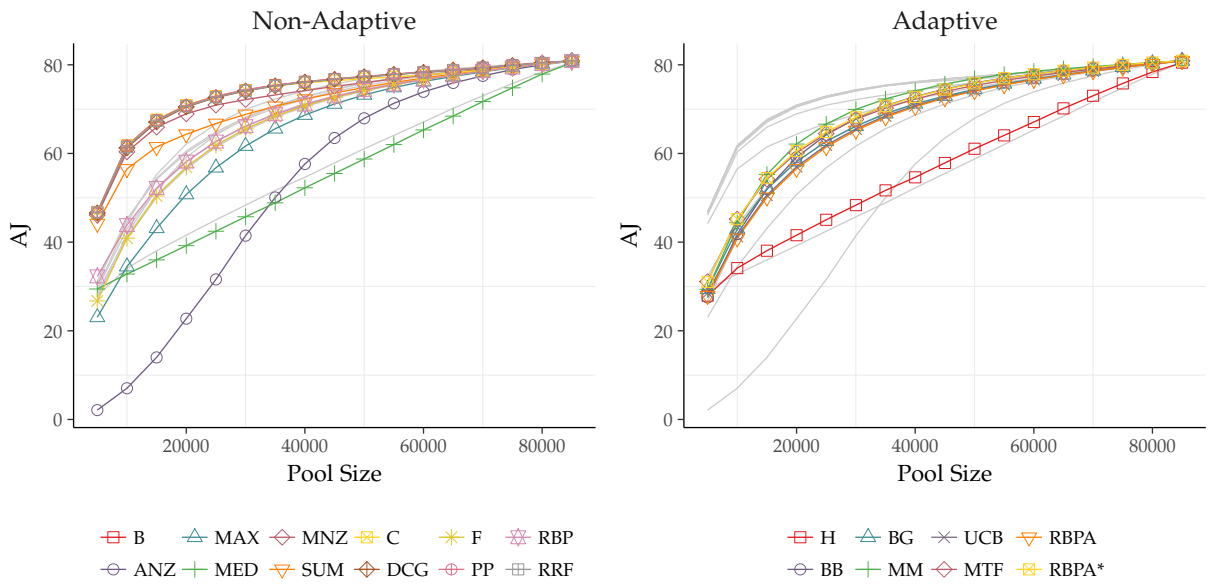

Fig. 7. Expected number of judged documents for the pair run-topic (AJ), for non pooled runs tested on Ad Hoc 8 test collection against all *adaptive* pooling strategies (on the right) and against all *non-adaptive* pooling strategies. Similar results are observed for the rest of the test collections. This plot is in function of the different pool sizes (i.e. aggregated number per topic of documents that require relevance judgment). The lines in gray are, on the left, the *adaptive* pooling strategies and, on the right, the *non-adaptive* pooling strategies for comparison.

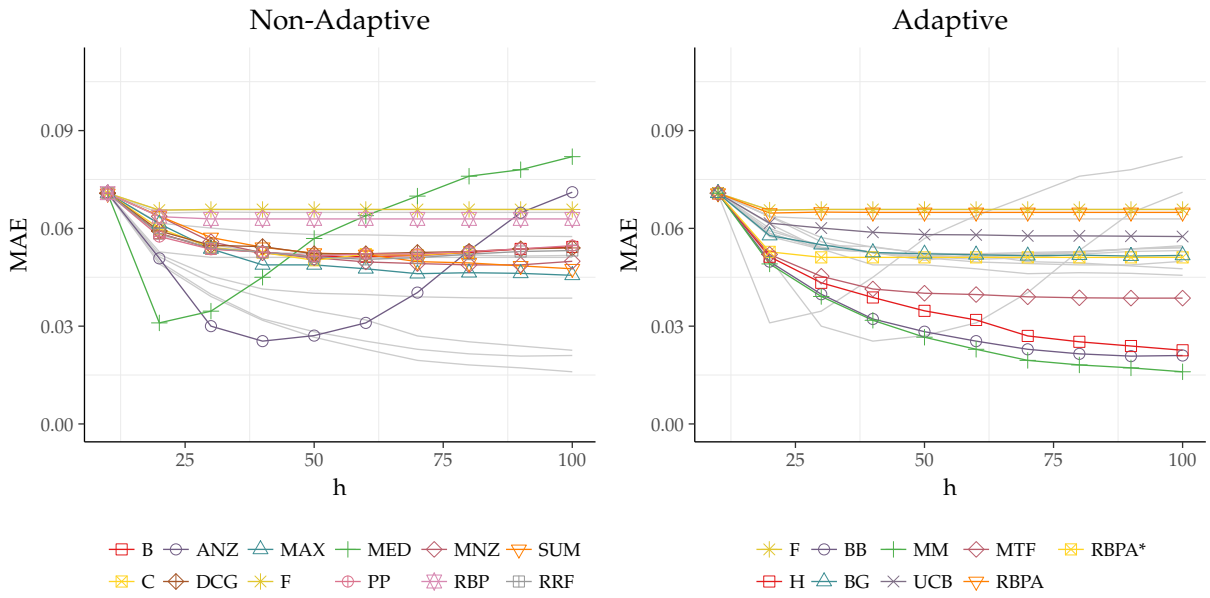

Fig. 8. Pool bias measured, for the *non-adaptive* pooling strategies on the left, and *adaptive* pooling strategies on the right, in terms of the measure of bias MAE and IR evaluation measure AP, for a fixed pool size $N = 10,000$ and for different horizons (i.e. the size of the runs available to the pooling strategies) for the test collection Ad Hoc 8. The lines in gray are, on the left, the *adaptive* pooling strategies and, on the right, the *non-adaptive* pooling strategies for comparison.

guarantees full control over the number of documents to be assessed. However, a direct comparison between these two pooling strategies has shown little difference in term of pool bias for the IR evaluation measures investigated in this paper.

We start our discussion analyzing the operationability of a pooling strategy. Next, we make a general observation about the results. Then, we focus on the non-adaptive strategies. Following that, we analyze the adaptive ones. Finally, we compare them to each other.

5.1 Pooling Operationability

The operationalization of a pooling strategy refers to the flexibility that a strategy gives to the test collection builder in gathering the relevance assessments. If a pooling strategy does not impose a constraint on how to gather this information, then we say that this pooling strategy is *operationalizable*. The advantage of such strategies is that the two processes, pooling and assessing of the documents, are independent. This lack of interdependency, since the assessments are performed by human beings, makes it easier to tackle the cognitive biases that may affect the assessors while performing the judgments. The standard way to address these biases is to make the assessors judge a randomized sample of the pooled documents. In general we identify the following operationability properties of a pooling strategy: aggregable, ordinable, and parallelizable. In the following discussion we will be primarily concerned with distinguishing those pooling strategies that do not have one or more of these properties.

A pooling strategy is *aggregable* when the collection builder is able to aggregate relevance assessments for a document across judgments from *multiple assessors*. Pooling strategies that do not present this quality put an additional burden on the collection builder. This is because these strategies require information about the relevance of documents already assessed to decide which documents to pool next. Thereby a non aggregable strategy requires that the assessment process is coordinated such that the assessment and selection of the next document to assess cannot start until all assessors have judged the current document: this may happen at different times due to different assessor cognitive abilities, workload, and work scheduling.

A pooling strategy is *ordinable* when the collection builder is able to control in which order the relevance assessments are performed. The absence of such a property may introduce cognitive biases. For example, some pooling strategies may favor such a bias because it requires the judgment of documents in order of their predicted relevance. This bias is instead usually overcome by the ordinable pooling strategies by randomizing the pooled documents before presenting them to the assessors.

For the *parallelizable* property of a pooling strategy we can distinguish two parallelization forms, cross-topic and per-topic parallelizations. The former refers to parallelizing the assessments by judging at the same time multiple topics, i.e. exclusively assign each topic to an assessor, but assigning different topics to different assessors. The latter refers to parallelizing, given a topic, the assessments for this topic, i.e., distributing documents that are retrieved for the same topic across multiple assessors to speed up the assessment process. While the former is often possible, the latter, always preferable, is sometimes difficult to obtain.

In general, all non-adaptive pooling strategies are aggregable, ordinable, and fully parallelizable; for the adaptive pooling strategies, all but RBPAdaptiveTake@N are only cross-topic parallelizable.

5.2 General Observation about the Results

In Figures 5 and 6 we can observe that for some test collections like Blog 15 and Legal 15 (and also for the omitted Web 23), the measured bias increases when increasing the number of judged documents N, notably also for the FairTake@N. This behavior does not exist when tested on P@10, and it has to be because of the recall component of the measures AP and NDCG. While this is apparently disturbing, in fact, for the purposes of selecting which strategy to apply in the future, it does not change our conclusions.

5.3 Non-Adaptive Strategies

Among the voting system-based strategies, we observe that BordaTake@N performs slightly better than the CondorcetTake@N in all evaluation measures, although BordaTake@N is a relaxation of CondorcetTake@N. Both strategies are better than FairTake@N when used with P@10 and only initially worse when used for AP and NDCG. In a previous work [53], the pooling strategy CondorcetTake@N was poorly performing and as stated in this work it was not as easily predictable. CondorcetTake@N has the issue that when comparing pairs of documents, if the two are not in the top K of the run, it neither adds nor subtracts anything from the value this strategy computes for the pair. This may lead to situations where it is impossible to compute a complete ordering of documents, e.g., in the situation where a document d_i is preferred to d_j, d_j to d_k, and also d_k to d_i. To bypass this theoretical limitation, Lipani et al. [53] followed the work of Montague and Aslam [24] by implementing a sorting method that avoids this limit case, but also does not guarantee an optimal result (compare Algorithms 3 and 2 in Montague and Aslam [24] paper), while in this article a better solution was found by using Copeland's method.

Among the retrieval fusion based-pooling strategies, as expected, we observe a poor performance of the CombMINTake@N strategy. In fact it clearly performs worse than the FairTake@N baseline across all IR evaluation measures and measures of bias for all test collections. The strategy CombMINTake@N prefers the lowest scoring documents and is therefore likely to identify mostly non-relevant items, making the final (evaluation) scores highly unstable. This happens also to CombMEDTake@N for all but one test collection (Microblog 20). The strategy CombANZTake@N usually performs poorly with all measures of bias except when computed on the IR evaluation measure NDCG. The strategy CombMAXTake@N performs consistently better than the baseline with all the IR evaluation measures but one, P@10. The strategies CombMNZTake@N and CombSUMTake@N behave similarly across both evaluation and bias measures. These strategies are better than FairTake@N when used with P@10 and only initially worse when used for AP and NDCG.

Among the evaluation measure based-pooling strategies, DCGTake@N, PPTake@N, and RRFTake@N correlate with each other, while RBPTake@N does not. They all tend to be better than the baseline only for P@10 and worse initially for NDCG and AP. RBPTake@N is the most conservative. The linear and logarithmic discounts remove the rank information from the documents rewarding more popular documents amongst the runs. The relationship between the discount and the top-heaviness of the evaluation measures AP and NDCG also explains the twist in preference, where FairTake@N is preferred for low N, then for higher N almost all non-adaptive methods outperform it, before they all converge to the same value. For P@10 we observe that DCGTake@N, RRFTake@N, and PPTake@N are the best, followed by RBPTake@N.

Juxtaposing all the non-adaptive strategies we observe that the voting system-based strategy BordaTake@N behaves similarly to the retrieval fusion method-based strategy CombMNZTake@N; and voting system-based strategies and IR evaluation measure-based strategies partially correlate with the retrieval fusion method-based strategy CombSUMTake@N.

For the non-adaptive pooling strategies we can conclude that most stable strategy is CombMAXTake@N. However, if the measure to be optimized is only P@10 DCGTake@N should be preferred. This is clearly visible in Figure 3 and in Table 1. However, although a selected non-adaptive pooling strategy performs better than the baseline, the collection builder, at the cost of losing some operationability properties, can move to lesser biased pooling strategies in the next category, the adaptive ones.

5.4 Adaptive Pooling Strategies

Between the two classic pooling strategies we observe that the traditional MTFTake@N pooling strategy outperforms the baseline in every evaluation measure and test collections. This strategy is one of the most stable pooling strategies across IR evaluation measures, and on average discovers over 25% of relevant documents more than the baseline. The HedgeTake@N strategy outperforms MTFTake@N in all IR evaluation measures but P@10, and in all test collections but Blog 15 where HedgeTake@N fails for AP and NDCG when compared against FairTake@N. We can observe that although HedgeTake@N discovers on average 27% more relevant documents than the baseline, it not effective in reducing the bias. This happens in the case of Blog 15 where the strategy is worse than the baseline. The reason for this failure has to be found in the parameter β that has been trained using test collections with a lower rate of relevant documents. In fact we predicted that increasing β from 0.1 to 0.9 would have increased the performance of HedgeTake@N to become higher than the baseline. This can be observed by the fact that when $\beta = 1$ this strategy reduces to an unbounded RRFTake@N like strategy (see Electronic Appendix 2), whose performance for AP is better than the baseline.

Between the two IR evaluation measure-based pooling strategies we observe that the performance of the RBPAdaptiveTake@N strategy is comparable to the FairTake@N. The RBPAdaptive*Take@N strategy outperforms the baseline in every evaluation measure and test collection.

Among the multi-armed bandit-based strategies the MABUCBTake@N strategy performs comparably to the FairTake@N strategy. Among MABGreedyTake@N, MABBetaTake@N, MABMaxMeanTake@N, they all outperform the baseline for all IR evaluation measures and bias measures. In particular MABMaxMeanTake@N is the best performing pooling strategy in terms of bias.

Comparing all the adaptive pooling strategies, we observe that RBPAdaptive*Take@N, MTFTake@N, MABGreedyTake@N, and MABMaxMeanTake@N are always better than the baseline for every IR evaluation measure. For the adaptive pooling strategies we can draw the following conclusion: the least biased pooling strategy is MABMaxMeanTake@N. It is interesting to observe that this pooling strategy is the one that discovers the highest number of relevant documents, above 45% more than the baseline.

5.5 Non-adaptive vs. Adaptive Pooling Strategies

We now consider all the tested pooling strategies together. We observe that the best pooling strategy is MABMaxMeanTake@N for all test collections. However if some operationalization properties are required, the CombMAXTake@N should be preferred. Overall the adaptive pooling strategies demonstrate to be more stable across IR evaluation measures. In fact RBPAdaptive*Take@N, MTFTake@N, MABGreedyTake@N, and MABMaxMeanTake@N always perform better than the baseline.

5.6 Accuracy and Stability of the Results

As discussed in Section 4.1, this experimental design raises two potential issues.

About the inconclusiveness of the results due to having too few documents judged in the non-pooled runs, Figure 7 tells us, indeed, about the accuracy of the computation of the term, $f(r, J_{\mathcal{R}_p \setminus \{r' \in \mathcal{R}_p : o_{r'} = o_r\}})$, which is present in all three bias measures. For example, if we consider the non-adaptive pooling strategy CombANZTake@N, we observe that for Ad Hoc 8, with a pool size $N = 5,000$, the expected number of documents judged per run per topic is around 2.10, which means that when computing an IR evaluation measure on these non pooled runs, their accuracy is probably compromised. However, because such pool sizes are still used, and there are no available guidelines in the literature on how many judged documents are really necessary, we chose to provide these plots to let the readers assess the results by themselves.

About the stability of the results when changing the horizon of the pooling strategies, Figure 8 shows that all the best pooling strategies but two are consistent with the results discussed above. In fact, the best strategies continue to be the best also when changing horizon. The two pooling strategies that show an unstable behavior are CombMEDTake@N and CombANZTake@N, which favor lower horizons. This experiment shows that the pooling strategies are stable when increasing their horizon. Based on this observation, we expect them to be consistent when increasing their horizon beyond the tested one.

5.7 Limitations

There are still a number of limitations and possible extensions to this work. First and foremost, we are constrained by the data available to us. As we have detailed in the beginning of Section 4, we do not see an alternative to a proper investigation of pool bias without "cleaning" the test collections and generating runs that have no documents beyond what we know to be evaluated. Nevertheless, this does significantly reduce the "knowledge" available to us as we have to discard a non-negligible percentage of the ground-truth. We see addressing this as a significant effort, to be perhaps undertaken as a new evaluation effort in TREC. Our study would hopefully serve as a initial step, to identify those pooling strategies that should be further tested in the context of such a large scale evaluation exercise.

Beyond this, a limitation that has appeared as we were presenting the various pooling strategies is the setting of their parameters. Throughout the paper we have considered only parameters that have been published in previous works, but often enough these parameters were used for different purposes (retrieval fusion methods, IR evaluation measures) and maybe different values would be better fitted for pooling strategies.

There are a number of decisions that are taken in every evaluation campaign, that complement the pooling strategy itself. The number of runs, the number of topics, the distribution of evaluation effort over topics are all elements that are worth further investigation in relation to the pooling strategy. Finally, as the title clearly indicates, we focus here on pools of a fixed size. While this is often a real-world constraint, the study of variable-sized pools and the balance between the effort to assess another document and the bias reduction expected from this effort is equally worth pursuing.

6 Conclusion

In this article we have explored a large array of pooling strategies, from the standard Depth@K (closely approximated here by FairTake@N in the context of fixed-cost pooling) to recent strategies based on voting systems, retrieval fusion methods, IR evaluation measures, and multi-armed bandits methods. In doing so, we have observed parallels between strategies that had been developed independently (e.g. BordaTake@N and CondorcetTake@N, or HedgeTake@N and RRFTake@N) and distinguished between adaptive and non-adaptive pooling strategies, with their different operationalizations.

The baseline, FairTake@N remains a solid candidate, but it can be improved upon. If we have constraints on operationalization and are therefore mandated to use a non-adaptive method, then CombMAXTake@N (using the maximum score obtained by a document across the runs) would be recommended, particularly when top-heavy measures like AP and NDCG are the target evaluation measures. There is one exception, the Blog 15 test collection, where the RBPTake@N provided better results. However, the Blog 15 collection is quite unusual: compared to all other test collections, it has an extremely high percentage of relevant documents being judged and the runs are very diverse (because topical relevance was not the main objective of the evaluation in that collection). However, for all test collections, if the measure to be optimized is P@10, DCGTake@N should be preferred.

If, however, adaptive pool generation is operationalizable (i.e. including feedback from assessors in the pool generation process), we should use a multi-armed bandit-based method, MABMaxMeanTake@N, which is the least biased among all the tested pooling strategies; moreover, it is the strategy that discovers the highest number of relevant documents, on average 45% more than the baseline.

In the course of this study we have also observed that the ability of a pooling strategy in discovering a high number of relevant documents is somewhat correlated with the less biased ones, but not completely, e.g. the best non-adaptive strategy, CombMAXTake@N, discovers a number of relevant documents comparable to the baseline but performs better in terms of bias than other non-adaptive strategies that discover on average even more than 15% relevant documents than the baseline. This verifies the statement made by Spärck Jones about the aim of the pooling strategy: a pooling strategy's objective is not to discover the highest number of relevant documents, but to discover an unbiased set of relevant documents [7].

Acknowledgments

Aldo Lipani is funded by the EPSRC Fellowship (EP/P024289/1). David E. Losada thanks the financial support obtained from (i) the "Ministerio de Ciencia, Innovación y Universidades" ("Agencia Estatal de Investigación") of the Government of Spain (research grant RTI2018-093336-B-C21) and the (ii) "Consellería de Educación, Universidade e Formación Profesional", Xunta de Galicia (grants ED431C 2018/29 and ED431G/08). All grants were co-funded by the European Regional Development Fund (ERDF/FEDER program). Mihai Lupu is partially funded by the Data Market Austria Project (FFG Project Number 855404).

References

[1] E. Voorhees and D. Harman, "Overview of the eighth text retrieval conference," in *Proc. of TREC-8*.
[2] B. Koopman and G. Zuccon, in *Medical Information Retrieval Workshop at SIGIR '14*.
[3] E. M. Voorhees and D. K. Harman, *TREC: Experiment and Evaluation in Information Retrieval*. The MIT Press.
[4] T. Sakai, *Laboratory Experiments in Information Retrieval*, 2018.
[5] K. Spärck Jones and C. J. van Rijsbergen, "Report on the need for and provision of an "ideal" information retrieval test collection," University of Cambridge, Tech. Rep.
[6] D. Harman, "Overview of the first trec conference," in *Proc. of SIGIR '93*.
[7] K. Spärck Jones, "Letter to the editor," *Information Processing & Management*, vol. 39.
[8] M. Sanderson, "Test collection based evaluation of information retrieval systems," *Foundations and Trends® in Information Retrieval*.
[9] G. V. Cormack, C. R. Palmer, and C. L. A. Clarke, "Efficient construction of large test collections," in *Proc. of SIGIR '98*.
[10] S. Büttcher, C. L. A. Clarke, P. C. K. Yeung, and I. Soboroff, "Reliable information retrieval evaluation with incomplete and biased judgements," in *Proc. of SIGIR '07*.
[11] W. Webber and L. A. F. Park, "Score adjustment for correction of pooling bias," in *Proc. of SIGIR '09*.
[12] A. Lipani, M. Lupu, and A. Hanbury, "Splitting water: Precision and anti-precision to reduce pool bias," in *Proc. of SIGIR '15*.

[13] A. Lipani, M. Lupu, E. Kanoulas, and A. Hanbury, "The solitude of relevant documents in the pool," in *Proc. of CIKM '16*.
[14] A. Lipani, G. Zuccon, M. Lupu, B. Koopman, and A. Hanbury, "The impact of fixed-cost pooling strategies on test collection bias," in *Proc. of ICTIR '16*.
[15] D. E. Losada, J. Parapar, and A. Barreiro, "Feeling lucky?: Multi-armed bandits for ordering judgements in pooling-based evaluation," in *Proc. of SAC '16*.
[16] A. Lipani, J. Palotti, M. Lupu, F. Piroi, G. Zuccon, and A. Hanbury, *Fixed-Cost Pooling Strategies Based on IR Evaluation Measures*.
[17] S. Robertson, "On the history of evaluation in ir," *Journal of Information Science*.
[18] C. Buckley, D. Dimmick, I. Soboroff, and E. Voorhees, "Bias and the limits of pooling for large collections," *Information Retrieval*, 2007.
[19] J. Zobel, "How reliable are the results of large-scale information retrieval experiments?" in *Proc. of SIGIR '98*.
[20] A. Lipani, "Fairness in information retrieval," in *Proc. of SIGIR '16*.
[21] A. Moffat, W. Webber, and J. Zobel, "Strategic system comparisons via targeted relevance judgments," in *Proc. of SIGIR '07*.
[22] E. Yilmaz, E. Kanoulas, and J. A. Aslam, "A simple and efficient sampling method for estimating ap and ndcg," in *Proc. of SIGIR '08*.
[23] J. A. Aslam and M. Montague, "Models for metasearch," in *Proc. of SIGIR '01*.
[24] M. Montague and J. A. Aslam, "Condorcet fusion for improved retrieval," in *Proc. of CIKM '02*.
[25] C. Macdonald, "The voting model for people search," *SIGIR Forum*.
[26] A. Lipani, M. Lupu, and A. Hanbury, *The Curious Incidence of Bias Corrections in the Pool*.
[27] W. B. Croft, *Combining Approaches to Information Retrieval*, 2000.
[28] J. H. Lee, "Analyses of multiple evidence combination," in *Proc. of SIGIR '97*.
[29] M. Montague and J. A. Aslam, "Relevance score normalization for metasearch," in *Proc. of CIKM '01*.
[30] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Trans. Inf. Syst.*
[31] S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng, "Web question answering: Is more always better?" in *Proc. of SIGIR '02*.
[32] G. V. Cormack, C. L. A. Clarke, and S. Buettcher, "Reciprocal rank fusion outperforms condorcet and individual rank learning methods," in *Proc. of SIGIR '09*.
[33] C. Buckley and E. M. Voorhees, "Evaluating evaluation measure stability," in *Proc. of SIGIR '00*.
[34] T. Sakai, "New performance metrics based on multigrade relevance: Their application to question answering." in *Proc. of NTCIR '04*.
[35] A. Moffat and J. Zobel, "Rank-biased precision for measurement of retrieval effectiveness," *ACM Trans. Inf. Syst.*, 2008.
[36] B. Carterette, "System effectiveness, user models, and user utility: A conceptual framework for investigation," in *Proc. of SIGIR '11*.
[37] L. A. Park and Y. Zhang, "On the distribution of user persistence for rank-biased precision," in *Proceedings of the 12th Australasian document computing symposium*, 2007.
[38] Y. Zhang, L. A. F. Park, and A. Moffat, "Click-based evidence for decaying weight distributions in search effectiveness metrics," *Information Retrieval*.
[39] J. A. Aslam, V. Pavlu, and R. Savell, "A unified model for metasearch, pooling, and system evaluation," in *Proc. of CIKM '03*.
[40] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *IEEE Transactions on Neural Networks*.
[41] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*
[42] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, 1933.
[43] D. Harman, "Overview of the third text retrieval conference (TREC-3)," in *Proc. of TREC '94*.
[44] E. M. Voorhees and D. Harman, "Overview of the eight text retrieval conference (TREC-8)," in *Proc. of TREC '99*.
[45] D. Hawking, "Overview of the trec-9 web track," in *Proc. of TREC '00*.
[46] K. Collins-Thompson, C. Macdonald, P. Bennett, F. Diaz, and E. M. Voorhees, "Trec 2014 web track overview."
[47] E. M. Voorhees, "Overview of the TREC 2005 robust retrieval track."
[48] W. Hersh, A. Cohen, J. Yang, R. T. Bhupatiraju, P. Roberts, and M. Hearst, "TREC 2005 genomics track overview."
[49] J. R. Baron, D. D. Lewis, and D. W. Oard, "TREC 2006 legal track overview," in *Proc. of TREC '06*.
[50] I. Ounis, M. de Rijke, C. Macdonald, G. Mishne, and I. Soboroff, "Overview of the TREC-2006 blog track," in *Proc. of TREC '06*.
[51] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff, "Overview of the TREC-2011 microblog track," in *Proc. of TREC '11*.
[52] E. M. Voorhees and C. Buckley, "The effect of topic set size on retrieval experiment error," in *Proc. of SIGIR '02*.
[53] A. Lipani, M. Lupu, J. Palotti, G. Zuccon, and A. Hanbury, "Fixed budget pooling strategies based on fusion methods," in *Proc. of SAC '17*.

Aldo Lipani is a Lecturer in Machine Learning at the University College London (UCL). Previously, he was, also at UCL, a Postdoctoral Research Associate in the group of Prof. Emine Yilmaz. He holds a BSc from the University of Catania, Italy and an MSc from the University of Bologna, Italy both in Computer Engineering. He earned his Ph.D. in Computer Science at the TU Wien, Austria, under the supervision of Prof. Allan Hanbury and Dr. Mihai Lupu. Aldo has furthered his studies at the National Institute of Standard and Technologies (NIST) in Gaithersburg, Microsoft Research Cambridge, University of Glasgow, University of Amsterdam, and National Institute of Informatics (NII) in Tokyo.

David E. Losada is an Associate Professor in Computer Science & Artificial Intelligence at CiTIUS (University of Santiago de Compostela, Spain). His current research interests include a wide range of Information Retrieval (IR) and related areas such as: early risk detection, text mining, IR evaluation, IR probabilistic models, summarization, novelty detection, and sentence retrieval. Losada is an active member of the IR community and he regularly serves in the Programme Committee of prestigious international conferences such as SIGIR or ECIR. In 2011, Losada was recognized with an ACM senior member award.

Guido Zuccon is a Senior Lecturer in Information Retrieval in the School of Information Technology and Electrican Engineering, at the University of Queensland (Australia), and an ARC Discovery Early Career Researcher Award Fellow and a Google Faculty Award recipient. Guido received his B.Eng. and M.Eng. (summa cum laude) at the University of Padua (Italy) and a Ph.D. in Computing Science from the University of Glasgow (U.K.). Before joining the University of Queensland, Guido was a senior lecturer at the Queensland University of Technology (Australia) and a postdoctoral research fellow at the Australian E-Health Research Centre, CSIRO (Australia). His research interests include formal models of search and evaluation methods, in particular applied to health search.

Mihai Lupu is, since January 2018, the Studio Director of the Data Science Studio at Research Studios Austria Forschungsgesellschaft. Before that he has been a researcher at the TUWien as well as a private entrepreneur, consulting small and large companies on search technology, with focus on search in the intellectual property domain. He graduated from the Singapore-MIT Alliance in 2008 and since then has published over 100 publications and three books on search technology. He is now the co-coordinator of Data Market Austria and the Coordinator of the H2020 Safe-DEED project.