The Wisdom of the Rankers: A Cost-Effective Method for Building Pooled Test Collections without Participant Systems

David Otero
david.otero.freijeiro@udc.es
Information Retrieval Lab
Centro de Investigación en Tecnoloxías da Información e as Comunicacións (CITIC)
Universidade da Coruña
A Coruña, Spain

Javier Parapar
javier.parapar@udc.es
Information Retrieval Lab
Centro de Investigación en Tecnoloxías da Información e as Comunicacións (CITIC)
Universidade da Coruña
A Coruña, Spain

Álvaro Barreiro
barreiro@udc.es
Information Retrieval Lab
Centro de Investigación en Tecnoloxías da Información e as Comunicacións (CITIC)
Universidade da Coruña
A Coruña, Spain

ABSTRACT

Information Retrieval is an area where evaluation is crucial to validate newly proposed models. As the first step in the evaluation of models, researchers carry out offline experiments on specific datasets. While the field started around ad-hoc search, the number of new tasks is continuously growing. These tasks demand the development of new test collections (documents, information needs, and judgments). The construction of those datasets relies on expensive campaigns like TREC. Due to the size of modern collections, obtaining the relevance for each document-topic pair is infeasible. To reduce this cost, organizers usually apply a technique called pooling. When building pooled test collections, assessors only judge a portion of the documents selected among the participants' results. Although the judgments will not be exhaustive, they will be sufficiently complete and unbiased if pooling is done correctly. Therefore, researchers may safely use pooled collections to evaluate new models. However, the application of pooling depends on the existence of participant systems. This need is a handicap for tasks for which it is necessary to release training data before the celebration of the competition or for those with few participants. In this paper, we present a simple method for building pooled collections when such restrictions exist. Our proposal relies on two principles: the wisdom of the rankers and the application of pooling. By creating enough artificial participant systems, we can apply pooling on their results to select the documents that merit human assessment. Using an innovative approach to evaluate our method, we show that researchers may use it to produce high-quality collections on the absence of participant systems.

CCS CONCEPTS

• **Information systems** → **Test collections**.

KEYWORDS

Information Retrieval, Test Collections, Pooling

ACM Reference Format:
David Otero, Javier Parapar, and Álvaro Barreiro. 2021. The Wisdom of the Rankers: A Cost-Effective Method for Building Pooled Test Collections without Participant Systems. In *The 36th ACM/SIGAPP Symposium on Applied Computing (SAC '21), March 22–26, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3412841.3441947

1 INTRODUCTION

Recently, the number of emerging tasks related to Information Retrieval (IR) has exploded. These tasks are very diverse, not exclusively about document ranking [22, 27]. They appear as new commercial needs (e.g., conversational assistants or recommender systems) or research community efforts. IR highly relies on strict evaluation procedures for assessing the effectiveness of new algorithms for these new tasks. Typically, a system is evaluated by measuring its effectiveness to retrieve relevant documents. This evaluation is made by using one or more test collections. Consequently, the proliferation of tasks demands the building of new benchmarks. The research community constructs these datasets in collaborative evaluation forums like TREC [41], NTCIR [31] or CLEF [14]. A test collection comprises a set of documents, information needs, and relevance judgments that capture the relevance relation between documents and information needs. Obtaining this ground truth is expensive because it requires human work. Although the ideal situation would be to have the complete set of relevance judgments (i.e., the judgment for every query-document pair), this is infeasible due to modern collections' size. In traditional evaluation workshops like TREC, *pooling* is used to reduce the cost of producing these assessments. In pooled datasets, the assessors only judge a fraction of the whole document corpus. This subset of documents is obtained from the results sent by the competition participants. Although pooling can alleviate the cost of obtaining new relevance judgments, organizing an evaluation task is difficult and expensive.

As pointed before, this sort of evaluation campaign relies on the existence of participant systems. This creates a problem for research groups or other organizations that may not have enough resources to build a new test collection. Also, with the traditional approach used in TREC campaigns, it is impossible to obtain pooled training data before the competition holds, as the final collection is the result of the campaign. Previous studies have tried to develop new methods to overcome these issues. In particular, Sanderson et al. proposed a dynamic method to build a set of reusable relevance assessments when there are not enough resources available [35]. They used relevance feedback combined with a single retrieval system to obtain the assessments while limiting the number of judgments. However, this approach has some caveats that we explain in the following section.

This paper aims to create a static method saving the logistic problems of dynamic ones in real environments. We propose a new method to build pooled test collections when organizing an expensive evaluation campaign is not an option. This work aims not to investigate or develop new pooling strategies but to define a simple (and yet effective) method to create pooled collections in a situation where gathering participant results is not possible. Our proposal is based on two main ideas. The first one is to generate enough simulated participant submissions by using query variants and well-known retrieval models. The second one is the use of intelligent pooling strategies that drastically reduce the number of judgments needed without harming the obtained datasets' quality. We apply pooling over systems instead of working with one single model, as one of the premises of pooling is that it has to be applied over a large enough variety of systems. Moffat et al. [30] demonstrated that user query formulations produce as diverse results as different ranking models. Therefore, we also use query variations to increase the number of simulated systems and the pooled documents' representativeness. Using this approach, we conduct experiments on various datasets, showing that it is possible to get reusable and fair collections when there are not enough resources to organize a traditional evaluation campaign.

2 RELATED WORK

IR is a field deeply rooted in strict evaluation procedures with test collections. The Cranfield paradigm is the standard evaluation method for assessing the effectiveness of new developments. Three components form a test collection: i) a document set, ii) a set of information needs, and iii) the relevance judgments. Because it is easy to apply and highly reproducible, it has become the *de facto* method to evaluate new developments in IR: the use of test collections allows the researchers to compare different retrieval methods effectively. However, building fair and reusable test collections is a challenging task. The two first components of a dataset (the documents and the information needs) are often not difficult to obtain. The third component, the *ground truth* (relevance judgments), is often created by humans and, thus, is a bottleneck when building a test collection. Assuming a judgment time of 30 seconds per document, it would take almost a year to judge the whole set for one topic for a dataset of 800,000 documents. This approach's problem is that the volume of information in modern test collections is too large to have complete judgments. Since the first use of test collections for IR evaluation, researchers have devoted much work to reduce the cost of building new testbeds. In this section, we will discuss some of these methods.

Pooling. The first and most applied technique to reduce the assessment effort is *pooling*. Spärck Jones and van Rijsbergen introduced this method in [37]. In pooled test collections, assessors only judge a subset of the entire document set, usually the top k results sent by the participant teams form the pool. This technique is useful because many relevant documents will appear near the top of the ranks obtained by each of the groups. Thus, these documents will enter the pool, and their relevance will be judged. Some relevant documents might not enter the pool. However, this will not harm the evaluation because the relative comparison among methods should be fair. For achieving that goal, the pooling method depends on the fact that enough diverse systems must participate in the pool and that the pool depth should be sufficient to obtain documents that, once assessed, should serve to evaluate new retrieval models effectively. If done correctly, researchers may assume the completeness of the obtained relevance judgments. If not, the collection may be biased and will not be appropriate for reuse to evaluate other new systems.

Subset pooling. Given a query and various document ranks, judging deep pools is usually infeasible. Due to this problem, many works have proposed methods for assessing only a subset of the pool. These pooling strategies' objective is to explore a subset of the pool, looking for relevant documents, and discard the pool's rest. These strategies are known to be effective and lead to high-quality datasets [11].

A pooling strategy is a method that takes a set of document ranks and the pool depth as inputs and returns a sequence of documents to judge. Historically, in TREC competitions, the pool's documents are judged following an arbitrary order, i.e., by DocID. The probability of bias in human assessments is reduced with this arbitrary order of presentation of documents. Although this is the traditional method in TREC workshops, many works tried to develop new pooling algorithms that impose alternative orders in the evaluation to reduce the assessment effort without affecting the quality of the obtained judgments. In particular, in TREC Common Core Track 2017 [1], NIST used for the first time a pooling strategy based on Bayesian Bandits [23, 24].

There are two main types of pooling strategies: static and dynamic. Static strategies, like DocID, establish an order of the pool before any judgment is done, and that order never changes. On the other hand, the order that the dynamic strategies impose changes as documents are judged. These strategies use the assessments to establish an order of the set of unjudged documents in the pool. When we use pooling strategies, we can adjust the assessment cost by imposing a constraint that limits the number of judgments to obtain. This is fixed-cost pooling [20]. Along with using one of the aforementioned sorting strategies, the pooling process can stop earlier and still judge many relevant documents. Some works have demonstrated that it is feasible to build new benchmarks with an acceptable quality judging only a subset of the pool [5, 9, 25].

Shallow pooling. Traditionally, in TREC-style pooling, the top 100 (or 50) documents returned by each run are assessed for relevance. Another possible approach to reduce the assessment effort is to reduce the depth at which the runs are examined: shallow pooling [43]. Picking a smaller number of documents from each run results in smaller pools, therefore judging fewer documents. Nevertheless, there is a high risk that this may harm the final collection's quality if the depth is too low with respect to the collection size [8].

Pseudo-relevance judgments. Soboroff et al. [36] studied a method of obtaining relevance assessments in the absence of manual judgments (a more extensive study can be found in [34]). In this work, they conclude that using pseudo-relevance judgments positively correlates with the official TREC judgments. However, they also say that "such a methodology would not be useful in an environment such as TREC, where there is already a commitment to conducting relevance assessments and building test collections in the Cranfield tradition."

Avoid system pooling. In [35], Sanderson et al. studied if it is possible to build a reusable set of judgments by working only with a single retrieval system in combination with manual feedback. Although they obtain competitive results in terms of reusability, the fact that they rely on a single system may cause the collection to be biased to some system that shares the same fundamental ideas as the method used to build the dataset. The diversity of the retrieval systems used is one critical condition that guarantees the quality of a TREC collection [8]. Moreover, the interactive search and judging approach that they use is not very suitable for the classical multiple-assessors evaluation approach [40] as it is a method that needs the judgments of one iteration to be available before the next iteration goes on.

Other works. Other methods seek to reduce the assessment effort using alternative approaches. One of these approaches uses crowdsourcing techniques to make judgments. Crowdsourcing delegates the assessment task to a large external group of people instead of having their employee group doing it. Reported results showed that it is feasible to do this and, additionally, these methods provide a better way of scaling up the assessment process [2].

3 SIMULATING PARTICIPANT SYSTEMS

Our main focus is to investigate if we may create new pooled test collections in a situation where there are not enough resources to organize a traditional evaluation campaign. As noted before, much work was done to reduce the cost of building new IR evaluation benchmarks. However, none of these techniques allows us to create pooled collections in a situation where having participant systems is not an option.

This section presents the three main components that we employ to build the collections: query variants, well-known retrieval models, and smart pooling strategies. These components are available to any team implied in constructing a test collection with limited resources.

We use the first two components to simulate the document ranks that participant groups would provide in a real evaluation campaign. To generate these ranks, we employ different well-known retrieval models such as BM25 or Language Models. The effectiveness of these models was widely shown. Ultimately, many methods used by the real participants of the competitions are variants of well-known models. Previous work has also shown that the system variability is as substantial as the user variability [6, 29, 30]. Thus we intend to consider both aspects by combining multiple retrieval systems with different query variants for each topic. The details of how it is done are explained later in the paper. Then, we use three different pooling strategies –two are static pooling methods, and the last is a dynamic pooling strategy– to select the documents that merit the human judgments.

Table 1: Notation summary.

Symbols and funcs.	Description
q	A query.
$\mathcal{R}_q$	Set of pooled runs for a query.
r	A run $r \in \mathcal{R}_q$.
d	A document.
$\mathcal{D}_{r,k}$	Top k documents retrieved by r.
$d_r@k$	Document d at position k of run r.

The whole method to create a new collection is as follows: first, we generate the simulated document ranks by combining the mentioned retrieval models with a set of query variants for each topic. These queries will serve as the inputs to the retrieval models. Then we use those ranks as the inputs to the pooling strategies to select the documents that will be candidates for assessment. Once these documents are selected, they are judged for building the final collection. We use the actual TREC assessments to judge the selected documents.

Query Variants. The queries that we employ as inputs to the retrieval models are obtained in two different ways: manually and automatically.

Manual query variants. The manual query variants that we use correspond with the ones constructed on the topics of the ROBUST TREC Collection by the RMIT team [7][1]. These queries include the topics 301-450 and 600-700.

Automatic query variants. Alternatively, we propose to obtain different queries from the same topic automatically. For doing so, we obtain query variants by adding to the short (*title*) query a term from the top IDF sorted terms from the *narrative* and *description.* We create N different variants by expanding the short query with one of the terms in the top-N ranked terms. Although we could employ better term selections schemes, we decided to use this approach to check how a simple and unbiased method works.

Ranking Methods. To simulate ranks that will supply enough relevant documents to assess, we have chosen well-known retrieval models whose effectiveness has been widely evaluated in the literature. In particular, we use BM25 [33], query likelihood [32] with Jelinek-Mercer [16] and Dirichlet [26] smoothing, Divergence From Randomness [3], Vector Space Model with TF-IDF weighting, the ranking model based on the vector space model and boolean queries provided by Apache Lucene, Divergence From Independence [18], Information-based model [10], and Axiomatic models [15]. We employ the 72 models implemented in Apache Lucene[2]. We set the parameters to the recommended values[3].

Pooling. The pooling process to obtain the relevance judgments of a new collection is divided into two parts. First, we have to choose the set of documents that form the pool. Second, we have to select the pooling strategy that establishes the order in which the assessors

[1]http://culpepper.io/publications/robust-uqv.txt.gz
[2]https://lucene.apache.org
[3]See Apache Lucene documentation for more info

judge the documents. The choice of any of these aspects has a direct impact on the final collection quality. We explore different pooling strategies and various methods to obtain the pool to evaluate which combination yields better results.

Adjudicating methods. In this work, we have analyzed three different strategies: two of them are *static* adjudicating methods, and one is a *dynamic* adjudicating method:

- **DocID** [41]: classic unbiased full pool assessment strategy. Documents are sorted by their identifier. It is the one employed typically in TREC workshops.
- **MoveToFront (MTF)** [12]: dynamic pooling strategy proposed by Cormack et. al. It is based on the idea that ranks are a good approximation of the ideal relevance order. Thus, a recent rank that has yielded a relevant document has more probabilities to yield another relevant document in the future. A more extensive and formal definition can be found in [20].
- **DocPoolFreq**: since we explore the idea of using the wisdom of the rankers to obtain relevance judgments by simulating enough participant systems, we also study the use of this idea to develop a new pooling strategy. The notation employed hereinafter is explained in Table 1.
  Given a rank r of documents for a query, and a document d, we define a function $\#(d, \mathcal{D}_{r,k})$ that returns 1 if the document belongs to the first k positions of the rank ($\mathcal{D}_{r,k}$), or 0 otherwise:
  $$\#(d, \mathcal{D}_{r,k}) = \begin{cases} 1 & \text{if } d \in \mathcal{D}_{r,k} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$
  Next, given a set of ranks for the query provided by the systems in the pool, $\mathcal{R}_q$, we define a scoring function that counts the times that each document appears in the top-k positions for the query:
  $$DocPoolFreq(d, \mathcal{R}_q) = \sum_{r \in \mathcal{R}_q} \#(d, \mathcal{D}_{r,k}) \quad (2)$$
  We compute Eq.2 for each document in the set $\mathcal{R}_q$. Then, the documents are sorted by decreasing order of this value. If there is a tie, we use their identifier to sort them in alphabetical order. Since it is a static adjudicating strategy, we compute this order before the assessment process, and it never changes. Note that the pool depth's value of k and the set of pooled documents depends on the strategy used to construct the pool. We discuss the different methods that we use in this work in the following section. This simple pooling strategy's main idea is that if a document is retrieved by a larger number of systems than other documents, it may have a higher probability of being relevant. So, instead of just sorting the pooled documents by an arbitrary order like DocID, we aggregate the knowledge of the submissions with the hope of improving the pooling process by reducing the assessment effort. Moreover, it is a static adjudicating strategy. Thus, we avoid the logistic problems that a dynamic method may suffer in a real environment.

Pool construction. In addition to these three strategies, we evaluate three different methods of building the document pool.

- **Full pool (pool depth = 100)**: we build the pool following the typical top-100 method used in TREC, i.e., the union of the first 100 documents retrieved by each submission.
- **Limited budget and pool with variable depth**: we explore how further to exploit the order information in the systems' ranking. When doing fixed-cost pooling, we have two possibilities to limit the number of judgments. Traditional approaches first build the set of documents that conform to the pool and then limit the number of decisions to a subset of those documents. Alternatively, we propose to restrict the judged documents by adequately selecting them and limiting the pool's size to the budget that we have. We select those documents by taking advantage of its original position in the systems, instead of blindly choosing documents from the pooled set by ignoring its original ranking. We pick the first document of every rank, then the second document of every rank, and so on. The process stops when n different documents are selected, being n the fixed-cost budget. In this way, we keep the budget while exploiting the fact that the systems' highly ranked documents are more likely to be relevant. How we select the set of variable pooled documents for a query q is defined formally in Alg. 1, where $d_r@k$ is the document at position k in rank r for the query. We say that the depth is variable because it is not assured that we sample every system the same number of times.

Algorithm 1: Variable depth pool construction for a query

```
Input: $\mathcal{R}_q$, set of ranks for a query.
       $n$, judgments budget.
Output: $\mathcal{P}$, set of documents to be judged.
$\mathcal{P} \longleftarrow \emptyset$
$k \longleftarrow 1$
while $|\mathcal{P}| \leq n$ do
    $\mathcal{P}_k \longleftarrow \emptyset$
    for $r \leftarrow 1$ to $min(n - |\mathcal{P}|, |\mathcal{R}_q|)$ do
        $\mathcal{P}_k \longleftarrow \mathcal{P}_k \cup \{d_r@k\}$
    end
    $\mathcal{P} \longleftarrow \mathcal{P} \cup \mathcal{P}_k$
    $k \longleftarrow k + 1$
end
```

- **Full pool but limited budget (pool depth = 100)**: as we said in a previous section, the pooling ordering strategy would not affect the final collection if we assess the entire pool. To evaluate how the pooling strategies behave when judging only a subset of the pool, we have followed this method: we build the pool, but we constrain the number of judgments to a limit. In this work, we have established a budget of 500 judgments.

4 EXPERIMENTS

This section presents the different experiments that we have done and how we combine the components exposed in the previous section. Finally, we also show the datasets that we have used.

Experiments design. First, we explain how we generate the document ranks to play the participant systems results' role. Last, we show how, using those ranks, we perform a pooling process to obtain the final documents that the assessors will judge.

Runs simulation. The first step is to generate sets of runs that play the role of the participant teams' results at the competition. A run is formed by the ranks produced by a system from a participant team for all the queries. We generate each run by using a set of queries and a retrieval model. We tested four experimental settings using four different query sets. The first setting (*title*) is composed of the short queries: we extract the topic title and use it as a query against the 72 retrieval models. We obtain the remaining experimental settings by adding another set of runs to this base set. All of them are obtained following the same procedure: we select a different set of query variants and use them as queries against the 72 retrieval models. To construct the second one, we add 72 runs (144 in total) produced by using *title + description* queries. To build the third one (*title + manual*), we use eight manually generated query variants per topic, resulting in $8 \times 72 = 572$ new runs (8 is the minimum number of manual variants that every topic has). Finally, we construct the fourth set (*title + automatic*) in the same way that the third, but instead of using the manual produced queries, they are automatically generated. In this case, we also create 8 query variants, the same as with the manuals, to fairly analyze the performance w.r.t to the manual variants. These queries are generated by expanding the short query with the eight top-ranked terms by IDF among the terms in *description* and *narrative* sections of the topic.

Pooling. We then use each of these four sets as the inputs to the three adjudicating –pooling– methods. Additionally, we have employed various strategies for building the pool of documents that are candidates to be judged. From the different combinations of the elements that form the method, we have made various sets of relevance assessments (*qrels*) that will then be used to evaluate our method.

Evaluation. While test collections are an essential piece for the research on Information Retrieval, building fair and reusable testbeds is a challenging task. This section presents our experiments to evaluate these two aspects of the datasets produced by our new proposed method. In this context, the collection's fairness is how it evaluates runs that contributed to the pool. On the other hand, reusability is that this evaluation is also fair to the systems that did not participate in the building process [43]. We have designed two new evaluation methodologies to assess both the reusability and the fairness of our method.

Typically, researchers evaluate the reusability and fairness of a collection using experiments like "leave-one-run-out" and "leave-one-group-out" [40, 43]. The first method compares the position of a system when evaluated with the official qrels w.r.t. using an alternative qrels set where documents uniquely retrieved by that system are removed. The second method is a similar process. The difference is that the relevant documents uniquely retrieved by a participant team's systems are excluded. These approaches are a way of examining how fairly the collection evaluates runs that did not contribute to the pool. However, these approaches would yield very high correlation values but for systems that provide many unique documents to the pool.

Reusability. We evaluate the reusability of the judgments with a more extreme scenario than traditional experiments. First, we compute a ranking of TREC participant systems using the official TREC judgments and, second, a different ranking using those produced with our method. Unlike existing reusability studies, none of the TREC systems were used to build the pool obtained with the evaluated methods. Therefore we are evaluating how the judgments built with our method evaluate a completely different set of runs. We use MAP as the evaluation measure. For computing the correlations, we use Kendall's (τ) correlation [17] and τ_{ap} correlation [42], which is more affected by swaps in the top positions of the rank. Typically, a correlation value over 0.9 is assumed to be acceptable [39].

Fairness. Traditional evaluation of fairness in our method, that is, assessing if some participant systems unfairly benefit from constructed judgments, is somehow irrelevant since we are not using real participant runs in the process. However, we propose evaluating the fairness of our method, that is, how it evaluates each of the original TREC systems with a different method from traditional approaches. To evaluate this aspect, we have developed a new evaluation procedure that we have denoted as *include-one-in* (IOI). We conduct this procedure in the following manner: we start from the judgments obtained with the simulated systems. Then, we build a new set of assessments by adding to the simulated runs one system from the official TREC participants. We generate a different set of qrels from each one of the official runs. Then, we calculate the position changes in the system's ranking that each official system suffers when ranking with and without it participating in the pool.

Datasets. We conducted our experiments on four standard collections from the well known Text REtrieval Conference (TREC). Three of these datasets were built in the ad hoc retrieval task: TREC 6, TREC 7, and TREC 8. The last one used was built in the ROBUST track: ROBUST 2004. As our manual query variants are restricted to a set of topics, we can only use collections that have relevance assessments on those topics to evaluate all of our methods. These topics are 301-450 and 600-700.

5 RESULTS & DISCUSSION

Tables 2 and 3 show the τ and τ_{ap} values computed to evaluate the reusability. Although we have computed these values for each of the datasets, we only present here the results for the ROBUST 2004 and TREC6 collections due to space restrictions. However, the results in the rest of the datasets follow the same trends. Table 2 shows the results when building the pool with the top-100 documents and constraining the number of judgments to 500. On the other hand, Table 3, depicts the results for the other strategies of building the document pool. In this second table, we have omitted the different adjudicating methods because they judge the same set of documents, thus yielding equal correlations.

Choosing among adjudicating methods. To select the best pooling strategy, we compare each technique in a situation where the number of relevance judgments is constrained. These are the results presented in Table 2. In light of these numbers, we conclude that it is possible to obtain a reusable collection, even when imposing a budget in the number of judgments. By using an intelligent pooling strategy over the simulated runs simulated with manual variants we can obtain strong correlations.

Another common way to compare different pooling strategies is to plot the number of documents judged against the number of positive assessments done. An optimal method would choose all relevant documents first. The rest of the documents, the non-relevant, would be left to the end of the process. A graphic like this is shown in Fig. 1. In this figure, we plot the recall curve of the three pooling strategies when doing a full pool assessment with a pool depth of 100. This result shows that DocPoolFreq and MTF are more effective pooling strategies than DocID because they could be used to extract the judgments and to stop earlier, thus reducing the cost.

At this point, we can conclude that DocPoolFreq performance is as good as MTF, with the difference that DocPoolFreq is a static adjudicating method. Thus, DocPoolFreq is more suitable to use in a real environment as it avoids the logistic problems of dynamic methods.

Besides, Table 3 shows that our strategy to build the pool with a variable depth yields to high correlations when doing fixed-cost pooling with 500 judged documents. Therefore, our pool construction proposal seems suitable to build new judgments under a reduced budget even when employing an arbitrary adjudicating method like DocID.

Manual or automatic variants. Obtaining manual variants is very expensive compared to the cost of obtaining automatic variants. For this reason, we study if we may be able to reach equivalent quality as the ones obtained with manual variants by using automatically generated ones. To evaluate this, we have generated a series of run sets considering different numbers of terms to expand the title query. These terms are selected from the top ranked by IDF from the topics' *narrative* and description. We pooled over those sets of runs, assuming a budget of 500 judgments under full pool for each topic. Finally, we computed the correlation in the system's order between our qrels and the official ones. The results obtained from this evaluation are illustrated in Figure 2. The x-axis represents the number of queries generated from the same topic. As we explained in a previous section, we expand the title of the topic with the terms from the *description* and *narrative* sections that have the highest IDF. The variants that we can generate for the same topic are only limited by the different terms that are present in those two sections. More complex models for generating query variants could be explored in the future [21]. The results show that, although we are not able to reach the correlation values obtained with the manual variants, we still reach a very acceptable quality (τ > 0.9). Moreover, the performance when varying the number of variants per topic remains stable (after 5) when employing intelligent adjudicating methods.

Fairness. Finally, the results obtained from the fairness evaluation are shown in Figure 3. This figure shows the rankings' position changes, in a boxplot graph, which were obtained for each run set on ROBUST 2004 and TREC 6. The evaluation was carried out under full pool (depth 100) and judging every document from the pool. In the x-axis are represented the different strategies to simulate the participant runs. We computed each run's drop under the include-one-in methodology explained in Section 4. These results continue to convey the idea of the previous section: using the manually generated queries provides a collection of better quality that is more reusable and fairer. Additionally, using automatically generated query variants provides acceptable results, although they are not as good as the former. Whether or not a better query generation may bring the quality closer to that achieved with the manual variants requires further investigation.

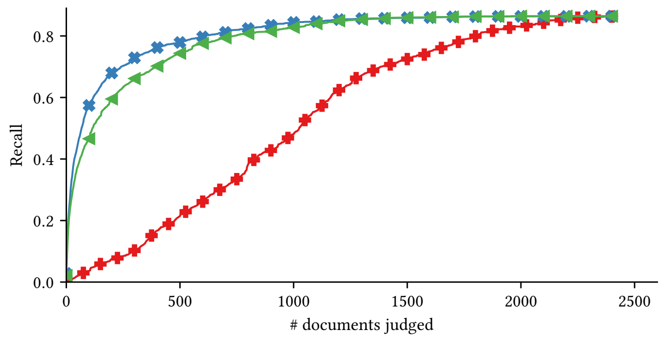
In terms if building reusable collections when no participant systems are available, the results presented here are similar to those presented in [35]. However, we have developed a static method to select the documents that merit the human judgments, thus avoiding the burden that supposes a dynamic method in a real scenario. Additionally, by pooling over query variations and system variations we make the method more robust to the evaluation of new models that have not been used to build the collection.

6 CONCLUSIONS AND FUTURE WORK

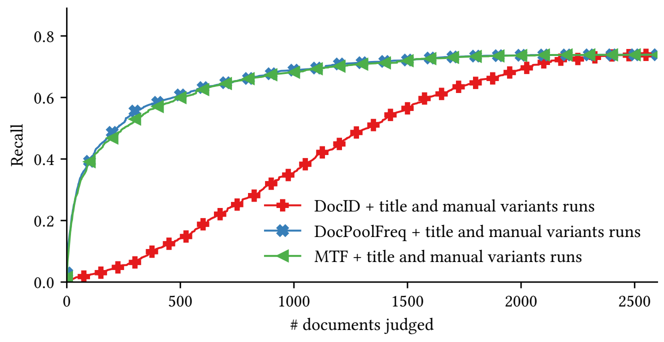
We have proposed a new method to ease the building of evaluation datasets when no participant systems are available. By pooling over systems and query variants, we are able to obtain high-quality judgments in a static and cheap manner. Although in our experiments we have employed TREC relevance assessments, we have defined a new methodology that in a real scenario, with the use of simulated participant systems and intelligent pooling strategies, would foster the creation of new judgments in a easy and cheap way. We found that with simulated systems using out-of-the-box retrieval models, and a suitable adjudicating method, we can obtain strong correlations when only judging 500 documents. Thus, we are significantly reducing the cost of constructing a new testbed. On the one hand, the best strategy to generate the document ranks is to use manual query variants. Our results open the development of new tools that will help researchers automate the process of building their own benchmarks when resources are scarce. On the other hand, we presented an adjudicating approach specially tailored for taking advantage of the wisdom of the rankers to obtain the most of the relevant documents earlier in the process. DocPoolFreq performs the best, the higher the number of systems in the pool. This fulfils our initial aim of developing a new method to build new collections when only few resources are available while keeping it simple but yet effective. Since it is a static strategy, it does not have the logistic problems of a dynamic method that can be a burden in a real environment. Additionally, we also demonstrated that, even when a simple strategy like DocID is used, high-quality collections can be obtained by adequately selecting the documents in the pool.

In summary, although we have employed TREC assessments, in practice this method could be used in a situation where no previous data is available and researchers may need to release training data to the participants, since it is a simple and static method that does no have the burden of a dynamic method in a real environment. In addition, our approach could also be used in combination with crowdsourcing techniques to gather more judgments when no real participant systems are available.

This paper paves the way for plenty of future work. First, we envision extending this work to more types of retrieval models. Some retrieval models may have a significant impact on final collection quality by providing more relevant documents to the pool, such

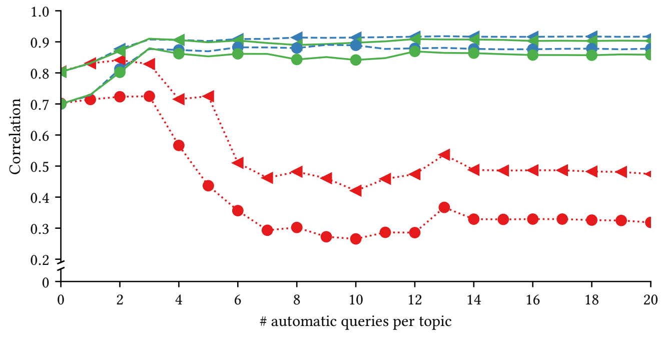


a: ROBUST 2004.

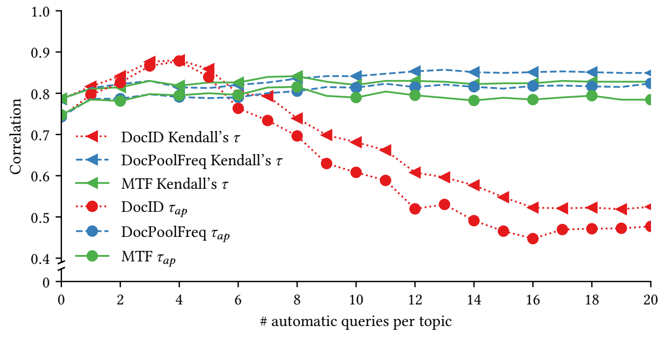


b: TREC 6.

Figure 1: Evolution of the number of relevant documents found in the pooling process using the runs produced with manual queries in ROBUST 2004 and TREC6 datasets. The number of documents judged is averaged over all queries. The rest of the possible combinations are omitted for the sake of clarity.

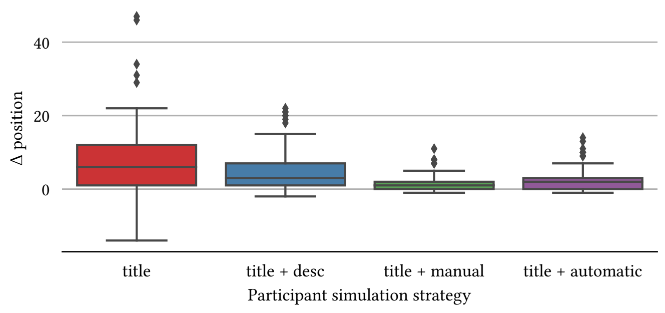


a: ROBUST 2004.

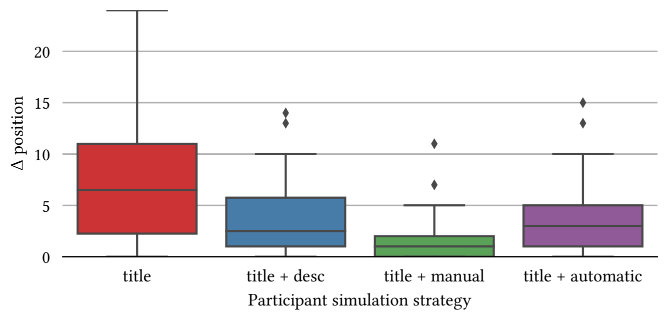


b: TREC 6.

Figure 2: Values of Kendall's τ and τ_{ap} correlations among the rankings of TREC participants obtained with the official qrels and qrels built with our method on the ROBUST 2004 and TREC6 datasets. The x-axis represents the number of different automatically generated query variants from each topic. A value of 0 means that only the topic title was used. These corrrleations where obtained when taking the top 100 documents of each run and constraining the number of judgments to 500.



a: ROBUST 2004.



b: TREC 6.

Figure 3: Rank position changes when ranking the official TREC runs with qrels built from the simulated runs and the qrels built from the simulated runs plus one official TREC run. Both sets of qrels were built from a top-100 full pool judging procedure.

Table 2: Kendall's τ and τ_{ap} correlation values among the rankings of TREC participants obtained with the official qrels and the qrels obtained with the pooling strategies and simulated runs and the pooling strategies and official runs (last row). All results were obtained when constraining the number of judgments to 500. Best results are highlighted in bold (excluding the results obtained with the official runs).

Collection	Run set	Kendall's τ			τ_{ap}		
		DocID	DocPoolFreq	MTF	DocID	DocPoolFreq	MTF
ROBUST 2004	Title	0.8048	0.8031	0.8031	0.7020	0.7000	0.7000
	Title + description	0.8675	0.8869	0.8859	0.8127	0.8588	0.8553
	Title + manual queries	0.4594	**0.9499**	0.9359	0.2889	**0.9238**	0.9032
	Title + automatic queries	0.4817	0.9139	0.8899	0.3022	0.8800	0.8432
	Official runs operating with the pooling strategies	0.6422	0.9927	0.9903	0.4737	0.9885	0.9856
TREC 6	Title	0.7855	0.7855	0.7874	0.7431	0.7431	0.7475
	Title + description	0.8319	0.8261	0.8184	0.7839	0.7759	0.7757
	Title + manual queries	0.5981	0.8821	**0.9034**	0.5338	0.8496	**0.8739**
	Title + automatic queries	0.7391	0.8357	0.8415	0.6965	0.8050	0.8155
	Official runs operating with the pooling strategies	0.6850	0.9633	0.9768	0.6421	0.9544	0.9684

Table 3: Kendall's τ and τ_{ap} correlation values among the rankings of TREC participants obtained with the official qrels and the qrels obtained with the pooling strategies and simulated runs. Two strategies were used to build the pool of documents: full pool and variable depth. Best results are highlighted in bold (excluding the results obtained with the official runs).

Pool	Run set	ROBUST 2004		TREC 6	
		Kendall's τ	τ_{ap}	Kendall's τ	τ_{ap}
Full pool	Title	0.8031	0.7000	0.7874	0.7475
	Title + description	0.8929	0.8663	0.8473	0.8140
	Title + manual queries	**0.9676**	**0.9556**	**0.9362**	**0.9241**
	Title + automatic queries	0.9419	0.9176	0.8551	0.8265
Variable depth (500)	Title	0.8031	0.7000	0.7874	0.7475
	Title + description	0.8852	0.8536	0.8222	0.7822
	Title + manual queries	**0.9239**	**0.8870**	**0.8802**	**0.8475**
	Title + automatic queries	0.8845	0.8356	0.8357	0.8117
	Official runs operating with the pooling strategies	0.9813	0.9709	0.9614	0.9442

as pseudo relevance feedback models such as Relevance Models [19] or LiMe [38], or Neural Models [13, 28]. Second, it would be interesting to evaluate other pooling strategies such as Hedge [4] or Bayesian Bandits [23, 24]. We have not tested them here because we aimed to develop a simple and easy to implement method that other researchers may use. Additionally, our approach may be combined with other strategies to reduce the assessment effort, such as shallow pooling. This will allow other researches to build new benchmarks at an even further reduced effort. Finally, this work can also be extended to other tasks beyond information retrieval, e.g., classification, where positive cases are labelled with pooling over the results of many classifiers.

ACKNOWLEDGMENTS

This work was supported by projects RTI2018-093336-B-C22 (MCIU & ERDF), and GPC ED431B 2019/03 (Xunta de Galicia & ERDF), and accreditation ED431G 2019/01 (Xunta de Galicia & ERDF).

REFERENCES

[1] James Allan, Donna K. Harman, Evangelos Kanoulas, Dan Li, Christophe Van Gysel, and Ellen M. Voorhees. 2017. TREC 2017 Common Core Track Overview. In *Proceedings of TREC 2017*. NIST. https://trec.nist.gov/pubs/trec26/papers/Overview-CC.pdf

[2] Omar Alonso and Stefano Mizzaro. 2012. Using Crowdsourcing for TREC Relevance Assessment. *Information Processing & Management* 48, 6 (Nov. 2012), 1053–1066. https://doi.org/10.1016/j.ipm.2012.01.004

[3] Gianni Amati and Cornelis J. van Rijsbergen. 2002. Probabilistic Models of Information Retrieval Based on Measuring the Divergence from Randomness. *ACM Transactions on Information Systems* 20, 4 (Oct. 2002), 357–389. https://doi.org/10.1145/582415.582416

[4] Javed A. Aslam, Virgil Pavlu, and Robert Savell. 2003. A Unified Model for Metasearch, Pooling, and System Evaluation. In *Proceedings of ACM CIKM 2003*. ACM, New York, NY, USA, 484–491. https://doi.org/10.1145/956863.956953

[5] Javed A. Aslam, Virgil Pavlu, and Emine Yilmaz. 2006. A Statistical Method for System Evaluation Using Incomplete Judgments. In *Proceedings of ACM SIGIR 2006*. ACM, New York, NY, USA, 541–548. https://doi.org/10.1145/1148170.1148263

[6] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2016. UQV100: A Test Collection with Query Variability. In *Proceedings of ACM SIGIR 2016*. ACM, New York, NY, USA, 725–728. https://doi.org/10.1145/2911451.2914671

[7] Rodger Benham, Luke Gallagher, Joel Mackenzie, Tadele T. Damessie, Ruey-Cheng Chen, Falk Scholer, and J. Shane Culpepper. 2017. RMIT at the TREC CORE Track. In *Proceedings of TREC 2017*. NIST. https://trec.nist.gov/pubs/trec26/papers/RMIT-CC.pdf

[8] Chris Buckley, Darrin Dimmick, Ian Soboroff, and Ellen M. Voorhees. 2007. Bias and the limits of pooling for large collections. *Information Retrieval Journal* 10, 6 (Aug. 2007), 491–508. https://doi.org/10.1007/s10791-007-9032-x

[9] Ben Carterette, James Allan, and Ramesh Sitaraman. 2006. Minimal Test Collections for Retrieval Evaluation. In *Proceedings of ACM SIGIR 2006*. ACM, New York, NY, USA, 268–275. https://doi.org/10.1145/1148170.1148219

[10] Stéphane Clinchant and Eric Gaussier. 2010. Information-Based Models for Ad Hoc IR. In *Proceedings of ACM SIGIR 2010*. ACM, New York, NY, USA, 234–241. https://doi.org/10.1145/1835449.1835490

[11] Gordon V. Cormack and Thomas R. Lynam. 2007. Power and Bias of Subset Pooling Strategies. In *Proceedings of ACM SIGIR 2007*. ACM, New York, NY, USA, 837–838. https://doi.org/10.1145/1277741.1277934

[12] Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. 1998. Efficient Construction of Large Test Collections. In *Proceedings of ACM SIGIR 1998*. ACM, New York, NY, USA, 282–289. https://doi.org/10.1145/290941.291009

[13] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *Proceedings of ACM SIGIR 2019*. ACM, New York, NY, USA, 985–988. https://doi.org/10.1145/3331184.3331303

[14] Experimental IR Meets Multilinguality, Multimodality, and Interaction - 10th International Conference of the CLEF Association 2019. *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 10th International Conference of the CLEF Association* (Lugano, Switzerland). Springer International Publishing, Cham.

[15] Hui Fang and ChengXiang Zhai. 2005. An Exploration of Axiomatic Approaches to Information Retrieval. In *Proceedings of ACM SIGIR 2005*. ACM, New York, NY, USA, 480–487. https://doi.org/10.1145/1076034.1076116

[16] Fred. Jelinek and Robert L. Mercer. 1980. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*. North Holland, Amsterdam, 381–397.

[17] Maurice G. Kendall. 1938. A new measure of rank correlation. *Biometrika* 30, 1/2 (1938), 81–93. https://doi.org/10.2307/2332226

[18] İlker Kocabaş, Bekir Taner Dinçer, and Bahar Karaoğlan. 2014. A Nonparametric Term Weighting Method for Information Retrieval Based on Measuring the Divergence from Independence. *Information Retrieval Journal* 17, 2 (April 2014), 153–176. https://doi.org/10.1007/s10791-013-9225-4

[19] Victor Lavrenko and W. Bruce Croft. 2001. Relevance Based Language Models. In *Proceedings of ACM SIGIR 2001*. ACM, New York, NY, USA, 120–127. https://doi.org/10.1145/383952.383972

[20] Aldo Lipani, David E. Losada, Guido Zuccon, and Mihai Lupu. 2019. Fixed-Cost Pooling Strategies. *IEEE Transactions on Knowledge & Data Engineering* (2019). https://doi.org/10.1109/TKDE.2019.2947049

[21] Binsheng Liu, Nick Craswell, Xiaolu Lu, Oren Kurland, and J. Shane Culpepper. 2019. A Comparative Analysis of Human and Automatic Query Variants. In *Proceedings ACM ICTIR 2019*. ACM, New York, NY, USA, 47–50. https://doi.org/10.1145/3341981.3344223

[22] David E. Losada, Fabio Crestani, and Javier Parapar. 2018. Overview of eRisk: Early Risk Prediction on the Internet. In *CLEF 2018*. Springer International Publishing, Cham, 343–361. https://doi.org/10.1007/978-3-319-98932-7_30

[23] David E. Losada, Javier Parapar, and Álvaro Barreiro. 2016. Feeling Lucky?: Multi-armed Bandits for Ordering Judgements in Pooling-based Evaluation. In *Proceedings of ACM SAC 2016*. ACM, New York, NY, USA, 1027–1034. https://doi.org/10.1145/2851613.2851692

[24] David E. Losada, Javier Parapar, and Álvaro Barreiro. 2017. Multi-armed bandits for adjudicating documents in pooling-based evaluation of information retrieval systems. *Information Processing & Management* 53, 5 (Sept. 2017), 1005–1025. https://doi.org/10.1016/j.ipm.2017.04.005

[25] David E. Losada, Javier Parapar, and Álvaro Barreiro. 2019. When to stop making relevance judgments? A study of stopping methods for building information retrieval test collections. *Journal of the Association for Information Science and Technology* 70, 1 (Nov. 2019), 49–60. https://doi.org/10.1002/asi.24077

[26] David J. C. MacKay and Linda C. Bauman Peto. 1995. A hierarchical Dirichlet language model. *Natural Language Engineering* 1, 3 (1995), 289–308. https://doi.org/10.1017/S1351324900000218

[27] Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the HASOC Track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. In *Proceedings of FIRE 2019*. ACM, New York, NY, USA, 14–17. https://doi.org/10.1145/3368567.3368584

[28] Bhaskar Mitra and Nick Craswell. 2018. *An Introduction to Neural Information Retrieval*. Now Foundations and Trends. https://doi.org/10.1561/1500000061

[29] Alistair Moffat. 2016. Judgment Pool Effects Caused by Query Variations. In *Proceedings of ACM ADCS 2016*. ACM, New York, NY, USA, 65–68. https://doi.org/10.1145/3015022.3015025

[30] Alistair Moffat, Falk Scholer, Paul Thomas, and Peter Bailey. 2015. Pooled Evaluation Over Query Variations: Users Are as Diverse as Systems. In *Proceedings of ACM CIKM 2015*. ACM, New York, NY, USA, 1759–1762. https://doi.org/10.1145/2806416.2806606

[31] NII Testbeds and Community for Information Access Research - 14th International Conference 2019. *NII Testbeds and Community for Information Access Research - 14th International Conference* (Tokyo, Japan). Springer, Cham. https://doi.org/10.1007/978-3-030-36805-0

[32] Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *Proceedings of ACM SIGIR 1998*. ACM, New York, NY, USA, 275–281. https://doi.org/10.1145/290941.291008

[33] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (April 2009), 333–389. https://doi.org/10.1561/1500000019

[34] Tetsuya Sakai and Chin-Yew Lin. 2010. Ranking Retrieval Systems without Relevance Assessments: Revisited. In *Proceedings of EVIA 2010*. National Institute of Informatics (NII), 25–33. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings8/EVIA/05-EVIA2010-SakaiT.pdf

[35] Mark Sanderson and Hideo Joho. 2004. Forming Test Collections with No System Pooling. In *Proceedings of ACM SIGIR 2004*. ACM, New York, NY, USA, 33–40. https://doi.org/10.1145/1008992.1009001

[36] Ian Soboroff, Charles Nicholas, and Patrick Cahan. 2001. Ranking Retrieval Systems without Relevance Judgments. In *Proceedings of ACM SIGIR 2001* (New Orleans, Louisiana, USA). ACM, 66–73. https://doi.org/10.1145/383952.383961

[37] K. Spärck Jones and Cornelis J. van Rijsbergen. 1975. Report on the need for and provision of an 'ideal' information retrieval test collection. *Computer Laboratory* (1975).

[38] Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. 2018. LiMe: Linear Methods for Pseudo-Relevance Feedback. In *Proceedings of ACM SAC 2018*. ACM, New York, NY, USA, 678–687. https://doi.org/10.1145/3167132.3167207

[39] Ellen M. Voorhees. 2001. Evaluation by Highly Relevant Documents. In *Proceedings of ACM SIGIR 2001*. ACM, New York, NY, USA, 74–82. https://doi.org/10.1145/383952.383963

[40] Ellen M. Voorhees. 2018. On Building Fair and Reusable Test Collections Using Bandit Techniques. In *Proceedings of ACM CIKM 2018*. ACM, New York, NY, USA, 407–416. https://doi.org/10.1145/3269206.3271766

[41] Ellen M. Voorhees and Donna K. Harman. 2005. *TREC: Experiment and Evaluation in Information Retrieval*. The MIT Press.

[42] Emine Yilmaz, Javed A. Aslam, and Stephen Robertson. 2008. A New Rank Correlation Coefficient for Information Retrieval. In *Proceedings of ACM SIGIR 2008*. ACM, New York, NY, USA, 587–594. https://doi.org/10.1145/1390334.1390435

[43] Justin Zobel. 1998. How Reliable Are the Results of Large-Scale Information Retrieval Experiments?. In *Proceedings of ACM SIGIR 1998*. ACM, New York, NY, USA, 307–314. https://doi.org/10.1145/290941.291014