


# Relevance feedback for building pooled test collections

Journal of Information Science  
1–18  
© The Author(s) 2023  
Article reuse guidelines:  
[sagepub.com/journals-permissions](https://sagepub.com/journals-permissions)  
DOI: 10.1177/01655515231171085  
[journals.sagepub.com/home/jis](https://journals.sagepub.com/home/jis)  


**David Otero** 

Information Retrieval Lab, CITIC, University of A Coruña, Spain

**Javier Parapar**

Information Retrieval Lab, CITIC, University of A Coruña, Spain

**Álvaro Barreiro**

Information Retrieval Lab, CITIC, University of A Coruña, Spain

## Abstract

Offline evaluation of information retrieval systems depends on test collections. These datasets provide the researchers with a corpus of documents, topics and relevance judgements indicating which documents are relevant for each topic. Gathering the latter is costly, requiring human assessors to judge the documents. Therefore, experts usually judge only a portion of the corpus. The most common approach for selecting that subset is pooling. By intelligently choosing which documents to assess, it is possible to optimise the number of positive labels for a given budget. For this reason, much work has focused on developing techniques to better select which documents from the corpus merit human assessments. In this article, we propose using relevance feedback to prioritise the documents when building new pooled test collections. We explore several state-of-the-art statistical feedback methods for prioritising the documents the algorithm presents to the assessors. A thorough comparison on eight Text Retrieval Conference (TREC) datasets against strong baselines shows that, among other results, our proposals improve in retrieving relevant documents with lower assessment effort than other state-of-the-art adjudicating methods without harming the reliability, fairness and reusability.

## Keywords

Pooling; relevance feedback; reranking; test collections

## 1. Introduction

Reproducible offline evaluation of retrieval systems relies on test collections [1]. These benchmarks include a set of documents, a series of information needs (or topics) and relevance judgements (also known as assessments or qrels), indicating what documents systems should retrieve for each topic [2]. Using these collections, researchers can perform reproducible evaluations to compare the performance of existing and new information retrieval (IR) systems.

From a research point of view, we would want to have the entire set of documents that are relevant for each topic. In this way, we would obtain absolute scores for each system, ensuring a fair comparison between them. Indeed, this was the case for the first retrieval benchmarks [3]. However, modern datasets are so large that judging every document is no longer feasible. Nowadays, pooling is the most common approach when building a new benchmark. When building pooled test collections, assessors only review a subset – the pool – of the entire corpus for each topic. This pool comprises the union of the top- $k$  results for a topic provided by several search systems (the result of each system for all topics is a run). An assessor only judges the documents in the pool, and the rest are assumed to be non-relevant. This methodology builds on the fact that documents most probably to be relevant are going to be located at top positions of the rankings. As a consequence, sampling from a diverse set of runs with sufficiently deep pools will still yield a fair evaluation, even when the number of reviewed documents is low compared with the whole corpus [4].

---

### Corresponding author:

David Otero, Information Retrieval Lab, CITIC, University of A Coruña, A Coruña 15071, Spain.  
Email: [david.otero.freijeiro@udc.es](mailto:david.otero.freijeiro@udc.es)

When the assessment budget is not a problem, we may judge the whole pool. However, if our resources are scarce – if we have a fixed budget of judgements that we can make – we might want to inspect the documents following a specific criterion and not just arbitrarily review the entire pool. We could perform some kind of focused sampling, with the aim of better selecting documents that are more probably to be relevant. In this way, we can optimise the assessor's time and find more relevant documents [5]. We use the name adjudication method to refer to methods that actively decide which document to judge next.

Much work has focused on how to perform this adjudication [6–9]. Past adjudication methods have focused on developing ad hoc heuristics to prioritise the documents to assess. These strategies ignore the documents' content, focusing only on their ranking positions or scores in the runs. We argue that documents can be prioritised by estimating their relevance using a feedback model based on documents' content. Statistical relevance feedback (RF) methods offer a great tool to combine the pooled documents' content with the assessors' judgements information for document adjudication.

In particular, we propose a novel and natural adaptation of well-known statistical RF models to construct pooling-based test collections. While other authors have used RF to build test collections without pooling [10], we adapt this idea to the adjudication of documents in a pooling-based scenario. Based on this proposal, we develop three different pooling strategies. The first one uses the RF model to rerank the whole pool on-the-fly, while the other two rerank the documents of each run system independently.

We thoroughly evaluate our methods on eight standard TREC<sup>1</sup> collections against state-of-the-art pooling strategies using a series of metrics that allow us to study our models from different perspectives. Among other results, we highlight that we can accelerate the rate of relevant documents found by the assessors without losing performance in terms of reliability, fairness and reusability, acknowledging these algorithms as an alternative for building new pooled test collections.

## 2. Background and related work

IR is a field with a strong empirical focus. This mandates extensive experimental and evaluation practices. We can evaluate models from two different perspectives: online and offline experiments. Although online experimentation is better at capturing actual user behaviour, it is costly. On the contrary, offline evaluation is cheaper and more reproducible. For this reason, IR has traditionally relied on offline evaluation as the first step to measure the effectiveness of retrieval methods.

Offline experimentation uses test collections that provide documents, topics and relevance assessments. We can evaluate a system by producing a ranking for each topic. Then, we can compute a metric for this ranking using the judgements and summarise the system's performance with the average of all topic scores.

This paradigm is known as the Cranfield paradigm [11]. It is based on three crucial assumptions [4]. The first one is that the information need of the user can be approximated by topical similarity. The second assumption is that a single set of relevance judgements is valid for any user, that is, relevance is independent of the user. The final assumption is that all relevant documents for a topic are known. Obviously, these assumptions are not always true. However, although they can be damper, making this evaluation still an effective paradigm.

One problem of this methodology is that modern collections are too large to have complete relevance judgements. This is only feasible for small-sized datasets. However, small benchmarks do not represent deployed systems' operational issues. The so-called pooling approach tries to tackle this problem [1,12].

### 2.1. Pooling

Pooling is the standard method for constructing new test collections in campaigns like TREC, NII Testbeds and Community for Information Research (NTCIR)<sup>2</sup> and conference and labs of the evaluation forum (CLEF).<sup>3</sup> Under this setting, the assessors only annotate the top retrieved documents by the systems participating in the competition. Although these judgements are far from complete [4], the resultant collections are valid to provide a good evaluation [4,13–15].

In detail, the process of gathering new assessments in a typical evaluation campaign is as follows: (1) the organisers establish a search task based on a set of documents and a series of topics. In this step, the organisers establish the criteria for what constitutes a relevant document for each topic, (2) different groups submit their results (their runs), (3) a pool of documents is built by taking the top  $k$  (the depth of the pool) documents of each submission for each topic and (4) finally, the assessors only judge the documents in the pool, using the given criteria to decide on the documents' relevance.

If enough resources are available, we may judge the entire pool. If this is not the case, it is interesting to follow a specific strategy to optimise the number of relevant documents found. It is known that the most productive use of the assessors' time is when they judge documents deemed to be relevant [5].

## 2.2. Adjudicating methods

Strategies for producing prioritisation of the pooled documents are adjudicating methods. Extensive research has been carried out on those methods. This research has mainly focused on reducing the assessment effort and maintaining an acceptable quality of the obtained judgements [6–10,16–19].

We may classify adjudication strategies into two categories: static methods and dynamic methods. We refer the reader to Lipani et al. [20] for an extensive taxonomy of different pooling strategies. Static methods, like DocID [2], obtain the order of the pooled documents before any judgement is made. On the contrary, dynamic methods select the next document to assess based on previous judgements. Thus, the order of candidate documents changes while the experts perform the pool assessments. MaxMean (MM) [7] and MoveToFront (MTF) [6] are examples of the latter. Intuitively, dynamic methods may behave better than static ones since they can extract documents from runs contributing with more relevant judgements in the past, avoiding sampling documents from poor runs. However, these strategies may bring some logistical burden to the judging process. Recent applications of dynamic methods in several TREC tracks show that they are applicable in a real scenario [21–27]. In addition, dynamic methods may have the risk of introducing run bias (underestimate runs that do not contribute with many relevant documents in top ranks) or judgement bias (the risk that presenting top-ranked documents first may produce different assessments). In this article, we study if our proposed methods suffer from run bias.

Other works on this line have explored the idea of building new benchmarks with alternative approaches. Sanderson and Joho [10] proposed using RF as an adjudication method without using any pool. Rahman et al. [18] suggested using active learning models to achieve such task. Alternatively, building test collections without participant systems or even assessors in the loop was also explored [28–33]. Some works proposed methods to automatically retrieve documents that normally would only be found by manual runs [34,35].

In this work, we develop a method to adjudicate documents using participant systems. Having diverse participant runs helps improve the quality of the resultant collections [13]. Moffat et al. [36] also demonstrated that user variations are essential for building a new collection of sufficient quality. However, in this work, we assume that the runs already exist, and we centre our efforts only on developing and evaluating the adjudicating strategy.

## 2.3. RF

Initially, search engines only considered the user's original query to produce a ranked list of documents. However, research soon showed that systems might improve the quality of the ranked list by taking into account feedback information given by the user on the presented results. Despite the usefulness of this feedback, real relevance is difficult to obtain in many situations. For this reason, most RF research has focused on developing techniques that could improve retrieval results without user interaction. One example of these techniques is pseudo-relevance feedback (PRF). Under this approach, the top- $r$  documents retrieved by a search engine are assumed to be relevant. Using this set of documents and the original query, these methods obtain an improved and expanded version of the original query, that is, reissued to the search engine to obtain a second ranking. Several PRF methods have been proposed in the past. However, we will centre this work on those based on the statistical language framework since they perform empirically the best [37].

## 2.4. Feedback methods in the Language Modeling (LM) framework

Within this framework, the basic retrieval model is the Kullback–Leibler divergence [38], denoted by  $D(\cdot \parallel \cdot)$ , between the query language model  $\theta_q$  and the document language model  $\theta_d$ . This is rank equivalent to the negative cross entropy between both distributions

$$\text{Score}(d, q) = -D(\theta_q \parallel \theta_d) \stackrel{\text{rank}}{=} \sum_{w \in V} p(w|\theta_q) \log p(w|\theta_d) \quad (1)$$

where  $V$  is the set of words in the vocabulary of the collection. In this case, we use Dirichlet priors to compute a smoothed document model

$$p(w|\theta_d) = \frac{tf(w, d) + \mu \cdot p(w|\theta_C)}{|d| + \mu}$$

where  $tf(w, d)$  is the frequency of  $w$  in  $d$ ,  $p(w|\theta_C)$  is the maximum likelihood estimate (MLE) of  $w$  in the collection and  $\mu$  is a parameter of the smoothing (commonly set to  $\mu = 1000$ ).

Without any extra information, query models (i.e.  $p(w|\theta_Q)$ ) are usually estimated using only the text of the query. However, we can exploit PRF information, that is, assuming that top-ranked documents are relevant (from now on, this set is  $F$ ) to estimate a more accurate query language model  $\theta_F$ . Besides, this new feedback model can be interpolated with the model of the original query to improve the retrieval effectiveness

$$p(w|\theta'_Q) = \alpha p(w|\theta_Q) + (1 - \alpha)p(w|\theta_F) \quad (2)$$

where  $\alpha \in [0, 1]$  controls the importance of the RF. Thus, the goal of a feedback model is to provide an estimation of  $\theta_F$ . With this model, the terms of the original query are expanded and reweighed (equation (2)) to obtain a second retrieval. In this work, we experimented with several models, namely, RM1 [39], RM3 [40], divergence minimisation model (DMM) [41] and MEDMM [42]. We tested the performance of all of them, from the perspective of recall, reliability, fairness and reusability (these characteristics are explained in detail later in the article). In these experiments, the adjudication methods using DMM always stood above those using RM1, RM3 and Maximum-Entropy Divergence Minimization Model (MEDMM). Therefore, we only provide a formal and detailed explanation for DMM.

**2.4.1. DMM.** DMM [41] is an RF technique which assumes that the feedback model  $\theta_F$  should be close to the language model of the pseudo-relevant documents  $F$  but far away from the background model. The model is computed as follows

$$p(w|\theta_F) \propto \exp\left(\frac{1}{1-\lambda} \frac{1}{|F|} \sum_{d \in F} \log p(w|\theta_d) - \frac{\lambda}{1-\lambda} \log p(w|\theta_C)\right) \quad (3)$$

where  $p(w|\theta_d)$ , in this case, is computed using additive smoothing (as recommended by Hazimeh and Zhai [43])

$$p(w|\theta_d) = \frac{tf(w, d) + \gamma}{|d| + \gamma \cdot |V|}$$

and  $p(w|\theta_C)$  is the MLE of  $w$  in the collection. This model has a parameter  $\lambda$  that controls the influence of the collection language model and a parameter  $\gamma$  that controls the smoothed document model. Finally, as is typically done [37,41], this feedback model is interpolated with the original query (see equation (2)).

In an RF scenario, the final ranking is obtained as follows:

1. First, an initial retrieval result is obtained using the original query.
2. Then, the user marks as relevant a set of documents (in a PRF scenario, we would assume the top- $r$  documents as relevant and perform the same steps). This set of documents is the relevance set  $F$ .
3. Using the information from  $F$ , we compute a feedback model  $p(w|\theta_F)$ , and then interpolate it with the original query to compute a new query model  $p(w|\theta'_Q)$  (see equation (2)).
4. Finally, this query model is plugged into equation (1) to obtain the final ranking.

### 3. RF for pooling

Pooling is a successful technique to reduce the cost of gathering new assessments when building new test collections. If applied correctly, we can assume that enough relevant documents have been found and rely on the reusability of that collection for the future [4]. Also, considering the results of diverse participant systems aids in the discovery and evaluation of documents that would otherwise have gone unnoticed, lowering the collection's quality [44]. On the contrary, PRF methods have been demonstrated as a powerful approach for improving systems' effectiveness for the ad hoc search task [37,39].

In this work, we want to study RF techniques to select the documents that merit assessments when building retrieval test collections. The RF methods presented above are usually applied for the PRF task when gathering feedback from the user is difficult. In our case, we take advantage of the real RF of the judges. We use the documents that assessors judge relevant as the relevance set  $F$  to estimate a new feedback model  $\theta_F$  and then a query model  $\theta'_Q$  (as in equation (2)). We then use that query model to prioritise the unjudged documents using the ranking method from equation (1).

**Algorithm 1** DMM

---

```

1: Input
2:    $\mathcal{P}_q$  set of rankings for a topic.
3:    $q$  a topic query.
4:    $b$  budget size.
5: Output
6:    $\mathcal{R}$  set of judgements for a topic.
7:  $\mathcal{R} \leftarrow \emptyset$  ▷ Set of judgements.
8:  $\mathcal{F} \leftarrow \emptyset$  ▷ Relevance set.
9:  $\mathcal{P} \leftarrow \text{get\_pooled\_docs}(\mathcal{P}_q)$  ▷ Union of top-k documents.
10:  $r \leftarrow \text{get\_initial\_ranking}(\mathcal{P}, q)$ 
11: while  $|\mathcal{R}| < b$  do
12:    $d \leftarrow \text{pop\_top\_ranked\_doc}(r)$ 
13:    $\mathcal{P} \leftarrow \mathcal{P} \setminus \{d\}$ 
14:    $j \leftarrow \text{judge}(d, q)$ 
15:    $\mathcal{R} \leftarrow \mathcal{R} \cup \{(d, j)\}$ 
16:   if  $j > 0$  then ▷ If the document is relevant.
17:      $\mathcal{F} \leftarrow \mathcal{F} \cup \{d\}$ 
18:      $r \leftarrow \text{rerank}(\mathcal{P}, \mathcal{F}, q)$  ▷ Rerank with DMM.
19:   end if
20: end while

```

---

### 3.1. Common naming

Before giving the details of the algorithms we propose, it is important to explain some details about the naming that we employ later. As we said earlier in this article, we use the term *run* to refer to the results of a particular system for all the topics in the benchmark. Since the adjudicating methods operate on a per-topic basis (the assessments for each topic are gathered independently of each other), we use the term *ranking* to refer to the results of a particular system for a particular topic.

We now explain the details of the three methods we propose in this work. In the first method (section 3.2), we use the feedback model to rerank all pooled documents and use this ranking to prioritise how assessors review the pool. In the other two methods (section 3.3), we use the feedback method to rerank the documents of each ranking independently.

### 3.2. Reranking the pool

The main idea of this approach is to use the estimated feedback model to prioritise all pooled documents as a whole. Here, we gradually enlarge the RF set  $F$  with the relevant documents. After a new relevant document is added to the set, we update our feedback model estimate and rank the documents in the pool. Assessors inspect the documents from the pool according to that ranking. When a new relevant document appears, we add the document to the relevance set, update the model estimate and reorder the pool again, repeating the loop until the assessment budget is consumed.

We describe the method in detail in Algorithm 1. This algorithm takes as inputs: (1) the set of pooled rankings for a topic, (2) the budget of judgements to perform (we assume that  $b$  is less than the size of the pool) and (3) the original query for a topic, which we construct by joining the title and the description of the original TREC topic.<sup>4</sup>

In our experiments, using long queries (title + description) instead of short ones (only title) yielded better results. Therefore, from now on, we assume the use of long queries.

The first step of this approach is to obtain a first ranking of the pooled documents (line 10). Since just taking the union of the top- $k$  documents of each participant does not give any order on the documents, we have to decide which document we judge first. We rank the documents by their alphabetical order to avoid introducing any biases here.

Once this first rank is created, we keep sampling and judging the top document from it until a relevant one appears (line 16). When the first relevant document arrives, we use it as the only document in the relevance set to estimate a new query model with DMM and obtain a new ranking of the rest of the pooled documents (line 18). Then, the top-ranked document of this new ranking is judged (lines 12 and 14). Therefore, the ranking of pooled documents changes every time we find a relevant document. This process continues until the budget is consumed (line 11).

**Algorithm 2** MTF + DMM

---

```

1: Input
2:  $\mathcal{P}_q$  set of rankings for a topic.
3:  $q$  a topic query.
4:  $b$  budget size.
5: Output
6:  $\mathcal{R}$  set of judgements for a topic.
7:  $\mathcal{R} \leftarrow \emptyset$   $\triangleright$  Set of judgements.
8:  $\mathcal{F} \leftarrow \emptyset$   $\triangleright$  Relevance set.
9: for  $i \leftarrow 0, |\mathcal{P}_q|$  do
10:  $p[i] \leftarrow 0$   $\triangleright$  Count of non-relevants found by each ranking.
11: end for
12:  $r \leftarrow \text{select\_random\_ranking}(\mathcal{P}_q, p)$  Select next ranking at random among the ones with the lowest non-relevant count.
13: while  $|\mathcal{R}| < b$  and  $|\mathcal{P}_q| \neq 0$  do
14:  $d \leftarrow \text{pop\_top\_ranked\_doc}(r)$ 
15: if  $r$  is empty then
16:  $\mathcal{P}_q \leftarrow \mathcal{P}_q \setminus \{r\}$ 
17: end if
18: if  $d \in \mathcal{R}$  and  $r$  is empty then
19:  $r \leftarrow \text{select\_random\_ranking}(\mathcal{P}_q, p)$ 
20: end if
21: if  $d \notin \mathcal{R}$  then
22:  $j \leftarrow \text{judge}(d, q)$ 
23: if  $j > 0$  then  $\triangleright$  if  $d$  is relevant
24:  $\mathcal{F} \leftarrow \mathcal{F} \cup \{d\}$ 
25: for all  $s \in \mathcal{P}_q$  do
26:  $s \leftarrow \text{rerank}(s, \mathcal{F}, q)$   $\triangleright$  Rerank  $s$  with DMM.
27: end for
28: if  $r$  is empty then
29:  $r \leftarrow \text{select\_random\_ranking}(\mathcal{P}_q, p)$ 
30: end if
31: else  $\triangleright$  if  $d$  is not relevant
32: for all  $s \in \{s \in \mathcal{P}_q : s \text{ contains } d\}$  do
33:  $p[s] \leftarrow p[s] + 1$ 
34: end for
35:  $r \leftarrow \text{select\_random\_ranking}(\mathcal{P}_q, p)$ 
36: end if
37:  $\mathcal{R} \leftarrow \mathcal{R} \cup \{d\}$ 
38: end if
39: end while

```

---

### 3.3. Reranking the participants

Recently, some works demonstrated that using dynamic methods that focus on sampling documents from good systems while avoiding the poor ones is a fruitful approach for gathering new assessments in a pooling scenario [7,45]. For this reason, we also want to explore the use of RF in combination with a dynamic strategy that selects the documents on a per-run basis. This means that documents are sampled from each system independently. Examples of these strategies are MTF [6] or Bayesian Bandits [7].

**3.3.1. MTF + DMM.** The idea here is to combine an RF model for reranking documents from each participant with a strategy to select the ranking from which we sample the next document to assess. In particular, we employ MTF [6] to select the next ranking to sample and DMM to rerank each participant every time a new relevant document appears.

MTF is a method that maintains a priority for each ranking. Initially, that priority is equal to the pool depth  $k$ . The algorithm selects the top-ranked document from a ranking among the ones with the highest priority. If the sampled document is relevant, it keeps retrieving from that ranking. When a non-relevant document appears, the priority of the current ranking is decreased, and the algorithm jumps to another one with the highest priority. If there are ties among the rankings with the highest priority, the algorithm selects one at random.

In our proposal, which we coin as MTF + DMM, we use the MTF strategy for selecting from which ranking to sample next. Then, before picking the top-ranked document, we use the RF estimate computed with DMM for reordering the documents of that particular ranking. Every time a new relevant document appears, we update the estimate of the feedback model. We repeat this loop until the budget is consumed.

Algorithm shows the details of this method for a particular topic. As we explained before, the original formulation of MTF maintains a priority for each ranking and decreases the priority when a non-relevant document appears. For description purposes, instead of having a priority for each ranking, we count how many non-relevant documents have appeared in each ranking (lines 10 and 33) and select the next one based on this. In practice, this is the same as the original MTF formulation.

First, we select a ranking at random among those that have provided the least number of non-relevant documents (line 12). Initially, this number is the same for all pooled rankings, so we select a random one among all. Next, we sample the top-ranked document from it (line 14). At this point, we may encounter three scenarios. First, the sampled document is already reviewed, and the ranking we have just sampled from is exhausted. In this case, we select a new ranking (line 19) and go over the while loop again. Second, the document is already reviewed, and the ranking is not exhausted. In that case, nothing happens, and we repeat the loop, sampling the next document from the same ranking (since we do not go into any of the if branches and we do not update the content of  $r$ ). Last, the document is not reviewed yet. In this case, the process continues with its assessment (line 22). If the document is relevant, we add it to the relevance set (line 24) and rerank the documents of every ranking with the updated feedback model estimate (line 25), and if the current one is exhausted, select a new one (line 28). On the contrary, if the document is not relevant, we update the priority of each ranking that retrieved this document (line 33) and jump to a new ranking. In both cases, we update the judgements set (lines 37). All this process continues until the budget is consumed.

**3.3.2. MM + DMM.** We also developed a third algorithm based on the same idea as MTF + DMM, but selecting the rankings with the MM [7] Bayesian Bandit model.

Bayesian Bandits apply Bayesian principles to the multi-armed bandit problem, formalising the uncertainty associated with the probabilities of pulling a positive reward from playing (in our case, sampling a relevant document). Under this setting, we associate each bandit (in our case, each ranking) with its probability of giving a relevant document. At first, since we do not have any information, we assume a uniform prior over the rankings. This is equivalent to assign each ranking a *Beta* distribution  $Beta(1, 1)$ , since the uniform distribution is an especial case of  $Beta(\alpha, \beta)$  when  $\alpha$  and  $\beta$  are both 1. Then, every time we sample a ranking, we can see this process as sampling from a Bernoulli distribution or conversely, a Binomial with just one trial. This trick allows us to easily compute the posterior distribution of each ranking since the *Beta* distribution is the conjugate prior for the binomial. Thus, given that a ranking has been sampled  $n$  times, with  $\alpha$  wins (relevant documents) and  $\beta$  loses (non-relevant documents), the probability of getting a positive outcome is given by  $Beta(\alpha, \beta)$ . With this formalisation, the original authors experimented with two different approaches for the task of pooling [16]. The first one, called Thompson Sampling, consists in sampling each *Beta* distribution independently and choosing the ranking that yields a higher value. The other one, MM, consists of choosing the ranking with the higher expectation of the posterior distribution. The expectation of a *Beta* distribution is given by  $\alpha/(\alpha + \beta)$ . We choose the latter since in the original article outperformed Thompson Sampling.

Similarly to the previous algorithm, we use MM for selecting from which ranking we sample next and DMM for reordering the documents before sampling. We begin by selecting a random ranking since they all have the same expectation. If its top-ranked document is relevant, we add it to the relevance set and reorder the documents from the current ranking, and again pick the new top-ranked document. When a non-relevant document appears, we update the distributions of all the rankings that retrieved it and jump to another one among the ones with the highest expectation. The process continues until the budget is consumed.

Having outlined the details of our methods, we now proceed to describe the details of our experimental setup, and the evaluation approach followed to test the performance of our methods.

## 4. Experimental setup

In this section, we explain the evaluation approach for assessing the performance of the proposed methods. We describe our evaluation methodology, the baselines and the parameter settings.

**Table 1.** Statistics of the collections used for experimentation.

	TREC5 (train)	TREC6 (test)	TREC7 (test)	TREC8 (test)	TREC9 (test)	CT14 (train)	CT15 (test)	CT16 (test)
Number of topics	50	50	50	50	50	30	30	30
Number of runs	101	46	84	71	59	102	102	115
Number of teams	30	33	41	38	21	26	36	26
Avg. pool size	2662	1445	1606	1733	1404	770	591	666
Max. pool size	4419	1905	2579	2977	2978	987	859	906
Min. pool size	1580	914	1025	1042	710	554	351	452
Pool depth ( $k$ )	100	100	100	100	100	20	20	15
Avg. number of relevants	110	92	93	94	52	82	95	112
Max. number of relevants	594	474	361	347	519	304	390	479
Min. number of relevants	1	3	7	6	1	10	2	3
Avg. unique documents	1808	963	985	1169	972	617	424	496

#### 4.1. Collections

We have performed experiments<sup>5</sup> on eight different collections. TREC5–8 are classic testbeds associated with the ad hoc retrieval task, while TREC9 comes from the web track. CT14–16<sup>6</sup> are newer collections created in the TREC Clinical Decision Support track. Table 1 depicts the main information of each collection: the number of topics, the number of pooled runs, the number of participant teams, the pool depth, the pool size per topic, the number of relevant documents per topic and the average of unique documents per topic. To compute the last one, we sum the number of unique documents of each team for each topic. We also include which collections we used for optimising the models' parameters and for testing them.

#### 4.2. Pooling baselines

We selected MTF [6] and MM [7] as baselines to assess the performance of our proposal. This decision is motivated by recent results obtained by Altun and Kutlu [45]. In this work, they studied the reliability, fairness and reusability (same metrics that we employ here) of MTF, MM and Hedge, among other methods. Results showed that MTF was more robust than the others regarding these metrics. On the contrary, MM was successfully employed to build the TREC 2017 Common Core Track collection [21,22]. Also, since these algorithms form the basis of two of our three proposed methods, we want to examine how employing RF-based reranking performs with respect to them. For these reasons, we believe they are strong baselines we can use here.

#### 4.3. Parameter setting

For reranking, we used the retrieval algorithm based on the KL divergence (See equation (1)) with Dirichlet priors smoothing with the parameter  $\mu$  set to 1000. In addition, for DMM, we used the values recommended by Hazimeh and Zhai [43], we set the parameter  $\lambda$  (see equation (3)) to 0.03 and the additive smoothing parameter  $\gamma$  for the estimation of  $p(w|\theta_d)$  to 1.

We also swept the number of expansion terms  $e$  among  $\{5, 10, 25, 50, 75, 100\}$  and the interpolation parameter  $\alpha$  (see equation (2)) from 0 to 1 in steps of 0.1.

When creating the pools and selecting the documents from the runs, we use the depth ( $k$ ) specified in Table 1. We employ these specific values because they were used in the original construction of the collections, and there are no judgements available for documents outside this depth. However, we must note that the use of  $k$  in a real scenario may be unnecessary since our algorithms do not need to limit the pool's depth to control the assessor budget.

#### 4.4. Metrics

We evaluated the proposals from four different perspectives: recall, reliability, fairness and reusability.

**4.4.1. Recall.** We study the pooling strategies in terms of their ability to identify relevant documents early, that is, the sooner they obtain high recall values, the better. We do this as follows: for each topic, an adjudicating method creates a sequence of judgements of the pooled documents. We can compute  $\text{recall}@n$  at any point in this sequence, where  $n$  is the



**Table 2.** Tuned parameters after optimisation on the training collections.

Algorithm	TREC5		CT14	
	$e$	$\alpha$	$e$	$\alpha$
DMM	75	0.1	75	0
MTF + DMM	75	0	75	0
MM + DMM	100	0	50	0.1

DMM: divergence minimisation model; MTF: MoveToFront; MM: MaxMean.

$e$  is the number of expansion terms, and  $\alpha$  controls the interpolation of the feedback model with the original query.

number of judgements. The most productive use of assessors' time is when they judge relevant documents. Therefore, our main metric is the recall averaged over the set of topics in each collection.

The optimal values of the parameters for the feedback models (the number of expansion terms  $e$  and the interpolation parameter  $\alpha$ ) were learned by optimising the area under the curve (AUC) of this recall curve on the training collections. The values of these parameters after optimisation are shown in Table 2. It is interesting to note that the low figures of  $\alpha$  acknowledge that the feedback model can capture the documents' relevance without the information of the original query.

**4.4.2. Reliability.** We also study the reliability of the methods to induce the same ranking of systems as the official qrels (we use the term qrels for referring to the set of judgements of a collection). To evaluate it, we compute two different ranking correlations (Kendall's  $\tau$  [46,47] and  $\tau_{AP}$  [48]) between the official rankings of systems and the ranking obtained with each adjudicating method.

**4.4.3. Fairness.** We must consider if the collection can provide a fair comparison between the runs that participated in the pool. Following the same approach as Voorhees [22], to evaluate the fairness, we compute the maximum drop (negative change) suffered by a run when ranking it with the evaluated method compared with the official ranking. In particular, we build a new qrels file using each of the adjudicating methods proposed. Then, we rank the runs using this reduced qrels file. We compute the difference between a run's position in the official ranking and the ranking obtained with the test qrels. We do this for every run. The maximum negative drop suffered by a run is what we call MaxDrop. A high MaxDrop means that a system is treated differently in both rankings.

**4.4.4. Reusability.** A test collection is reusable if it is able to correctly evaluate runs that did not contribute to the pool. We performed a leave-out-uniques (LOUs) experiment to measure the reusability [4]. This type of tests are a common way of evaluating the reusability of retrieval test collections [13,23]. In these tests, the ground-truth rankings of a team's runs are compared with the rankings that those runs would have obtained if the team had not participated in the construction. When these rankings are similar, we can conclude that the collection is reusable. In this work, in particular, we perform the experiment in this way. We create a reduced qrels file for each team that participated in the competition without that team's runs and the corresponding adjudicating method. Next, we rank the participant runs using both the ground-truth qrels and the reduced qrels. Finally, we compute the Kendall's  $\tau$  correlation between both rankings. Our evaluation measure is Kendall's  $\tau$  averaged overall teams.

For all cases, where we rank the participant runs, we score them using average precision (AP) with a cut-off of 1000 averaged overall topics.

## 5. Results and discussion

In this section, we provide the results and discuss the performance of our proposals.

### 5.1. Recall

We summarise recall evaluation results in Table 3. In this table, we report the AUC (averaged overall topics) of all evaluated methods at different levels of judgement budget. We also flag the statistically significant improvements according to the randomised version of the Tukey Honestly Significant Difference (HSD) test [49,50], which ensures that the family-wise error is not larger than the confidence level  $\alpha$  [51,52].

**Table 3.** Values of AUC (averaged overall topics) for all evaluated methods at different budgets.

	Budget of judgements per topic					Budget of judgements per topic				
	100	300	500	750	1000	100	300	500	750	1000
	TREC5 (train)					TREC6 (test)				
MTF (*)	0.27 <sup>§</sup>	0.42	0.51	0.59	0.64	0.39 <sup>§</sup>	0.56 <sup>§</sup>	0.65	0.71	0.76
MM (†)	0.27 <sup>§</sup>	0.45 <sup>§</sup>	0.55 <sup>*§</sup>	0.64*	0.70*	<b>0.42</b> <sup>§</sup>	<b>0.62</b> <sup>*§</sup>	<b>0.71</b> <sup>*§</sup>	0.77 <sup>*§</sup>	<b>0.82</b> <sup>*§</sup>
DMM (§)	0.21	0.40	0.52	0.61	0.67	0.31	0.51	0.61	0.70	0.77
MTF + DMM (§)	0.26 <sup>§</sup>	0.42	0.51	0.59	0.65	0.38 <sup>§</sup>	0.57 <sup>§</sup>	0.67 <sup>§</sup>	0.75	0.80
MM + DMM (‡)	<b>0.28</b> <sup>§</sup>	<b>0.48</b> <sup>*§</sup>	<b>0.58</b> <sup>*§</sup>	<b>0.66</b> <sup>*§</sup>	<b>0.72</b> <sup>*§</sup>	<b>0.42</b> <sup>§</sup>	<b>0.62</b> <sup>*§</sup>	<b>0.71</b> <sup>*§</sup>	<b>0.78</b> <sup>*§</sup>	<b>0.82</b> <sup>*§</sup>
	TREC7 (test)					TREC8 (test)				
MTF (*)	0.35 <sup>§</sup>	0.54 <sup>§</sup>	0.64	0.71	0.77	0.35 <sup>§</sup>	0.54 <sup>§</sup>	0.64	0.71	0.76
MM (†)	0.39 <sup>*§</sup>	0.60 <sup>*§</sup>	0.70 <sup>*§</sup>	0.77 <sup>*§</sup>	0.82 <sup>*§</sup>	0.42 <sup>*§</sup>	0.64 <sup>*§</sup>	0.73 <sup>*§</sup>	0.80 <sup>*§</sup>	0.84 <sup>*§</sup>
DMM (§)	0.27	0.49	0.61	0.71	0.77	0.26	0.48	0.60	0.69	0.75
MTF + DMM (§)	0.35 <sup>§</sup>	0.58 <sup>§</sup>	0.68 <sup>§</sup>	0.76 <sup>§</sup>	0.81 <sup>§</sup>	0.33 <sup>§</sup>	0.55 <sup>§</sup>	0.66 <sup>§</sup>	0.74 <sup>§</sup>	0.80 <sup>§</sup>
MM + DMM (‡)	<b>0.40</b> <sup>*§</sup>	<b>0.63</b> <sup>*§</sup>	<b>0.72</b> <sup>*§</sup>	<b>0.79</b> <sup>*§</sup>	<b>0.83</b> <sup>*§</sup>	<b>0.43</b> <sup>*§</sup>	<b>0.66</b> <sup>*§</sup>	<b>0.75</b> <sup>*§</sup>	<b>0.81</b> <sup>*§</sup>	<b>0.85</b> <sup>*§</sup>
	TREC9 (test)					CT14 (train)				
MTF (*)	0.35	0.54	0.64	0.71	0.76	0.16	0.37	0.52	0.63	0.66
MM (†)	0.35	0.57	0.69	0.77*	0.81*	0.18	0.44	0.60*	0.71*	0.73*
DMM (§)	0.33	0.56	0.67	0.76	0.81*	<b>0.33</b> <sup>*†‡</sup>	<b>0.63</b> <sup>*†‡</sup>	<b>0.75</b> <sup>*†‡</sup>	<b>0.81</b> <sup>*†‡</sup>	<b>0.83</b> <sup>*†‡</sup>
MTF + DMM (§)	0.36	0.57	0.67	0.75	0.80	0.26 <sup>*†</sup>	0.53 <sup>*†</sup>	0.67*	0.76*	0.78*
MM + DMM (‡)	<b>0.37</b>	<b>0.61</b> *	<b>0.72</b> *	<b>0.79</b> *	<b>0.84</b> *	0.25*	0.51*	0.65*	0.74*	0.76*
	CT15 (test)					CT16 (test)				
MTF (*)	0.21	0.45	0.60	0.66	0.66	0.14	0.35	0.51	0.61	0.62
MM (†)	0.23	0.49	0.65	0.70	0.70	0.17	0.42	0.59*	0.67*	0.68*
DMM (§)	<b>0.30</b> <sup>*†</sup>	<b>0.59</b> <sup>*†</sup>	<b>0.71</b> <sup>*†</sup>	<b>0.76</b> <sup>*†</sup>	<b>0.76</b> <sup>*†</sup>	<b>0.23</b> <sup>*†</sup>	<b>0.51</b> <sup>*†</sup>	<b>0.66</b> <sup>*†‡</sup>	<b>0.73</b> <sup>*†‡</sup>	<b>0.74</b> <sup>*†‡</sup>
MTF + DMM (§)	0.26	0.54*	0.68*	0.73*	0.73*	0.21*	0.47*	0.62*	0.70*	0.71*
MM + DMM (‡)	0.26	0.53*	0.67*	0.72*	0.72*	0.20	0.44*	0.60*	0.68*	0.69*

AUC: area under the curve; MTF: MoveToFront; MM: MaxMean; DMM: divergence minimisation model.

For each level and collection, the best values are bolded. Statistically significant improvements according to the randomised Tukey HSD test [49,50] (permutations = 1,000,000,  $\alpha = 0.05$ ).

w.r.t. MTF, MM, DMM, MTF + DMM and MM + DMM are superscripted with \*, †, §, \* and ‡, respectively.

Overall, these results confirm that using real RF is an effective option when we aim to build a new testbed with less assessor effort. We obtain better results when using DMM with MM and MTF than with the original versions of these methods.

In the TREC5-9 collections, MM + DMM always gets the best figures, tying with MM in TREC6. In these benchmarks, it is statistically significantly better than MTF in most cases, although it never reaches to obtain significant differences with MM. We can also observe that in the TREC9 dataset, there are fewer significant differences among all methods. We believe this is due to the lower density of relevant documents per topic that this collection has. On average, the rest of the collections have more than 90 relevant judgements per topic, while TREC9 only has 52.

In the case of CT testbeds, DMM is always the best-performing method. In the CT14 and CT15 collections, it obtains statistically significant improvements over both baselines in every evaluated budget. In the CT16 dataset, it is always statistically significantly better than MTF, while the significant differences with respect to MM do not appear until 500 judgements. It is interesting to note that the CT collections were built using shallow pools, unlike the previous ones that used deep pools, as shown in Table 1.

## 5.2. Reliability

Reliability results are summarised in Table 4. This table shows the number of judgements each evaluated strategy needs to obtain high values of Kendall's  $\tau$  correlation with the official ranking of systems. We also analyse  $\tau_{AP}$  correlation. This metric was designed with the idea that errors in high positions are worse when comparing two rankings of search systems than errors in deeper positions. Kendall's  $\tau$  penalises both errors the same. Typically, when evaluating search systems, we aim to find the best ones, those at top positions. Thus, we think that  $\tau_{AP}$  correlation is a good measure to

**Table 4.** Number of judgements per topic needed to obtain values of 0.90 for Kendall's  $\tau$  correlation and  $\tau_{AP}$  correlation between the official ranking of systems and the ranking produced by each adjudicating method.

	$\tau \geq 0.90$	$\tau_{AP} \geq 0.90$	$\tau \geq 0.90$	$\tau_{AP} \geq 0.90$
	TREC5 (train)		TREC6 (test)	
MTF	47	198	144	<b>144</b>
MM	49	<b>93</b>	<b>124</b>	147
DMM	149	402	287	390
MTF + DMM	65	399	194	194
MM + DMM	<b>44</b>	135	143	154
	TREC7 (test)		TREC8 (test)	
MTF	76	149	43	108
MM	70	131	<b>33</b>	<b>101</b>
DMM	134	303	203	431
MTF + DMM	77	105	73	185
MM + DMM	<b>62</b>	<b>101</b>	34	110
	TREC9 (test)		CT14 (train)	
MTF	53	157	132	189
MM	161	314	147	227
DMM	131	289	<b>53</b>	217
MTF + DMM	57	164	96	<b>148</b>
MM + DMM	<b>47</b>	<b>153</b>	141	198
	CT15 (test)		CT16 (test)	
MTF	137	199	204	276
MM	152	205	249	269
DMM	151	242	182	287
MTF + DMM	<b>80</b>	199	<b>164</b>	283
MM + DMM	137	<b>184</b>	232	<b>259</b>

MTF: MoveToFront; MM: MaxMean; DMM: divergence minimisation model.

For each collection and each correlation, the best values are highlighted in bold.

evaluate the adjudicating strategies, we develop here. Note that we do not perform a statistical test in this case since it is a global metric. In addition, although we report the results on the training datasets here, we only include those on the test ones in the following comments.

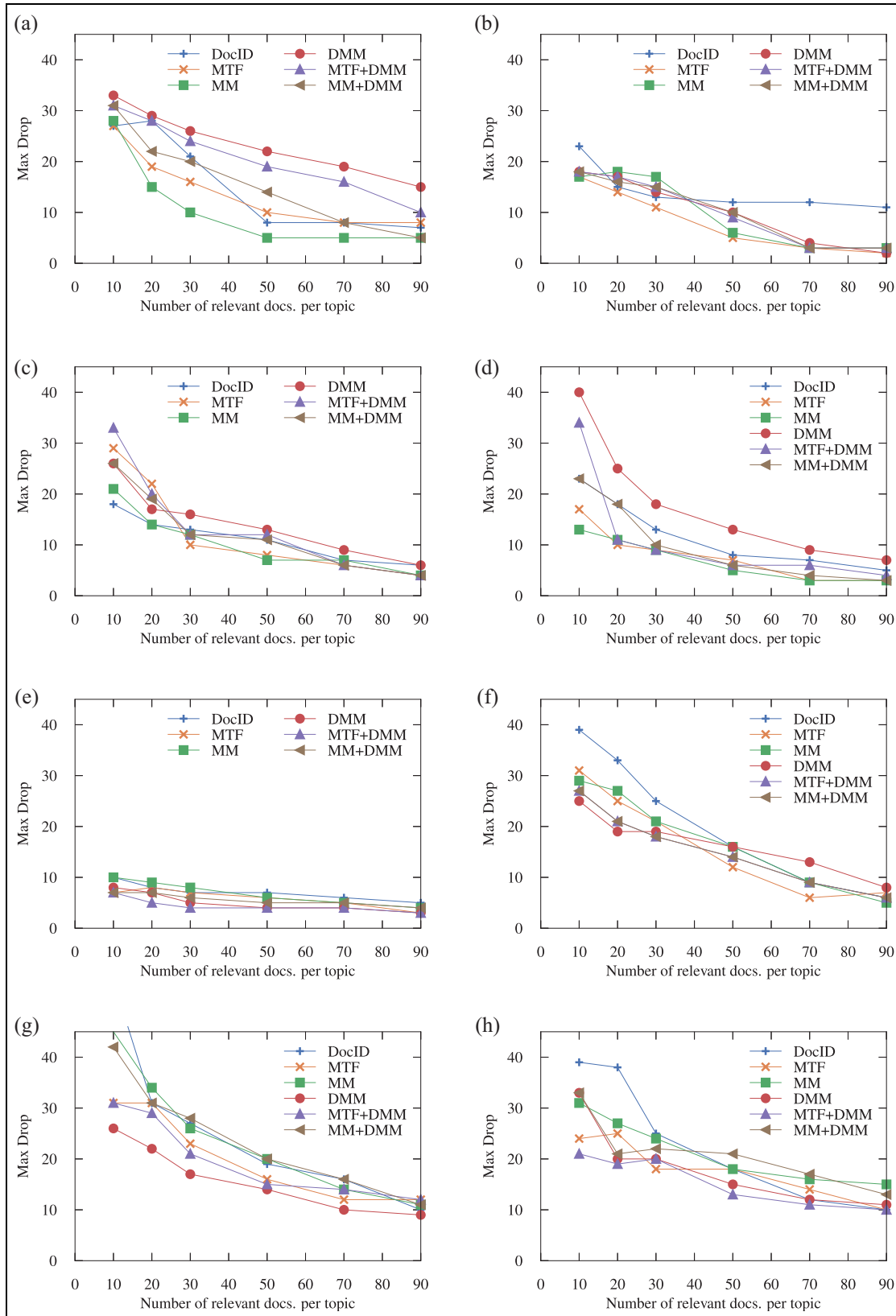
Overall, these results acknowledge that we need very few judgements to obtain strong correlations with the official ranking of runs. In particular, if our focus is Kendall's  $\tau$ , the minimum number of judgements varies between 33 and 164. On the contrary, if we focus on  $\tau_{AP}$  correlation, this number lies between 101 and 259. These figures are tiny in comparison with the average size of the pool in the original qrels. This means we can greatly reduce the assessor effort and still produce a reliable benchmark.

When comparing our methods against the baselines, we observe the following results. The method using RF solely (DMM) or its combination with MTF (MTF + DMM) or MM (MM + DMM) achieve the best results on TREC7, TREC9, CT15 and CT16 (that is, four out of six test collections). On the contrary, on TREC6 and TREC8, MM and MTF are the best-performing methods.

The recall and reliability results we have presented above acknowledge that these algorithms are a good choice if we aim to find relevant documents early, and we can build reliable benchmarks with lower assessor costs. Nonetheless, we do not know if we are harming the fairness and reusability of the constructed collection. In the following sections, we analyse to what extent this effect exists.

### 5.3. Fairness

We report fairness results in Figure 1. This figure shows, for each collection, the maximum drop in the position that a run suffers when comparing it with the official ranking. We also include the traditional depth- $k$  pooling (coined as DocID) used in evaluation campaigns such as TREC [2]. This method produces fair and reusable benchmarks – when assessors inspect the entire pool. Thus, we believe our performance should be compared against it. However, in this case, we



**Figure 1.** Values of MaxDrop for all evaluated methods at different levels of relevant documents per topic. The x-axis shows the number of relevant documents per topic. The y-axis indicates the maximum drop in rank suffered by a participant run: (a) TREC5 (train), (b) TREC6 (test), (c) TREC7 (test), (d) TREC8 (test), (e) TREC9 (test), (f) CT14 (train), (g) CT15 (test) and (h) CT16 (test).

**Table 5.** Average of Kendall's  $\tau$  correlation between ranking induced by ground-truth qrels and ranking induced by the qrels without each team's runs.

	Relevant documents per topic					Relevant documents per topic				
	10	30	50	70	90	10	30	50	70	90
	TREC5 (train)					TREC6 (test)				
DocID	0.81	0.88	0.92	0.94	0.96	0.56	0.75	0.85	0.89	0.90
MTF	<b>0.86</b>	0.92	0.94	<b>0.96</b>	<b>0.97</b>	0.61	0.81	<b>0.92</b>	<b>0.95</b>	<b>0.97</b>
MM	<b>0.86</b>	<b>0.93</b>	<b>0.95</b>	<b>0.96</b>	<b>0.97</b>	0.58	0.80	0.91	0.94	0.95
DMM	0.82	0.89	0.92	0.94	0.96	<b>0.67</b>	<b>0.84</b>	0.91	<b>0.95</b>	0.96
MTF + DMM	0.84	0.90	0.93	0.95	0.96	0.61	0.79	0.90	0.94	0.96
MM + DMM	0.85	0.92	0.94	<b>0.96</b>	<b>0.97</b>	0.60	0.80	0.90	0.94	0.96
	TREC7 (test)					TREC8 (test)				
DocID	<b>0.79</b>	<b>0.88</b>	0.90	0.92	0.94	0.76	0.85	0.91	0.94	0.95
MTF	0.71	0.87	<b>0.93</b>	<b>0.95</b>	<b>0.97</b>	0.78	<b>0.92</b>	0.94	0.96	0.97
MM	0.70	0.87	<b>0.93</b>	<b>0.95</b>	<b>0.97</b>	0.82	<b>0.92</b>	<b>0.95</b>	<b>0.97</b>	<b>0.98</b>
DMM	0.75	0.87	<b>0.93</b>	<b>0.95</b>	0.96	0.74	0.87	0.91	0.93	0.95
MTF + DMM	0.75	0.87	0.92	<b>0.95</b>	0.96	0.78	0.90	0.93	0.95	0.96
MM + DMM	0.73	0.87	0.92	<b>0.95</b>	<b>0.97</b>	<b>0.83</b>	<b>0.92</b>	0.94	0.96	0.97
	TREC9 (test)					CT14 (train)				
DocID	0.83	0.93	0.94	0.95	0.95	0.62	0.77	0.88	0.94	0.95
MTF	0.88	0.92	0.95	0.96	<b>0.98</b>	0.77	<b>0.88</b>	<b>0.93</b>	<b>0.96</b>	<b>0.96</b>
MM	0.82	0.90	0.94	0.96	0.97	0.72	0.85	0.91	0.94	<b>0.96</b>
DMM	0.85	0.93	0.95	0.95	0.96	<b>0.83</b>	0.86	0.91	0.94	0.95
MTF + DMM	0.89	<b>0.94</b>	<b>0.96</b>	<b>0.97</b>	<b>0.98</b>	0.81	0.90	<b>0.93</b>	0.95	<b>0.96</b>
MM + DMM	<b>0.90</b>	<b>0.94</b>	<b>0.96</b>	<b>0.97</b>	0.97	0.75	0.85	0.91	0.94	0.95
	CT15 (test)					CT16 (test)				
DocID	0.48	0.72	0.81	0.87	0.91	0.63	0.74	0.84	0.87	0.91
MTF	0.68	0.81	0.88	<b>0.92</b>	<b>0.94</b>	0.77	<b>0.85</b>	<b>0.89</b>	<b>0.91</b>	<b>0.92</b>
MM	0.64	0.79	0.86	0.91	<b>0.94</b>	0.67	0.78	0.83	0.87	0.89
DMM	<b>0.81</b>	0.83	0.88	0.91	0.92	0.76	<b>0.85</b>	0.88	0.90	0.91
MTF + DMM	0.76	<b>0.85</b>	<b>0.90</b>	<b>0.92</b>	<b>0.94</b>	<b>0.79</b>	<b>0.85</b>	0.88	<b>0.91</b>	<b>0.92</b>
MM + DMM	0.70	0.81	0.88	<b>0.92</b>	<b>0.94</b>	0.70	0.80	0.84	0.88	0.90

MTF: MoveToFront; MM: MaxMean; DMM: divergence minimisation model.

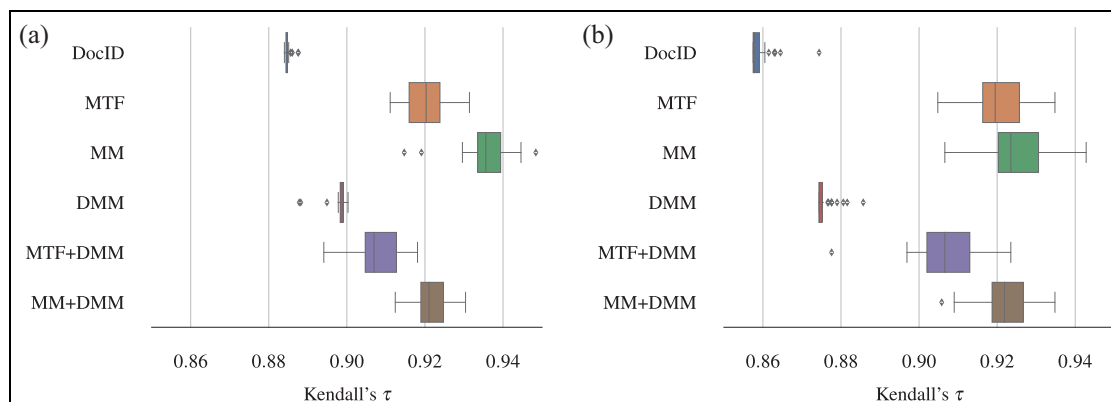
For each level and collection, the best values are bolded.

compare all the algorithms at different numbers of relevant documents per topic, instead of just judgements per topic. DocID was not designed to prioritise the documents in a fixed-budget scenario. Instead, the idea behind this algorithm is to avoid assessor bias when there is enough budget to cover the entire pool. For this reason, we can evaluate if the relevant documents found by DocID and the other methods are fair to evaluate the pooled runs.

Overall, we can observe that no best-performing method arises over the rest. In TREC5, MM obtains the best figures for all levels. In CT15, it is DMM the method that wins. However, there is no clear winner for the rest of the benchmarks.

As we noted previously, dynamic methods that change the order of the pooled documents while new assessments are made have the risk of introducing run bias, that is, underestimating the runs that provide fewer documents to the pool. We can observe that these dynamic methods do not suffer this effect more than the traditional depth- $k$  pooling in this respect. Moreover, we can see that these drop figures greatly improve while more relevant documents are found.

At this point, we can conclude that our methods are valuable for finding relevant documents as soon as possible and producing reliable assessments that do not harm the fairness of the collection. Now, as the final step of our evaluation, we study the reusability of the judgements produced by these adjudication strategies.



**Figure 2.** Distribution of Kendall's  $\tau$  between the ground-truth qrels and the qrels obtained with the corresponding adjudicating method when removing each team at a time from the runs: (a) TREC5 (30 relevants per topic) and (b) TREC8 (30 relevants per topic).

#### 5.4. Reusability

For evaluating the reusability, we performed a LOU experiment. In this experiment, the unique, relevant documents retrieved by each team are removed from the pool for building a new qrels file by applying each adjudicating method over each of those reduced pools. Then, we compute the correlation between the official ranking of runs and the ranking obtained with these reduced qrels. We include the results of this experiment in Table 5. This table depicts the average of Kendall's  $\tau$  correlation between ground-truth qrels – the official TREC judgements – and the qrels built with each adjudicating method when limiting the number of relevant documents per topic.

We can observe that all methods except DocID achieve strong correlations over 0.90 with only 50 relevant documents per topic in six out of eight collections (CT15 and CT16 are the exceptions). If we analyse these results along with the recall figures, we can argue that adjudication methods using RF are a well-performing alternative when gathering new assessments: from this table, we know they produce reusable judgements, and from above, we know that we can obtain them with less assessor effort.

The main objective of this reusability evaluation was to study if adjudication methods that perform some prioritisation may underestimate some runs and harm the quality of the assessments compared with the standard depth- $k$  method. From these results, we can conclude that this effect does not exist and that judgements built with these strategies are reusable.

To gain more insights into the reusability results we have just presented, we have plotted the distribution of Kendall's  $\tau$  values for TREC5 and TREC8 datasets for the case of 30 relevant documents per topic. The previous table includes the average of all the correlations obtained when leaving each participant team out of the pools. In Figure 2, we show a box-plot of those correlations.

In this figure, we can observe that adjudication methods that operate on a per-run basis (MTF, MM, MTF + DMM and MM + DMM) produce wider distributions than algorithms that operate on the whole pool (DocID and DMM). This suggests that having fewer relevant judgements per topic is what most affects the reusability of the judgements produced by the latter. We observe that removing one team at a time from the pool does not affect very much, and for this reason, the distributions are narrower. On the contrary, the other algorithms can obtain high correlations even when reducing the number of relevant documents per topic. However, removing one team may affect their performance since we see a larger distribution variability.

## 6. Conclusion and future work

In this article, we have explored the use of RF as adjudicating strategy to form pooled test collections. Under this approach, we tested different state-of-the-art methods for estimating a feedback model based on the assessors' judgements of the pooled documents. We proposed three adjudicating methods. The first one employs the assessors' judgements for reranking the entire pool, while the other two perform the reranking on a per-run basis. The best-performing statistical feedback model was DMM, which consistently outperformed the others that we evaluated: RM1, RM3 and MEDMM.

Overall, we observed that, in general, employing real RF for prioritising pooled documents improves over the existing algorithms. Experimental findings show that our proposed methods outperform the baselines in retrieving relevant documents with fewer judgements overall. We also showed that our strategies can keep the reliability, fairness and reusability of the judgements. For this reason, a method that retrieves relevant documents earlier and is also reliable, fair and reusable might be a better option to build a new test collection in situations where the budget is restricted.

This work paves the way for further investigation on RF for the prioritisation of documents in a pooling-based scenario. The good results achieved by employing popular RF models indicate that there may be room for improvement using other methods that explore different techniques. Thus, we plan to research other approaches here. In particular, we want to test the behaviour of RF with matrix factorisation and linear methods [53,54] in this experimental setup. We also plan to investigate a method based on reinforcement learning [55], as well as variations of relevance models [56,57]. Finally, in this scenario, there is explicit negative feedback (documents judged as non-relevant). Thus, we find interesting to study the use of the negative RF in the estimation of the models used for reranking [58].

### 6.1. Limitations

We recognise that our work has some limitations the reader should be aware of.

We have evaluated the fairness of our proposed methods in terms of ranking participant runs. However, prioritising the pooled documents may incur in some kind of judgement bias. That is, actively changing the order in which assessors see the documents may cognitively influence their decisions on the documents' relevance. We believe this issue is out of the scope of this work, and we will investigate it in the future.

We did not tune the parameters of the smoothing techniques used in the RF models' computation. This would entail a much bigger search space, with an increase in the computation time needed, something we could not afford. Tuning these parameters may, or may not, lead to better performance of the proposed methods.

The statistical feedback models studied here work on a bag-of-words fashion, disregarding the semantics and word order. We acknowledge that this might be a limitation of these models, and we plan to study in the future other kind of methods, such as transformer-based ones that are able to capture latent features from the text.

### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.


### Funding

The author(s) disclosed receipt of the following financial support for the research, authorship and/or publication of this article: this work has received support from: (1) project PLEC2021-007662 (grant no. MCIN/AEI/10.13039/501100011033, Ministerio de Ciencia e Innovación (MCIN), Agencia Estatal de Investigación, Plan de Recuperación, Transformación y Resiliencia, Unión Europea-NextGenerationEU), (2) Programa de Ayudas para la Formación de Profesorado Universitario, grant number FPU20/02659 (Ministerio de Universidades), (3) project PID2022-137061OB-C21 (Proyectos de Generación de Conocimiento, MCIN), (4) Consellería de Educación, Universidade e Formación Profesional (accreditation 2019-2022 ED431G 2019/01) and the European Regional Development Fund, which acknowledges the Centro de Investigación en Tecnologías de la Información y la Comunicación (CITIC) Research Centre in Information and Communications Technology (ICT) of the University of A Coruña as a Research Centre of the Galician University System and (5) project ED431-B 2022/33 (Xunta de Galicia/European Regional Development Fund (ERDF)).

### Notes

1. National Institute of Standards and Technology Text REtrieval Conference (<https://trec.nist.gov>).
2. National Institute of Informatics testbeds and community for information access research (<https://research.nii.ac.jp/ntcir/index-en.html>).
3. Conference and laboratories of the evaluation forum (<https://clef2022.clef-initiative.eu>).
4. Each topic includes a title, description and narrative sections that describe the topic with different level of detail.
5. For all experiments reported here, we sorted the documents from each run by score.
6. According to the overviews of these tracks, the qrels include some documents sampled outside the top-*k* pool. Since reproducing this would be very hard, we just use the top-*k* pools.

### ORCID iD

David Otero  <https://orcid.org/0000-0003-1139-0449>

## References

- [1] Sanderson M. Test collection based evaluation of information retrieval systems. *Found Trend Inform Retrieval* 2010; 4(4): 247–375.
- [2] Voorhees EM and Harman DK. *TREC: experiment and evaluation in information retrieval*. Cambridge, MA: The MIT Press, 2005.
- [3] Harman D. Information retrieval evaluation. *Synth Lect Inform Concept Retriev Serv* 2011; 3(2): 1–119.
- [4] Voorhees EM. The philosophy of information retrieval evaluation. In: Peters C, Braschler M, Gonzalo J et al. (eds) *Evaluation of cross-language information retrieval systems: second workshop of the cross-language evaluation forum (CLEF '01)*. Berlin: Springer, 2001, pp. 355–370.
- [5] Sanderson M and Zobel J. Information retrieval system evaluation: effort, sensitivity, and reliability. In: *Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval SIGIR '05*, Salvador, Brazil, 15–19 August 2005, pp. 162–169. New York: Association for Computing Machinery.
- [6] Cormack GV, Palmer CR and Clarke CLA. Efficient construction of large test collections. In: *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval. SIGIR '98* (eds WB Croft, A Moffat, CJ Van Rijsbergen et al.), Melbourne, VIC, Australia, 24–28 August 1998, pp. 282–289. New York: Association for Computing Machinery.
- [7] Losada DE, Parapar J and Barreiro A. Multi-armed bandits for adjudicating documents in pooling-based evaluation of information retrieval systems. *Inform Proces Manag* 2017; 53(5): 1005–1025.
- [8] Moffat A, Webber W and Zobel J. Strategic system comparisons via targeted relevance judgments. In: *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval. SIGIR '07* (eds W Kraaij, AP De Vries, CLA Clarke et al.), Amsterdam, 23–27 July 2007, pp. 375–382. New York: Association for Computing Machinery.
- [9] Aslam JA, Pavlu V and Savell R. A unified model for metasearch, pooling, and system evaluation. In: *Proceedings of the 12th ACM international conference on information and knowledge management. CIKM '03*, New Orleans, LA, 3–8 November 2003, pp. 484–491. New York: Association for Computing Machinery.
- [10] Sanderson M and Joho H. Forming test collections with no system pooling. In: *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval SIGIR '04*, Sheffield, 25–29 July 2004, pp. 33–40. New York: Association for Computing Machinery.
- [11] Robertson S. On the history of evaluation in IR. *J Inform Sci* 2008; 34(4): 439–456.
- [12] Spärck Jones K and van Rijsbergen CJ. *Report on the need for and provision of an 'ideal' information retrieval test collection*. Cambridge: University Computer Laboratory, 1975.
- [13] Zobel J. How reliable are the results of large-scale information retrieval experiments? In: *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval SIGIR '98*, Melbourne, VIC, Australia, 24–28 August 1998, pp. 307–314. New York: Association for Computing Machinery.
- [14] Cormack GV and Lynam TR. Power and bias of subset pooling strategies. In: *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval. SIGIR '07*, Amsterdam, 23–27 July 2007, pp. 837–838. New York: Association for Computing Machinery.
- [15] Lu X, Moffat A and Culpepper JS. The effect of pooling and evaluation depth on IR metrics. *Inform Retriev J* 2016; 19(4): 416–445.
- [16] Losada DE, Parapar J and Barreiro A. Feeling lucky? Multi-armed bandits for ordering judgements in pooling-based evaluation. In: *Proceedings of the 31st annual ACM symposium on applied computing SAC '16*, Pisa, 4–8 April 2016, pp. 1027–1034. New York: Association for Computing Machinery.
- [17] Li D and Kanoulas E. Active sampling for large-scale information retrieval evaluation. In: *Proceedings of the 2017 ACM conference on information and knowledge management CIKM '17* (eds E Lim, M Winslett, M Sanderson et al.), Singapore, 6–10 November 2017, pp. 49–58. New York: Association for Computing Machinery.
- [18] Rahman MM, Kutlu M, Elsayed T et al. Efficient test collection construction via active learning. In: *Proceedings of the 2020 ACM SIGIR on international conference on theory of information retrieval. ICTIR '20*, Oslo, 14–17 September 2020.
- [19] Rahman MM, Kutlu M and Lease M. Constructing test collections using multi-armed bandits and active learning. In: *The World Wide Web conference WWW '19*, San Francisco, CA, 13–17 May 2019, pp. 3158–3164. New York: Association for Computing Machinery.
- [20] Lipani A, Losada DE, Zuccon G et al. Fixed-cost pooling strategies. *IEEE Trans Knowl Data Eng* 2019; 33: 1503–1522.
- [21] Allan J, Harman DK, Kanoulas E et al. TREC 2017 Common core track overview. In: *Proceedings of the twenty-sixth text REtrieval conference. TREC '17*, Gaithersburg, MD, pp. 500–324, <https://trec.nist.gov/pubs/trec26/papers/Overview-CC.pdf>
- [22] Voorhees EM. On building fair and reusable test collections using bandit techniques. In: *Proceedings of the 27th ACM international conference on information and knowledge management CIKM '18*, Torino, 22–26 October 2018, pp. 407–416. New York: Association for Computing Machinery.
- [23] Craswell N, Mitra B, Yilmaz E et al. TREC deep learning track: reusable test collections in the large data regime. In: *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval SIGIR '21*, 11–15 July 2021, pp. 2369–2375. New York: Association for Computing Machinery. DOI: 10.1145/3404835.3463249.



- [24] Voorhees EM, Craswell N and Lin J. Too many relevants, whither Cranfield test collections? In: *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval SIGIR '22*, Madrid, 11–15 July 2022.
- [25] Craswell N, Mitra B, Yilmaz E et al. Overview of the TREC 2019 deep learning track. In: *Proceedings of the twenty-eight text REtrieval conference*, Gaithersburg, MD, 13–15 November 2019, <https://trec.nist.gov/pubs/trec28/papers/OVERVIEW.DL.pdf>
- [26] Craswell N, Mitra B, Yilmaz E et al. Overview of the TREC 2020 deep learning track. In: *Proceedings of the twenty-ninth text REtrieval conference*, Gaithersburg, MD, 16–20 November 2020, <https://trec.nist.gov/pubs/trec29/papers/OVERVIEW.DL.pdf>
- [27] Craswell N, Mitra B, Yilmaz E et al. Overview of the TREC 2021 deep learning track. In: *Proceedings of the thirtieth text REtrieval conference*, Gaithersburg, MD, 15–19 November 2021, <https://trec.nist.gov/pubs/trec30/papers/Overview-DL.pdf>
- [28] Otero D, Parapar J and Barreiro A. The wisdom of the rankers: a cost-effective method for building pooled test collections without participant systems. In: *Proceedings of the 36th annual ACM symposium on applied computing SAC '21*, 22–26 March 2021, pp. 672–680. New York: Association for Computing Machinery. DOI: 10.1145/3412841.3441947.
- [29] Soboroff I, Nicholas C and Cahan P. Ranking retrieval systems without relevance judgments. In: *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval SIGIR '01*, New Orleans, LA, 9–13 September 2001, pp. 66–73. New York: Association for Computing Machinery.
- [30] Sakai T and Lin CY. Ranking retrieval systems without relevance assessments: revisited. In: *Proceedings of the 3rd international workshop on evaluating information access EVIA '10*, National Institute of Informatics (NII), pp. 25–33, <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings8/EVIA/05-EVIA2010-SakaiT.pdf>
- [31] Roitero K, Brunello A, Serra G et al. Effectiveness evaluation without human relevance judgments: a systematic analysis of existing methods and of their combinations. *Inform Proces Manag* 2020; 57(2): 102149.
- [32] Cormack GV and Grossman MR. Beyond pooling. In: *Proceedings of ACM SIGIR*, Ann Arbor, MI, 8–12 July 2018, pp. 1169–1172. New York: ACM.
- [33] Roitero K, Passon M, Serra G et al. Reproduce. Generalize. Extend. On information retrieval evaluation without relevance judgments. *J Data Inform Qual* 2018; 10(3): 11.
- [34] Jayasinghe GK, Webber W, Sanderson M et al. Improving test collection pools with machine learning. In: *Proceedings of the 2014 Australasian document computing symposium ADCS '14*, Melbourne, VIC, Australia, 27–28 November 2014, pp. 2–9. New York: Association for Computing Machinery.
- [35] Jayasinghe GK, Webber W, Sanderson M et al. Extending test collection pools without manual runs. In: *Proceedings of the 37th international ACM SIGIR conference on research and development in information retrieval SIGIR '14*, Gold Coast, QLD, Australia, 6–11 July 2014, pp. 915–918. New York: Association for Computing Machinery.
- [36] Moffat A, Scholer F, Thomas P et al. Pooled evaluation over query variations: users are as diverse as systems. In: *Proceedings of the 24th ACM international on conference on information and knowledge management CIKM '15*, Melbourne, VIC, Australia, 18–23 October 2015, pp. 1759–1762. New York: Association for Computing Machinery.
- [37] Lv Y and Zhai C. A comparative study of methods for estimating query language models with pseudo feedback. In: *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval SIGIR '01*, New Orleans, LA, 9–13 September 2001, pp. 1895–1898. New York: Association for Computing Machinery.
- [38] Lafferty J and Zhai C. Document language models, query models, and risk minimization for information retrieval. In: *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval SIGIR '01*, New Orleans, LA, 9–13 September 2001, pp. 111–119. New York: Association for Computing Machinery.
- [39] Lavrenko V and Croft WB. Relevance based language models. In: *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval SIGIR '01*, New Orleans, LA, 9–13 September 2001, pp. 120–127. New York: Association for Computing Machinery.
- [40] Abdul-Jaleel N, Allan J, Croft WB et al. UMass at TREC 2004: novelty and hard. In: *Proceedings of TREC 2004*, Gaithersburg, MD, 16–19 November 2004, pp. 1–13, <https://trec.nist.gov/pubs/trec13/papers/umass.novelty.hard.pdf>
- [41] Zhai C and Lafferty J. Model-based feedback in the language modeling approach to information retrieval. In: *Proceedings of the tenth international conference on information and knowledge management CIKM '01*, Atlanta, GA, 5–10 October 2001, pp. 403–410. New York: Association for Computing Machinery.
- [42] Lv Y and Zhai C. Revisiting the divergence minimization feedback model. In: *Proceedings of the 23rd ACM international conference on information and knowledge management CIKM '14*, Shanghai, China, 3–7 November 2014, pp. 1863–1866. New York: Association for Computing Machinery.
- [43] Hazimeh H and Zhai C. Axiomatic analysis of smoothing methods in language models for pseudo-relevance feedback. In: *Proceedings of the 2015 international conference on the theory of information retrieval ICTIR '15*, Northampton, MA, 27–30 September 2015, pp. 141–150. New York: Association for Computing Machinery.
- [44] Voorhees EM, Soboroff I and Lin J. Can old TREC collections reliably evaluate modern neural retrieval models? *arXiv*, 2022, <https://arxiv.org/abs/2201.11086>
- [45] Altun B and Kutlu M. Building test collections using bandit techniques: a reproducibility study. In: *Proceedings of the 29th ACM international conference on information and knowledge management CIKM '20*, 19–23 October 2020, pp. 1953–1956. New York: Association for Computing Machinery. DOI: 10.1145/33405313412121.
- [46] George Kendall M. *Rank correlation methods*. London: Charles Griffin, 1948.
- [47] Kendall MG. A new measure of rank correlation. *Biometrika* 1938; 30(1–2): 81–93.

- [48] Yilmaz E, Aslam JA and Robertson S. A new rank correlation coefficient for information retrieval. In: *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval SIGIR '08*, Singapore, 20–24 July 2008, pp. 587–594. New York: Association for Computing Machinery.
- [49] Sakai T. *Laboratory experiments in information retrieval* (The information retrieval series, vol. 40). New York: Springer, 2018.
- [50] Carterette B. Multiple testing in statistical analysis of systems-based information retrieval experiments. *ACM Trans Inform Syst* 2012; 30(1): 4.
- [51] Fuhr N. Some common mistakes in IR evaluation, and how they can be avoided. *SIGIR Forum* 2018; 51(3): 32–41.
- [52] Sakai T. On Fuhr's guideline for IR evaluation. *SIGIR Forum* 2021; 54(1): 1–8.
- [53] Valcarce D, Parapar J and Barreiro A. LiMe: linear methods for pseudo-relevance feedback. In: *Proceedings of the 33rd annual ACM symposium on applied computing SAC '18*, Pau, 9–13 April 2018, pp. 678–687. New York: Association for Computing Machinery.
- [54] Valcarce D, Parapar J and Barreiro A. Document-based and term-based linear methods for pseudo-relevance feedback. *SIGAPP Appl Comput Rev* 2018; 18(4): 5–17.
- [55] Montazerlghaem A, Zamani H and Allan J. A reinforcement learning framework for relevance feedback. In: *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval SIGIR '20*, 25–30 July 2020, pp. 59–68. New York: Association for Computing Machinery. DOI: 10.1145/33972713401099.
- [56] Roy D, Bhatia S and Mitra M. Selecting discriminative terms for relevance model. In: *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval SIGIR'19*, Paris, 21–25 July 2019, pp. 1253–1256. New York: Association for Computing Machinery.
- [57] Parapar J and Barreiro A. Promoting divergent terms in the estimation of relevance models. In: *Proceedings of the third international conference on advances in information retrieval theory ICTIR'11*, Bertinoro, 12–14 September 2011.
- [58] Wang X, Fang H and Zhai C. A study of methods for negative relevance feedback. In: *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval SIGIR '08*, Singapore, 20–24 July 2008, pp. 219–226. New York: Association for Computing Machinery.