Fixed Budget Pooling Strategies based on Fusion Methods

anonymized

ABSTRACT

The empirical nature of Information Retrieval (IR) mandates strong experimental practices. The Cranfield/TREC evaluation paradigm represents a keystone of such experimental practices. Within this paradigm, the generation of relevance judgments has been the subject of intense scientific investigation. This is because, on one hand, consistent, precise and numerous judgements are key to reduce 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) has therefore great impact in IR evaluation. In this paper, we contribute a set of 8 novel pooling strategies based on retrieval fusion strategies. 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 best performing pooling strategy according to three evaluation metrics.

Keywords

Pooling Method, Pooling Strategies, Pool Bias

1. INTRODUCTION

IR systems are primarily evaluated for effectiveness against a benchmark, a test collection consisting of a predefined, fixed set of documents, a set of topics (information needs), and a set of relevance judgments for pairs of topics and documents.

This set of 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 TREC-8 Ad Hoc test collection [21]), the total relevance judgments to be made would be 5×10^6. At a very optimistic rate of 120 seconds/judgment, this represents the equivalent of 95 years of work for one person [9]. 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 engines [22].

Pooling fundamentally relies on the assumption that if sufficiently many and sufficiently diverse systems participate in a pool (i.e., provide lists of documents they consider to be relevant for each topic), a set of <topic, document> pairs can be identified that, once evaluated, will be 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 Spark-Jones and van Rijesbergen [7], 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 hidden implication 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 [8].

Since the proposal of *Depth@k* pooling strategy, substantial research effort has gone into improving the evaluation procedure, to reduce the cost and increase the reliability of the test collections, including by devising alternative pooling strategies, e.g. [4, 17, 3, 11, 13, 23, 15, 14].

Reliability is understood here as the opposite of *bias* in a test collection. 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 actually 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 [20]. Incomplete judgments, i.e., the presence among the retrieved results of unpooled 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 the collections unreliable [2, 25].

The research effort in this area channeled in two directions: On one hand, prior work has attempted to reduce bias at test collection creation time, by considering different pooling strategies [4, 17, 3]. On the other hand, for already existing test collections, other work has attempted to adopt metrics that reduce the effect of the bias [11, 13, 23]. Sometimes, the two directions intertwine and a new pooling method is proposed together with a matching evaluation metric [24], but that significantly restricts the future use of the collection to specific metrics.

In this paper, we focus on the first type of approach, ex-

ploring different pooling strategies to identify the most efficient way to create the pool, while controlling the bias. Recently, Losada et al. [15] have considered a new perspective on pool creation, using a multi-armed bandit — an established method for resource use optimization. The current study complements the previous work by exploring a set of eight resource selection strategies, in addition to the traditional *Depth@k* pooling strategy.

The remainder of the paper is structured as follows: the following section briefly describes each of the pooling strategies analyzed. Section 3 presents the experimental procedure and the results of the experiments. These are discussed in Section 4. We conclude in Section 5.

2. POOLING STRATEGIES

We examine each of the pooling strategies that we empirically investigate in this paper as alternative to the original *Depth@k* strategy. Apart from *Take@N* (which is an alternative approach commonly used in IR), the other pooling strategies below are new to IR, although the underlying intuitions have been extensively used in IR as retrieval and fusion methods [1, 19, 16]. These new 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 to determine the amount of voting to cast towards that document, and (2) majoritarian voting systems that base the preference for a document based on pairwise comparisons between candidate documents.

The pooling strategies that we investigate in this paper are reported below; note that here we consider pools formed by exactly N documents, but the methods may be further generalized to other stopping criteria (left for future work). The constraint used here is motivated by the fact that we aim to study pooling when a fixed amount of budget is available for collecting relevance judgments (i.e., the budget to judge N documents), a constraint that is typical in most IR evaluation exercises like TREC, CLEF and NTCIR.

Take@N (strategy T): This strategy is based on the rank at which documents have been retrieved. The strategy starts by assigning to every retrieved document d the highest rank ρ at which d has been retrieved by a contributing IR system. Then, the N documents with the highest ρ are selected and included in the pool, so that the pool can be specified according to a fixed pool size (N). Compared to *Depth@k* pooling, this strategy presents a drawback in that it does not guarantee fairness among all the pooled runs. In fact, with *Depth@k* all runs contribute equally to the pool with their first k documents, while with *Take@N* some runs can express more documents in the pool than others.

BordaTake@N (strategy B): This is a positional voting strategy where candidate documents to be pooled are ranked in order of preference: each document is assigned a number of votes corresponding to the sum of the rank positions at which that document has been retrieved by the different systems. To determine preference for pooling, documents are ordered in increasing order of the sum of the ranks (the lower the sum of the ranks, the higher the pooling preference). The top N documents are finally pooled. *BordaTake@N* is different from *Depth@k* in that it considers the sum of all ranks at which a document has been retrieved, while *Depth@k* only considers the highest rank (the earliest rank).

CondocertTake@N (strategy C): This is a majoritarian voting strategy and ensures that pooled documents are those that, when compared to any not-pooled document, have been retrieved at higher ranks by more systems. Indeed, strategies that guarantee this condition satisfy the *Condocert criterion*: as such, it is easy to demonstrate that *Depth@k*, *Take@N* and *BordaTake@N* do not satisfy this condition. Specifically, this strategy first forms a list containing the set of all documents retrieved by the pooled systems. Then, it sorts the list according to the following procedure: Compare each document pair d_i and d_j. Iterate through the document rankings of each system and increment a counter if d_i is ranked above d_j (or decrement 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.

CombMAXTake@N (strategy MAX): In general, a document may be retrieved by multiple systems, and this likely happens with different scores. For each topic, this strategy only considers the maximum retrieval score that has been assigned by any system to a specific document. After constructing a new document ranking with the combined scores from multiple runs, the strategy Take@N is applied, i.e., only the documents with the highest N scores are included in the pool. Document scores are normalized across each topic and each system run, mapping the highest score of a document for a topic to 1 and the smallest to 0, as recommended in prior work that examined fusion methods for retrieval [1, 5, 10, 18]. The CombMAX retrevial fusion strategy that shares the same underlying intuition of *CombMAXTake@N* is a commonly used strong baseline in the IR literature that investigates fusion strategies for retrieval.

CombMINTake@N (strategy MIN): While *CombMAXTake@N* minimizes the number of relevant documents being poorly ranked, the purpose of *CombMINTake@N* is to minimize the probability that a non-relevant document would be ranked at early ranks. This strategy also combines the scores from different runs, as the other fusion-based strategies. The only practical difference between *CombMAXTake@N* and *CombMINTake@N* is that the former uses the maximum score, while the latter uses the minimum score.

CombMEDTake@N (strategy MED): This strategy takes a middle-ground approach to the selection of pooling documents based on fusion, by selecting the median score of the list of all document scores returned by systems for a topic (as opposed to the maximum or minimal score as in *CombMAXTake@N* and *CombMINTake@N*, respectively).

CombSUMTake@N (strategy SUM): Instead of selecting one single score such done in *CombMAXTake@N*, *Comb-*

Test Collection Properties			
$\|R\|$:	129	$\|O\|$:	41
$\|R_p\|$:	66	$\|T\|$:	50
	Original		**Depth@100**
$\|Q\|$:	86.830		79.090
$\|Q_+\|$:	4.728		4.090

Table 1: Pool properties of the TREC-8 Ad Hoc test collection, for the original pool and the *Depth*@100 pool; $|R|$ number of runs; $|R_p|$ number of pooled runs; $|O|$ number of organizations; $|T|$ number of topics; $|Q|$ number of judged documents; and $|Q_+|$ number of relevant documents.

MINTake@N, and *CombMEDTake@N*, this strategy combines the sum of the scores of each document obtained for all participating systems.

CombANZTake@N (strategy 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).

CombMNZTake@N (strategy MNZ): This strategy aims to provide 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.

3. EXPERIMENTS & RESULTS

In this section we first present the material and experimental setup used in this paper, which follows the methodology set by prior work [11, 12, 23]; we then present the results.

3.1 Material & Experimental Setup

To test the effectiveness of the different pooling strategies we use the TREC-8 Ad Hoc test collection [21]. We selected this collection because of: 1) the large number of judged documents in the collection; 2) the large number of organizations that submitted system results that were used for pooling – we assume that the number of participating organizations is proportional to the variety of the submitted runs, and; 3) the pooling strategy used to build it, i.e., *fixed depth at cut off 100* pooling strategy (*Depth*@100). The latter makes it suitable for testing new pooling strategies that, by employing different sampling strategies, attempt to maximize the number of relevant documents while minimizing the overall number of judged documents. Note that when analysing the relevance assessments performed for this TREC-8 collection, we discovered that more than the expected number of judged documents were actually marked in the collection as being judged, i.e., the number of judged documents is larger than the pool formed by the submitted runs using the *Depth*@k (k=100) strategy. In order to ensure fairness between the pooling strategies investigated here, we rebuilt the relevance assessments for a clean *Depth*@100 pool. The pool properties are presented in Table 1.

We measure the pool bias provided by each pooling strategy using a *leave-one run-out* process, where for each run originally pooled we rebuild the test collection simulating the absence of that run. Next, given an IR evaluation measure, we measure the difference between the score obtained by the run when it is and is not part of the pool. Finally we compute the bias using the following three measure of bias, as in previous studies [11, 14, 13]: Mean Absolute Error (MAE); System Rank Error (SRE) and System Rank Error with Statistical Significance (SRE*, paired two-tailed t-test, with $p < 0.05$). MAE is the mean of the absolute value of the measured biases across the runs. SRE is the sum, across all the runs, of the absolute difference between the rank of the run when it is and is not pooled. SRE* is like SRE but it only considers results that are statistically significantly different.

To better simulate the case that the retrieval method used by the organization has not contributed to the pool, instead of the *leave-one run-out* evaluation we perform a *leave-one organization-out*, by removing at once all the runs submitted by an organization. This experimental procedure is a stronger indication of the presence of unfavorable bias towards specific retrieval models because of the implicit dependencies between runs that have been submitted by the same organization. In addition, due to the prototypical nature of the evaluation campaigns' challenges organized to build test collections, we filter out the bottom 25% of low performing runs from the bias measurement. This is because these runs are likely to contain bugs or very exploratory methods. This procedure is in line with previous studies [23, 11].

In the *leave-one organization-out* experiment, to avoid discovering non-judged documents in the original test collection, when re-pooling the selected runs with the tested pooling strategies we fixed the run sizes (i.e., the number of documents returned by a system) to 100, the depth of the pooling strategy used to construct the original test collection.

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 20 times varying the pool size from 5,000 to 100,000 at steps of 5,000.

As IR evaluation measures we use P@100, MAP, and $NDCG$ because these measures (a) are widely used in IR, and (b) encompass common features of most IR measures: top-heaviness, precision based, recall based, and utility based.

The software used in this paper to evaluate the proposed pooling strategies is available at *http://<anonymized>* along with the raw results of the empirical experiments.

3.2 Results

In Figure 1 we show the results obtained using the investigated pooling strategies. In the figure, 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 of bias. Every line is a pooling strategy.

From Figure 1 we can observe that all lines converge to a bias value of zero for large pool size values. This is because for a large enough pool, all alternative pooling strategies will

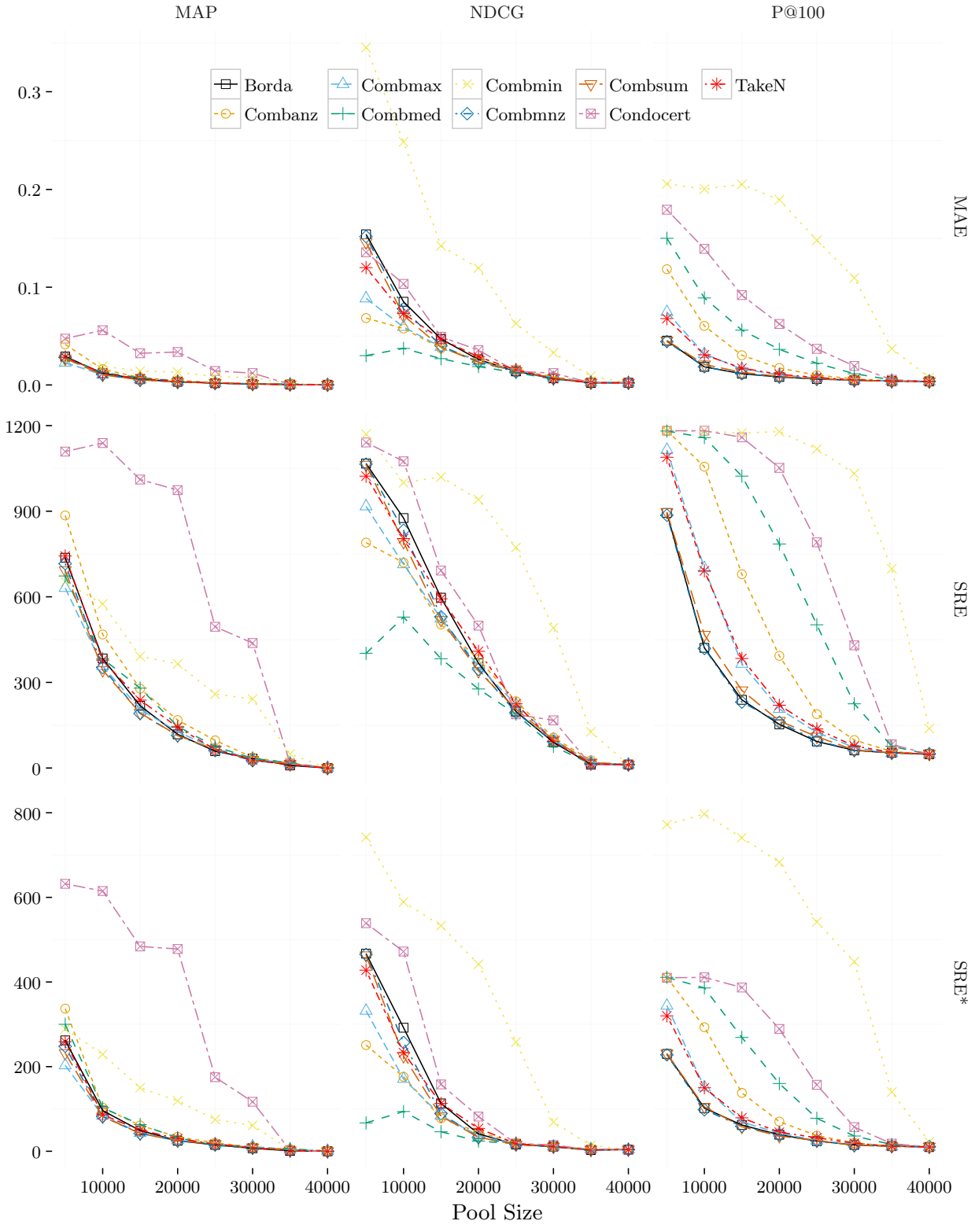

Figure 1: Pool bias measured in terms of MAE, SRE, and SRE* for the pooling strategies on the TREC-8 Ad Hoc test collection, for different pool sizes (i.e., number of documents that require relevance judgment.

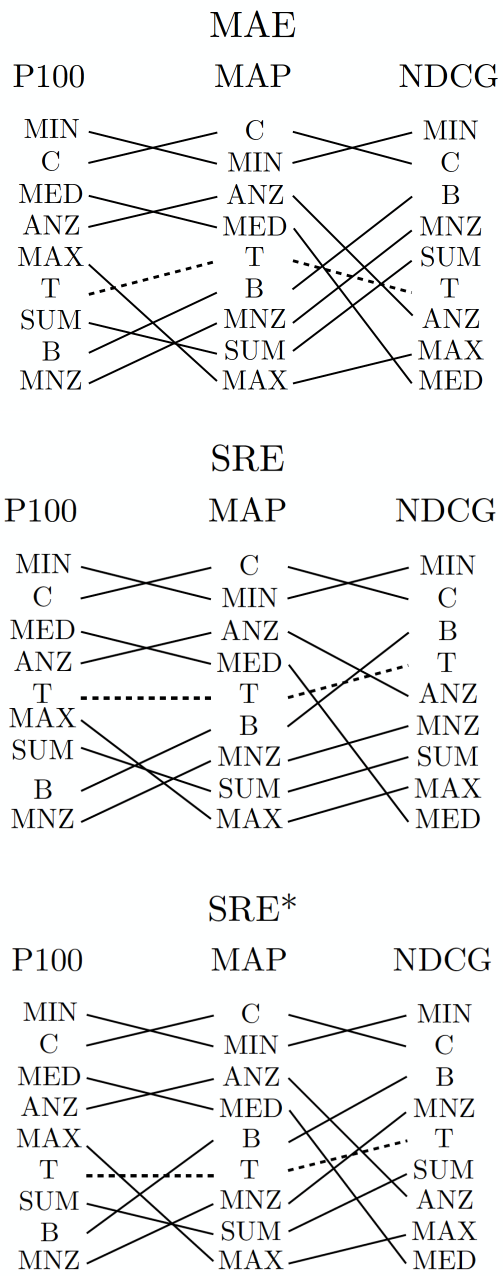

Figure 2: Ranking of the tested pooling strategies based on the average pool bias errors shown in Figure 1, sorted in descending order, from best on the bottom to worst on the top.

reduce to the *Depth*@100 strategy.

4. DISCUSSION

In the following discussion of the results reported in Figure 1, we refer to the *Take*@*N* strategy as our baseline. While this strategy is slightly different from *Depth*@*k* (see Section 2), *Take*@*N* is the pooling strategy closest to *Depth*@*k* that guarantees full control over the number of documents to be assessed.

#+	P@100	MAP	NDCG	MAE	SRE	SRE*
0	−	−	−	C MIN	C MIN	C MIN
1	+	−	−			B
	−	−	+	ANZ MED	ANZ MED	ANZ
2	+	+	−	B MNZ SUM	B	MNZ
	−	+	+	MAX		MAX MED
3	+	+	+		MAX MNZ SUM	SUM

Table 2: Summary of the cases when each pooling strategy is better than the baseline. The second column refers to IR evaluation measures (+ means better than the baseline and − worse); the third refers to pool bias measures (the values in the column explicitly state which measure is the best). The table is divided in categories (first column, indicated with #+): the categories refer to the number of times each pooling strategy is better than the baseline for each of the evaluation measures.

The CombMINTake@N and CondocertTake@N pool generation strategies clearly perform worse than the *Take*@*N* baseline across both evaluation metrics (MAP, NDCG, or P@100) and bias measures (MAE, SRE, and SRE*).

The poor performance of CombMINTake@N as a pooling strategy was to be expected, considering that, by definition, the strategy prefers the lowest scoring documents and is therefore likely to identify mostly non-relevant items, making the final (evaluation) scores highly unstable. The low performance of CondocertTake@N was perhaps not as easily predictable, but is reasonable. CondocertTake@N essentially prefers popular documents. CondocertTake@N also has another issue, whose effects are as yet unquantified. 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, we follow the work of Montague and Aslam [19] by implementing a sorting method that avoids this limit case, but also does not guarantee an optimal result (compare Algorithms 3 and 2 in [19]).

We can also observe that some pool generation strategies always outperform the baseline (but are not necessarily always the best). These are CombSUMTake@N, CombMAXTake@N, and CombMNZTake@N (see Table 2). The performance of the remaining pooling strategies vary when using different effectiveness and bias metrics. Their changes are shown in Figure 2.

Some pool generation strategies perform very differently when evaluating them using NDCG versus P@100. CombMEDTake@N is an example of such case. This strategy is the least biased when NDCG is considered, yet the most biased when P@100 is considered (just after Condocert@N and CombMINTake@N). The reason for this differnece in the behaviour of the pooling strategy can be traced back to the different nature of the two effectiveness metrics: NDCG in fact favors relevant documents at the top of the list while P@100 considers the list as an unordered set. CombMEDTake@N, like CombMAXTake@N and CombANZTake@N, will sample more from the top of the runs, thus, when NDCG is calculated, there is enough knowledge about relevant documents at the top of the lists to make the scoring stable.

Finally, we make an orthogonal observation to the pool generation strategies: the presented experiments demostrate once again the relative stability of mean average precision. Looking at the overall picture in Figure 1 we see that, with the exception of CombMINTake@N and CondocertTake@N, the MAP column is extremely tighter together than the NDCG and certainly than the P@100 columns. This indicates that the pool strategy has a smaller effect on MAP than on the other two metrics.

5. CONCLUSION

In this paper we have proposed and investigated a set of 8 new pooling strategies for fixed sized pooling inspired by ranking fusion strategies. The fixed sized pooling constraint allows to control relevance judgement costs. The experiments were conducted on the TREC-8 Ad-Hoc collection using three effectiveness metrics (MAP, NDCG, and P@100) and three bias metrics (MAE, SRE, and SRE*) and compared the proposed pooling strategies with the Take@N baseline. The results of the experiments show that some of the proposed strategies are always to be avoided (CombMINTake@N and CondocertTake@N), some always outperform the baseline (CombMAXTake@N, CombMNZTake@N, and CombSUMTake@N —though are not necessarily always the best), and the rest are dependent on the effectiveness metric used (e.g., CombMEDTake@N should be preferred if NDCG is used for measuring IR effectiveness).

6. REFERENCES

[1] J. A. Aslam and M. Montague. Models for metasearch. In *Proc. of SIGIR*, 2001.

[2] C. Buckley, D. Dimmick, I. Soboroff, and E. Voorhees. Bias and the limits of pooling for large collections. *Information retrieval*, 2007.

[3] S. Büttcher, C. L. Clarke, P. C. Yeung, and I. Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. In *Proc. of SIGIR*, 2007.

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

[5] W. B. Croft. *Combining Approaches to Information Retrieval*. Springer US, Boston, MA, 2000.

[6] D. Harman. Overview of the first text retrieval conference (trec-1). In *Proc. of TREC*, 1992.

[7] K. Jones and C. Van Rijsbergen. *Report on the Need for and Provision of an Ideal Information Retrieval Test Collection*. 1975.

[8] K. S. Jones. Letter to the editor. *Information Processing & Management*, 39(1), 2003.

[9] B. Koopman and G. Zuccon. Why assessing relevance in medical ir is demanding. In *Medical Information Retrieval Workshop at SIGIR 2014*, 2014.

[10] J. H. Lee. Analyses of multiple evidence combination. In *Proc. of SIGIR*, 1997.

[11] A. Lipani, M. Lupu, and A. Hanbury. Splitting water: Precision and anti-precision to reduce pool bias. In *Proc. of SIGIR*, 2015.

[12] A. Lipani, M. Lupu, and A. Hanbury. The curious incidence of bias corrections in the pool. In *Proc. of ECIR*, 2016.

[13] A. Lipani, M. Lupu, E. Kanoulas, and A. Hanbury. The solitude of relevant documents in the pool. In *Proc. of CIKM*, 2016.

[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*, 2016.

[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*, 2016.

[16] C. Macdonald. *The voting model for people search*. PhD thesis, University of Glasgow, 2009.

[17] A. Moffat, W. Webber, and J. Zobel. Strategic System Comparisons via Targetted Relevance Judgments. In *Proc. of SIGIR*, 2007.

[18] M. Montague and J. A. Aslam. Relevance score normalization for metasearch. In *Proc. of CIKM*, 2001.

[19] M. Montague and J. A. Aslam. Condorcet fusion for improved retrieval. In *Proc. of CIKM*, 2002.

[20] S. Robertson. On the history of evaluation in IR. *Journal of Information Science*, 34(4), 2008.

[21] E. M. Voorhees and D. Harman. Overview of the Eighth Text REtrieval Conference. In *Proc. of TREC*, 2000.

[22] E. M. Voorhees, D. K. Harman, et al. *TREC: Experiment and evaluation in information retrieval*. 2005.

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

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

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