Information Retrieval manuscript No.
(will be inserted by the editor)

Pooling-Based Continuous Evaluation of Information Retrieval Systems

**Alberto Tonon · Gianluca Demartini ·
Philippe Cudré-Mauroux**

Received: date / Accepted: date

Abstract The dominant approach to evaluate the effectiveness of Information Retrieval (IR) systems is by means of reusable test collections built following the Cranfield paradigm. In this paper, we propose a new IR evaluation methodology based on pooled test-collections and on the continuous use of either crowdsourcing or professional editors to obtain relevance judgements. Instead of building a static collection for a finite set of systems known a priori, we propose an IR evaluation paradigm where retrieval approaches are evaluated iteratively on the same collection. Each new retrieval technique takes care of obtaining its missing relevance judgements and hence contributes to augmenting the overall set of relevance judgements of the collection. We also propose two metrics: Fairness Score, and Opportunistic Number of Relevant Documents, which we then use to define new pooling strategies. The goal of this work is to study the behavior of standard IR metrics, IR system ranking, and of several pooling techniques in a continuous evaluation context by comparing continuous and non-continuous evaluation results on classic test collections. We both use standard and crowdsourced relevance judgements, and we actually run a continuous evaluation campaign over several existing IR systems.

Alberto Tonon
University of Fribourg
Tel.: +41 26 300 84 76
Fax: +41 26 300 97 26
E-mail: alberto@exascale.info

Gianluca Demartini
University of Sheffield
Tel.: +44 114 22 22637
Fax: +44 114 22 22631
E-mail: g.demartini@sheffield.ac.uk

Philippe Cudré-Mauroux
University of Fribourg
Tel.: +41 26 300 83 32
Fax: +41 26 300 97 26
E-mail: phil@exascale.info

1 Introduction

Evaluating the effectiveness of IR systems (IRSs) has been a focus of IR research for decades. Historically, the Cranfield paradigm [13] defined the standard methodology with which IRSs are evaluated by means of reusable test collections.

Over the past 20 years, the Text REtrieval Conference (TREC) has created standard and reusable test collections for different search tasks by refining and improving the original evaluation strategies first proposed by Cleverdon [13] and later refined by Salton [23]. A standard IR evaluation collection is composed of:

1. a fixed document collection;
2. a set of topics (from which keyword queries are created);
3. a set of relevance judgements defining the relevance of the documents with respect to the topics according to human assessors;
4. ranked results (called 'runs') for all topics and for all participating IRSs.

One of the pressing issues encountered by TREC and by commercial search engines over the years is the rapid growth of the document collections. Very large document collections are essential for assessing the scalability of the systems being evaluated, yet make it impractical to obtain relevance judgements for all the documents in the collection. This led to the idea of *pooling* [19], that is, judging only the top documents retrieved by the set of IRSs being evaluated, and assuming that the rest of the results are non-relevant (which might not be the case in reality).

Recently, evaluation measures dealing with incomplete judgements have been proposed [9,39,40,3,11]. Some of those only consider judged documents (e.g., bpref [9]), while others attempt to estimate the values by randomly sampling the ranked list of retrieved documents in order to select the documents to be judged (e.g., infAP [39]). Anyway, it is worth noticing that using a wide range of metrics based on weighted precision, an "unknown score" (that is, the contribution that a non judged document would give to the measurement) can be accumulated and reported as part of the evaluation. Another issue of the pooling strategy is that, while the initial systems contributing to the pool are fairly compared, IRSs being evaluated on the same test collection afterwards are disadvantaged, as it may happen that some of their top results should be treated as relevant but are actually considered as non-relevant if they were left unjudged after the initial evaluation campaign. As the document collection grows, it is more likely that an IRS which did not participate in the pool construction might retrieve documents that were not judged, making it impossible to accurately measure system effectiveness. Such bias has already been highlighted by previous work (e.g., [38,3]).

Lately, crowdsourcing has been suggested as a practical means of building large IR evaluation collections using the Web. Instead of employing trained human assessors, micro-tasks are created on online crowdsourcing platforms with the goal of collecting relevance judgements from the *crowd* (i.e., untrained Web users willing to perform simple tasks on-line). One example of this recent trend is the SemSearch initiative[1], which uses crowdsourcing techniques to produce relevance judgements by granting a small economic reward to anonymous Web users who judge the relevance of semi-structured entities [16]. Moreover, the TREC Crowdsourcing track has studied how to best obtain and aggregate relevance judgements from the crowd [33].

Such a novel approach to relevance judgements triggers obvious questions about the reliability of the results. Previous work [6, 7] experimentally showed how such an approach is reliable and, most importantly, repeatable.

This result opens the doors to new methodologies for IR evaluation where the crowd is exploited in a *pay-as-you-go* manner. That is, as a new search strategy or ranking feature is developed, its evaluation can trigger the update of existing test collections, which i) ensures a fair comparison of the new system against previous baselines and ii) provides the research community with an improved collection (i.e., more complete relevance judgements).

In this paper, we propose a new evaluation methodology, which iteratively and continuously updates the evaluation collections by means of either crowdsourcing or professional assessors as new IRSs appear and get evaluated. We claim that crowdsourcing relevance judgements is helpful to run continuous evaluation of IRSs: It would be infeasible to involve TREC-style assessors each time a new IRS is developed or a new variant of a ranking function needs to be tested and compared to previous approaches. Thus, thanks to continuously available crowd workers, it is possible to create and maintain continuous evaluation collections in an efficient and scalable way.

We introduce in the following our novel evaluation methodology as well as a set of metrics to monitor the evolution of IR system rankings as the evaluation progresses. We also experimentally evaluate the feasibility of the proposed approach for different settings, ranging from a continuous evaluation based on a collection built according to the Cranfield paradigm (i.e., a TREC collection) to the use of crowdsourcing to create continuous evaluation collections from the ground up. Given an existing test collection and new IRS runs, our approach crowdsources the relevance judgements of selected results (i.e., those which would have contributed to the pool) and updates the evaluation metrics for all runs belonging to the evaluation collection proposing an updated ranking of approaches. We also propose a novel result selection approach for pool construction based on the number of documents judged for each evaluated run.

In summary, the main contributions of this paper are:

- A novel continuous evaluation methodology based on the adoption of crowdsourcing in a *pay-as-you-go* fashion to update existing IR evaluation collections.
- New measures and metrics to compare and to assess the stability of the IRSs ranking during the evaluation as well as the fairness and bias of the evaluation over different IRSs.
- New techniques to align the results of heterogeneous *crowds* in a continuous evaluation setting.

[1] http://km.aifb.kit.edu/ws/semsearch10/ and http://km.aifb.kit.edu/ws/semsearch11/

- An extensive experimental evaluation of our techniques over several standard document collections as well as the resources and relevance judgements produced for the sake of reproducibility.
- A set of tools to support continuous IR evaluations, which can easily be integrated with TREC tools such as trec_eval.

The rest of the paper is structured as follows. We start by reviewing related work below. We present our approach to continuous IR evaluation in Section 3. We experimentally evaluate the feasibility of our approach in Section 4, by comparing the results of a continuous evaluation campaign against standard static collections and by running a continuous evaluation of IRSs in which, for each system, crowdsourcing is used to obtain missing relevance judgements. In Section 5 we discuss the key points to consider when running a continuous evaluation campaign and, finally, Section 6 concludes the paper and presents our future work.

2 Related Work

IR evaluation is a well-studied research topic. In this section, we briefly review the efforts in this area that are most relevant to the novel continuous evaluation paradigm we propose in this work.

Reusability of IR Test Collections: Early research efforts studied the evaluation bias caused by incomplete relevance judgements and thus pointed out the limited reusability of test collections. The most relevant work in that context is [41], where Zobel studied the bias introduced by pooling in evaluating IRSs. While he concluded that available evaluation collections are still viable, he experimentally illustrated the drawback of pooling by estimating the number of relevant results in the entire collection beyond those actually observed by the assessors. Later work by Büttcher *et al.* [10] presented more alarming results. Analyzing larger collections than those studied by Zobel, the authors found that the IRSs rankings were fluctuating when one considered or discarded certain IRSs contributing to the pool. Buckley *et al.* also observed that runs not participating to the pool are unfairly evaluated [8]. Further research work has looked at how different judgements can modify the outcome of the evaluation. Voorhees [37] measured the correlation of IRSs rankings using different sets of relevance judgement. Results show that test collections are reliable since high ranking correlations were observed.

Pooling Strategies: After analyzing the bias introduced by pooling, some researchers focused on the definition of novel evaluation metrics that could cope with incomplete relevance judgements. Most of those metrics replace the standard fix-depth pooling method and evaluate the results by using sampling strategies to select the documents to be judged (e.g., infAP [39]). Another example is xinfAP [40], which, instead, considers stratified sampling to create the judgement set in order to use a higher sampling rate for highly ranked results. Compared to such a family of metrics, our approach is orthogonal and can be applied to any pooling strategy: in our work we experiment with fix-depth, random sampling [3], and selective pooling [11] strategies.

The problem of evaluating new runs after the judgement pool has been constructed was also studied [38]. The authors propose to measure the bias of new IRSs based on the unjudged documents they retrieve. Then, they use the measured bias to adjust the evaluation score of new IRSs which did not participated in the pool. As compared with it, we instead propose to extend the existing pool with new unjudged documents the new IRS has retrieved.

Other alternatives to fix-depth pooling (i.e., judging the the top n results from each run) have been proposed. Aslam *et al.* [3,4] proposed a pooling method based on non-uniform random sampling. The documents to be judged are selected at random following a non-uniform distribution defined over different strata having a different sampling probability. Carterette *et al.* [11] propose an iterative process where the pool is constructed selecting the next document to be judged after each relevance judgement. In that case, the best document for relevance judgement can be selected based on its expected probability of relevance. In our paper, we compare fix-depth pooling against the approaches proposed in the context of continuous IR evaluation [3,4,11]. As we show in Section 3, our new continuous evaluation techniques can be applied to such settings as well. An approach to create test collections that aims to measure its future reusability was proposed by Carterette *et al.* [12]. Instead, we propose to update existing test collections over time by increasing collection quality and reliability and compare different pool construction strategies in such settings.

An alternative to judging the relevance of documents has been proposed by Pavlu *et al.* [30] who suggest to judge relevant nuggets of information instead and to match them to retrieved documents in order to automatically generate relevance judgements for documents. While the goal of obtaining scalable and reusable test collections is the same as ours, we instead propose to keep people judging documents instead of inferring their relevance based on imperfect text matching algorithms.

Crowdsourcing Relevance Judgements: Some of the most recent research efforts in the field of IR evaluation focus on the use of *crowdsourcing* to replace trained assessors and create relevance judgements. One relevant piece of work in that context is the study of repeatability of crowdsourced IR evaluation by Blanco *et al.* [6,7]. Their findings show that, by repeating the crowdsourced judgements over time, the evaluation measures may vary, though the IRSs ranking is somewhat stable. These results open the door to continuous IR evaluations such as the methodology we present in this paper.

Other IR evaluation collections based on crowdsourcing have been created and studied. Crowdsourcing has been successfully adopted to create relevance judgements for the TREC 2010 Blog track [25]. In [21,22], Kazai *et al.* studied how the crowdsourcing tasks (HITs) should be designed, and suggested quality control techniques in the context of book search evaluations. In [1,20], the authors studied how to design the HITs for IR evaluations and observed that the crowd can be as precise as TREC assessors. Hosseini *et al.* [17] proposed a technique that takes into account each worker's accuracy to weight each answer differently. While previous work in this area aims at identifying reliable workers, in our work, we develop strategies to balance the *assessment diversity* of the crowd creating judgements for one IRS as compared to previous ones by aligning different crowds participating in different evaluation steps. One of the latest work in this area is [2], where Alonso and Mizzaro crowdsourced some of the relevance judgements from TREC-7 and compared them against the original judgements, finding that crowdsourcing is a valid alternative to trained relevance assessors. Moreover, the TREC Crowdsourcing track [33] has studied how to best obtain and consolidate crowd answers to obtain relevance judgements. In this paper, we systematically analyze the feasibility and the stability of crowdsourced relevance judgements for a continuous IR evaluation campaign.

Finally, Scholer *et al.* analyze the effect of threshold priming, that is, how people's relevance perception changes when seeing varying degrees of relevant documents [32]. They show that people exposed to only non-relevant documents tend to be more generous when deciding about relevance than people exposed to higher relevant documents. While the authors did not experiment with anonymous Web users but rather with university employees, we believe their results are applicable to online crowdsourcing platforms; however, addressing this effect is out of the scope of this paper.

3 Continuous IRS Evaluation

In this section we describe the evaluation methodology we propose. We start by highlighting the limitations of current IR evaluations, and by formally introducing our methodology; we then continue by discussing each part of it in a separate subsection. In particular: we describe a new set of statistics one can use to have a preliminary evaluation of its system; we discuss existing strategies to select the documents to judge for relevance, and we propose two novel approaches for this task; finally, discuss how to obtain and integrate relevance judgements using different sets of assessors.

3.1 Limitations of Current IR Evaluations

Two problems often surface when applying current evaluation methodologies to large-scale evaluations of IRSs:

- The difficulty in gathering comprehensive relevance judgements for long runs.
- The unfair bias towards systems that are evaluated as part of the original evaluation campaign (i.e., when the collection is created).

Both issues relate to the potential lack of information pertaining to the relevance of documents from the collection. First, as the document collection grows (current collections, such as *ClueWeb12*[2], contain around one billion pages) the problem of unjudged documents becomes more evident. Often, a significant fraction of the relevant results are not retrieved by IRSs participating in an evaluation initiative and, thus, are never going to be judged. As a result, an increasing number of relevant documents do not appear in the judgement pool (this was already observed by previous research on document collections that were much smaller than the ones the IR research community

[2] http://lemurproject.org/clueweb12/

currently uses [41]). While recent efforts [3,4,11] have addressed such issues by sampling ranked documents, relevance judgements are still incomplete for large collections. This motivates the need for new evaluation strategies that take into account or try to compensate for this shortcoming.

The second aspect highlights the bias of evaluating an IRS participating to the pool versus another system being evaluated afterwards [38]. While the early IRSs will have, by definition of fix-depth pooling, their top retrieved documents judged, the later IRS may have a significant number of its top documents unjudged (this occurs whenever one of its retrieved documents was not retrieved by the original IRSs that participated in the pool). Hence it penalizes later approaches that might actually be more effective than the early ones but retrieve very different sets of results. This motivates the need for a different evaluation methodology that provides a fairer comparison of IRSs not participating in the original pool.

Next, we give an overview of our proposed methodology, introduce a series of instruments to help participants of the campaign assessing the evaluation reliability of their ranking algorithm, and discuss existing pooling strategies in a continuous setting. Finally, we propose novel pooling approaches and present methods to obtain additional judgements and integrate them with existing ones.

3.2 Assumptions

In this section we report the assumptions under which the evaluation methodology we propose works.

First of all, the goal of a continuous evaluation campaign is to compare information retrieval systems based on relevance judgements made by humans. The methodology we propose does not require any user study nor any direct contact with the final users of the system. In particular, in this work we do not consider methods like A/B testing, in which the behavior of users using different systems is analyzed. The productivity of the users is estimated by standard IR metrics taking into account relevant and non-relevant documents as it is common practice in Information Retrieval research [24]. Such an evaluation does not reflect the satisfaction of the users as well as a user study does but allows to reliably compare several different systems with less effort. It has already been proven that crowdsourcing can be used to run user studies [29]. This suggests that it could be feasible to run a continuous evaluation featuring user studies, but this is not the focus of this paper.

Another assumption we make is that re-judging the top documents retrieved by all systems of interest is out of the question as this would mean imposing a higher “price” in order for a system to join a continuous evaluation campaign. Moreover, in order to facilitate the organization of the campaign and to foster participation, we assume that the corpus and the topics composing the dataset used during the evaluation cannot change over time. On one hand, this allows the participants to submit one run of their method over the fixed dataset instead of either releasing their system to the other participants of the campaign or providing and maintaining an endpoint to it; on the other hand, the organizers and the participants of the campaign can easily and rapidly update the scores of all the IRSs. This assumption implies that it is not possible to extend the evaluation by including new topics (the older system cannot be evaluated on them) thus, contrary to what happens in TREC, *all the participants* know the topics used for the evaluation. Nevertheless, notice that this also applies to all researchers using one of the past TREC datasets in order to evaluate their systems.

Another aspect we do not tackle here is multi-graded relevance as it is harder to integrate multi-graded relevance judgements made by different judges in different points of times (see Section 3.6 for more details on the integration of relevance judgements). Nevertheless, the results obtained by Blanco et al. [6] suggest that our evaluation methodology can be extended to multi-graded relevance and, consequently, to metrics based on it. The price to pay for this choice is less expressive (and thus less precise) relevance judgements.

Finally, as we describe in Section 3.6.1, we assume that judges can be characterized by their strictness/leniency; we thus do not take into account the case in which the dataset contains idiosyncratic queries or documents the judge cannot understand. Such cases can be handled, for example, by exploiting a push-crowdsourcing methodology [15], that is, pushing tasks not to anonymous Internet users but rather to people of whom we know (part of) the background.

3.3 Organizing a Continuous IR Evaluation

We now describe how to organize and run a *Continuous Evaluation Campaign*, a novel IR evaluation methodology based on the continuous collection of new relevance judgements. Such a methodology addresses the two limitations described above by creating and using evaluation collections that are not static since their sets of relevance judgements get updated with each new IRS being evaluated.

Formally, a continuous evaluation campaign (or just campaign, for short) is characterized by:

1. a fixed document collection $\mathcal{D}$;
2. a fixed set of topics $\mathcal{T}$;
3. a set of relevance judgements $\mathcal{J}$, whose size increases over time;
4. a set of runs $\mathcal{R}$ participating to the campaign, whose size increases over time.

We note that, while the first two components are fixed, the third and the fourth ones vary over time. The idea behind this is that a new system can join the campaign at any time but, in order to do that, its creators have to provide the missing relevance judgements needed to evaluate it, and to make both the evaluated runs of the system and the new relevance judgements available to future participants (thus, increasing the cardinality of $\mathcal{J}$ and $\mathcal{R}$). When the i-th system joins the continuous evaluation campaign we say that we are at the *i-th step* of the campaign. Figure 1 illustrates the difference between a traditional evaluation methodology (top), in which the systems are evaluated

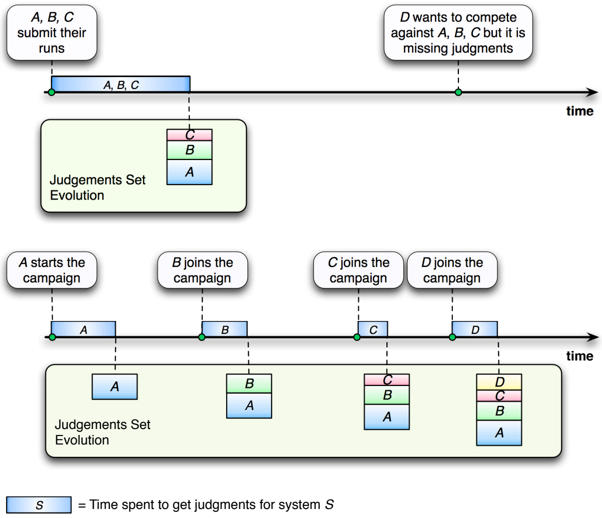

Fig. 1: Traditional evaluation methodology (top) vs Continuous evaluation campaign (bottom).

all at the same time, and a continuous evaluation campaign (bottom). The picture shows that, in the first case, relevance judgements for systems A, B, and C are all produced at the same time, while judgements for system D, which did not participate in the evaluation, will never be done. On the contrary, when a system joins a continuous evaluation campaign, it enriches the ground truth by providing additional relevance judgements. Notice that the set of documents judged after C arrived is the same in both cases, however, in the latter it is extended when D joins the campaign.

Every step of a continuous evaluation campaign is composed of four stages, namely, *document selection*, *relevance judgements collection*, *relevance judgements integration*, and *run evaluation*. During the document selection stage, the new documents that have to be judged are chosen and, in the next stage, relevance judgements are obtained by means of crowdsourcing or with professional assessors. The new judgements are then integrated into those from the previous steps of the campaign in the relevance judgements integration stage and, finally, the current system run is evaluated and the scores of all the other runs participating to the campaign are updated (or recomputed) to take into account the new judgements. This is necessary since metrics computed by using different sets of judgements are typically not comparable. In the rest of this paper we use the notation $m|_i$ indicating the value of the evaluation metrics m computed by using the judgement set of the i-th step of the continuous evaluation campaign. We also use $\mathcal{J}|_i$, $\mathcal{R}|_i$ to denote, respectively, the set of judgements and the set of runs of the i-th step of the campaign. It is up to the organizers of the continuous evaluation campaign to select the evaluation metrics used to evaluate the runs and the algorithms to use for pooling (i.e., document selection) and for relevance judgements integration.

3.4 Continuous IR Evaluation Gauges

As already discussed, people can submit their system to a continuous evaluation campaign at any time and, most important, they are allowed to use the data shared by the participants (runs and relevance judgments). The latter point allows for assessing if it is worth or not joining submitting a new system to the campaign and, in particular, if it has any chance to outperform the current best method. In this section, we introduce several instruments that are useful to continuous evaluation participants to understand the cost/benefit tradeoff of joining a continuous evaluation campaign and, thus, obtaining additional relevance judgements for a new run. Such measures can also be used in order to set the minimum requirements for new runs joining the evaluation.

3.4.1 Measuring the Fairness of the judgement Pool

Throughout the continuous evaluation, the runs will have a varying number of judged documents.[3] Given that unjudged documents are assumed to be non-relevant (or have a relevance probability or score lower than 1), runs having a higher number of judged (and retrieved) documents are typically advantaged since their evaluation results are closer to those performed in the ideal setting of complete relevance judgements.

For that reason, we introduce a measure called *Fairness Score* (FS), which is defined similarly to Average Precision (AP) but focuses on the judgements rather than on the relevance of the results:

$$FS(run) = \frac{\sum_{k=1}^{n} JudCov(k) \cdot J(k)}{n}, \tag{1}$$

where $run = [1, \ldots, n]$ is a ranked list of retrieved results, $JudCov(k)$ is the proportion of documents judged among the top-k retrieved by run, and $J(k)$ equals to 1 if the document retrieved at rank k has been judged and 0 otherwise. The Fairness Score, similarly to AP, puts more weight on the judged documents that appear higher in the ranking. This follows from the intuition that the end user of an IRS is most likely to take into consideration the top results that are displayed, so, having top documents judged is more critical than having documents also at lower ranks. Moreover, since this intuition is followed by many well-known metrics built to evaluate ranked lists of results (AP, DCG [18], RBP [28], etc.), having a high-ranked judged document can make the difference when comparing two IRSs, as information on the relevance of a highly-ranked document may contribute more in increasing (or, possibly, decreasing) the score of the metrics used during the evaluation.

The Fairness Score of a given run equals to 1 if all its retrieved documents have been judged and to 0 if not a single retrieved document has been judged. However, notice that in some cases it does not make sense to consider the whole list of retrieved results. For instance, when computing the fairness score related to Precision10, one is interested only in the top-10 retrieved documents. In those cases we assume that the input run is a ranked list composed of only

[3] A similar situation already occurs for classic block evaluation initiatives such as TREC.

the documents influencing the value of the considered metrics. Leveraging on this definition, we can compute the Fairness Score of each run of a test collection in order to check whether the runs were all treated on a fair basis (see Section 4.2.2). The Fairness Score can be also used to asses the potential of a run before submitting it to a continuous evaluation campaign: if its Fairness Score is significantly lower than those of the other participants then many of its top-ranked documents are unjudged, thus, its ranking (computed without obtaining new relevance judgments) has chances to increase.

3.4.2 Optimistic and Pessimistic Effectiveness

Before participating in a continuous evaluation campaign, it is possible to assess the performance of an IRS by exploiting existing relevance judgements to compute optimistic and pessimistic bounds on its future performance.

Some metrics come with methods to bound the effectiveness of the systems being evaluated. In RBP [28], for example, it is possible to compute the "residual" of a measurements to quantify its uncertainty value coming from missing judgements. By using this residual it is possible to simply compute an upper and a lower bound to the effectiveness of the IRSs being evaluated. In the following we propose bounds that give an idea on how the performance of the new system r compare to those of the best system of the current step of the campaign. On the one hand, the *optimistic effectiveness* (denoted by $\Delta^{+}(r)$) gives information on the relative effectiveness difference of the two systems with the assumption that all documents the new IRS needs to judge are relevant; on the other hand, the *pessimistic effectiveness* (denoted by $\Delta^{-}(r)$) is based on the opposite hypothesis. In both cases a score of 0 means that the new IRS has the same effectiveness as the best system found so far, a score greater than 0 means that the new IRS outperforms the best, and a negative score means that the system underperforms it. It is worth noticing that, since relevant documents are often a very small part of the entire document collection, the actual score of a system is closer to the pessimistic bound rather then to the optimistic one. Nevertheless, optimistic relative distance can be used to understand if the new system has any chance to outperform the best system found so far or not. Formally, let m be the evaluation metrics used in the campaign to rank the runs and r be the new run willing to join the campaign at the $(n+1)$-th step, we define the relative effectiveness of r at the step n of the campaign as

$$\Delta_m(r)|_n = \frac{m|_n(r) - \max_{s\in\mathcal{R}|_n}(m|_n(s))}{\max_{s\in\mathcal{R}|_n}(m|_n(s))}, \tag{2}$$

where $\max_{s\in\mathcal{R}|_n}(m|_n(s))$ is the score attained by the best system of the n-th step of the campaign. Practically, Δ_m is the relative difference (measured by the specified metrics m) between the effectiveness of the new system and that of the best system found so far. Starting from Δ_m we define $\Delta^{-}_m(r)$ and $\Delta^{+}_m(r)$ as the values of Δ_m computed by setting all the documents selected to be judged as non-relevant and relevant, respectively. We note that for many well-known metrics $\Delta_m(r)|_n = \Delta^{-}_m(r)|_n$ because unjudged are assumed not relevant, however, this may not be true if the probability of relevance is used in order to evaluate runs.

We observe a link between the two gauges we proposed and the use of the RBP residuals: in both cases an unjudged document is assumed to be relevant in order to compute a best case bound of the considered metrics. However, optimistic and pessimistic effectiveness are strictly related to a continuous evaluation setup as their goal is to give hints on how likely an IRS is to be better than the best among an existing set of systems.

3.4.3 Opportunistic Number of Relevant Documents

Another measure based on $\varDelta_m$ is the *opportunistic number of relevant documents*, denoted by $\rho_m^+(r,t)|_n$. This is a per topic metrics defined as the minimum number of new relevant documents needed in order to attain $\varDelta_m(r)|_n \geq t$, where t is a predefined improvement threshold over the current best system, and n and r are as defined previously. ρ_m^+ can be used to assess how much money one has to spend *in the best case* (that is, all judged documents are relevant) in order to obtain enough judgements to outperform the best system so far by $t\%$. Obviously, it may happen that the current best system is actually more effective than r, no matter how many judgements one does. In this case, it is not possible to reach the desired threshold, thus we set $\rho_m^+(r)|_n = +\infty$. We note that, in order to compute $\rho_m^+(r)|_n$ for a specified topic, one needs to take into consideration the documents retrieved by both r and the by the current best run b since it may happen that the same non-judged document retrieved by both runs is ranked higher in r but gives a greater improvement in the effectiveness of b rather than in that of r.

3.5 Selecting Documents to Judge

The next step of the continuous evaluation process involves the selection of the additional documents to judge at step $n+1$. The selection of the documents to be judged goes under the name of *pooling*. In the IR field, different pooling approaches have been proposed and are currently being used.

3.5.1 Existing Pooling Strategies

In this paper we take into consideration four well-known pooling strategies: fix-depth pooling [19], Aslam and Pavlu's random sampling [3], Carterette et al.'s selective pooling [11], and Moffat et al.'s adaptive pooling [27].

Fix-depth pooling is a widely used technique, which defines the set of documents to be judged as the set containing the top-n documents retrieved by a run for each topic. Documents that are already judged are not re-evaluated.

Alsam and Pavlu's random sampling selects, for each topic, a specified number of random documents among the ranked lists produced by different runs following a probability distribution that gives more weight to high-ranked shared documents. In this method also, documents that are already judged are not re-evaluated.

Carterette et al.'s selective pooling selects one document at a time and collects its judgements. Each time, it picks the document that is most likely to maximize the difference in AP between each pair of systems. This process continues until it is possible to conclude with a 95% confidence that the difference among each pair of systems is greater than zero.

Moffat et al.'s adaptive pooling (RBP-pooling, for simplicity) is based on RBP and exploits the contribution of the document to the effectiveness of all the systems being evaluated and the RBP residuals in order to give it a score. We experiment with the "Method C" approach described by Moffat *et al.* [27].

3.5.2 Novel Pooling Strategies

We propose two novel pooling strategies, namely *fair pooling* and *opportunistic pooling*, which are based on the Fairness Score and on the opportunistic number of relevant document defined in Section 3.4.

3.5.3 Fair Pool Construction

Algorithm 1 shows how to construct a judgement pool by maintaining the Fairness Scores as similar as possible across the participating runs. Our algorithm takes as input the list of the runs that participated in the previous steps of the continuous evaluation (*prevRuns*), a set of already judged documents (*judgedDocs*), the new run joining the continuous evaluation (*newRun*), and a fixed number of judgement tokens representing the overall number of documents that can be judged at this step of the continuous evaluation ($judgTokens$).[4] First, the judgement tokens are distributed among ranked documents according to the classic pooling strategy in use. We note that the chosen strategy may assign a token to an already judged document (this happens quite often with fix-depth pooling) or it may be designed not to spend all the available tokens. In both cases, the unassigned tokens are fairly disributed among all runs: For each one of them, the run with the lowest Fairness Score is selected and, among all its ranked lists of documents (one for each topic), the one with the lowest Fairness Score is taken into consideration and its top-ranked unjudged result is inserted into the pool.

This algorithm has two desirable properties: i) it ensures that all runs participating to the pool contribute the same number of judgements (i.e., $judgTokens$ judgements for each topic in Algorithm 1) and ii) it systematically attempts to improve the score of the run that was treated most unfairly so far (i.e., the run that has the lowest number of its top-results judged). We discuss those two points in more detail and run experiments showing how our fair judgement pool construction compares to previous methods in Section 4.3.

[4] For example, given a pre-defined budget of new relevance judgements that can be obtained or crowdsourced.

Algorithm 1 judgement pool construction based on Fairness Scores.

```
Input: $prevRuns$, $judgedDocs$, $newRun$, $judgTokens$
  $FS[\,]$                                         ▷ Fairness Scores for the runs
  $FStopics[\,][\,]$                               ▷ FS for each run for each topic
  $toJudge \leftarrow$ poolingStrategy($newRun$, $judgTokens$)
  $morejudgements \leftarrow$ TRUE
  while $|toJudge| \leq judgTokens \wedge moreJudgememts$ do
      $poolDocs \leftarrow judgedDocs \cup toJudge$
      for all $run \in prevRuns \cup \{newRun\}$ do
          $FS[run] \leftarrow$ FS($run$, $poolDocs$)
          $FStopics[run] \leftarrow$ FSPerTopic($run$, $poolDocs$)
      end for
      $unfairRun \leftarrow \min(FS)$
      $unfairTopic \leftarrow \min(FStopics[unfairRun])$
      $fairestDoc \leftarrow$ topUnjudged($unfairRun$, $unfairTopic$, $poolDocs$)
      $toJudge \leftarrow toJudge \cup fairestDoc$
      $morejudgements \leftarrow fairestDoc \neq \emptyset$
  end while
  return $toJudge$
```

3.5.4 Opportunistic Pooling

Opportunistic pooling is an application of the ρ^+_m metric previously defined in Section 3.4 and is designed to work in the context of a continuous evaluation campaign. In order to use this pooling strategy, one needs to set two parameters: the overall number n of documents that can be judged (i.e., the judging budget), and an improvement threshold t. At the i-th step of the campaign, $j = \min\left(n, \rho^+_m(r,t)|_{i-1}\right)$ judgements per topic are made, where r is the new run joining the campaign. For each topic, the documents to be judged are chosen by running j times Algorithm 2 to select the j documents that maximize the gap between the effectiveness of r and that of the current best run, measured by using $m|_{i-1}$. In order to select the best document to judge, Algorithm 2 scans all r and, for each unjudged document d, compares the increment in effectiveness obtained if d were relevant with the current effectiveness score. In this way, new judgements are selected to favor r. Notice that the number n of documents to judge can be obtained from the threshold t, and viceversa. On the one hand, we decide to impose a limit (n) on the maximum number of documents to judged in order to avoid situation in which too many documents need to be judged; on the other hand, the number of documents selected for judgement can be lower than n if the threshold is reached with fewer documents. In this case, relevance judgements (and thus budget) are saved by stopping the document selection process when a satisfying optimistic improvement over the currently best system is reached. Notice that n could be not large enough to achieve the specified improvement or that the new system cannot actually outperform the best system found so far. As this can be computed before obtaining the relevance judgements (and, in particular, before

Algorithm 2 Opportunistically Select Best Document to Judge.

```
Input: $b$, $r$ ranked lists of documents for a given topic.
  $maxImp \leftarrow 0$
  $bestDoc \leftarrow$ None
  for all unjudged documents $d$ in $r$ do
      $r\_imp \leftarrow \mathrm{m}_{rel(d)=1}(r)$ - $\mathrm{m}(r)$
      $b\_imp \leftarrow \mathrm{m}_{rel(d)=1}(b)$ - $\mathrm{m}(b)$
      $imp \leftarrow r\_imp$ - $b\_imp$
      if $imp > maxImp$ then
          $bestDoc \leftarrow d$
          $maxImp \leftarrow imp$
      end if
  end for
```

participating in the campaign), it is up to the aspiring participant to decide if it is worth joining the evaluation anyway by gathering n relevance judgements per topic or not. An alternative could consist in generalizing opportunistic pooling to sequentially compare the new system against more than one run (e.g., against the top-k best runs). The algorithm should be thus run once for each comparison. For example, the new system may not outperform the best system so far in n judgements, but it could outperform the second best system, and so on. Finally, opportunistic pooling differs from the approach by Carterette et al. [12] since it does not require judgements to be done one after the other and it is generalizable to any metrics. We study the effectiveness of opportunistic pooling in Section 4.3.

3.6 Obtaining and Integrating Judgements

When creating the set of judged documents, the more diverse the IRSs participating to the pool, the more likely it is to include all relevant documents in the judgement set. Following this intuition, we observe that if all potential IRSs could participate in the pool, both the evaluation of the original set of IRSs as well as the completeness of the collection would improve.

How shall one obtain new relevance judgements? Ideally, the same group of human assessors who originally judged the relevance of documents for topic $t \in \mathcal{T}$ should judge the subsequently retrieved documents to extend $\mathcal{J}$. Even if this were possible, the resulting judgements might still be biased since they would be judged at a different point in time [26]. In order to extend $\mathcal{J}$, one could either use professional assessors or crowdsourcing. In either case, there is a need to integrate the new judgements into the set of previously available ones.

3.6.1 Dealing with Assessment Diversity

As previously shown [26], different human assessors may provide different relevance judgements for the same topic/document pair as relevance is a somewhat subjective notion. Moreover, the same human assessor may provide different judgements at different points in time as pointed out above. Those problems are even more apparent when we replace human judges with a crowd of Web users. While Blanco *et al.* [6] showed that IR evaluation performed by means of crowdsourcing is reliable and repeatable (that is, IRSs rankings are stable according to Kendall's correlation), they also observed that absolute values of effectiveness measures (e.g., Average Precision) can vary as judgements are made by different "crowds". Therefore, while it is sound to evaluate distinct sets of IRSs using crowdsourced or professional judgements, merging judgements coming from heterogeneous crowds/judges queried at different points in time might generate unstable rankings over the course of a continuous evaluation campaign (see Section 4.4 for an illustration of this point).

One thus has to also consider *assessment diversity* when extending a set of relevance judgements, for example, by means of crowdsourcing. The approach we propose below consists in selecting a judgement baseline, i.e., a set of topic/document pairs that must be judged by all assessors involved in the creation and extension of the collection. Thanks to the judgements made over this set, it is possible to assess the *strictness* or *tolerance* of individual judges as compared to the rest of the assessors.

More specifically, after evaluating the first run in the campaign we select a set of topic/document pairs to create a *Common Judgement Set* (CJS) that will be assessed by every other participating assessor. We note that bigger $CJSs$ yield higher costs to balance the crowd assessment diversity, with no addition to the actual set of relevance judgements (since we do not let assessors get any additional judgement but rather let them align their results to existing judgements). However, the more common judgements we gather, the better the adjustments we can potentially make. In the context of crowdsourced relevance judgements, for example on the Amazon Mechanical Turk (AMT) platform (which is commonly used for crowdsourcing experiments), it is possible to implement the CJS feature as a Qualification Test each worker has to complete in order to get access to relevance judgement HITs. An alternative is to implement the common judgements as additional tasks in each HIT in order to better integrate the CJS with the rest of the relevance judgement tasks.

Once the integration stage has finished, it is possible to use the new set of judgements in order to evaluate all the runs participating to the campaign. The $m|_n$ notation previously described helps avoiding comparing values of the same metrics computed using different sets of judgements.

4 Experimental Evaluation

Here we evaluate what we described in the previous section: we show how evaluation metrics and continuous evaluation gauges evolve in a continuous evaluation setting, we present a comparison of the pooling strategies described in Section 3, and we present a small deployment of a continuous evaluation campaign based on the data from the SemSearch 2011 competition.

4.1 Experimental Setting

We use three standard collections in order to evaluate our approach: the testset created in the context of the SemSearch challenge 2011[5] (SemSearch11) for ad-hoc object retrieval, and the collections created in the context of the Ad Hoc task at TREC-7 (TREC7) [35] and TREC-8 (TREC8) [36].

The SemSearch11 collection is based on the Billion Triple Challenge 2009 dataset which consists of 1.3 billion RDF triples crawled from the Web. The TREC7 and TREC8 collections are based on a dataset of 528,155 documents from the Financial Times, the Federal Register, the Foreign Broadcast Information Service, and the LA Times. All collections come with 50 topics together with relevance judgements. However, SemSearch11 differs from the two TREC collections in the number of documents the systems should return for each topic (respectively, 100 and 1,000), in the number of submitted runs (10 for SemSearch11, 103 for TREC7, and 129 for TREC8), and—most importantly—in the way the relevance judgements are computed: in both the collections fix-depth pooling is used but in SemSearch11 the top 10 documents were judged by means of *crowdsourcing*, while in TREC7 and in TREC8 the top 100 documents were evaluated by NIST annotators. Using crowdsourcing to obtain relevance judgements can lead to situations in which relevance judgements for a topic are made by different assessors; this contrasts with the approach used by TREC, in which the same annotator who created the topic also makes the relevance judgements, and no assumption about the generalizability of the relevance assessments is made. To compare the effectiveness of the various IRSs, we base our rankings on Average Precision (AP), which was used both in the TREC-7 and TREC-8 Ad Hoc task as well as in the SemSearch challenge 2011.

One of our goals is to compare the results obtained through a continuously updated evaluation collection against the optimal case of a collection having complete relevance judgements. For this reason, we created the fully-judged TREC7 sub-collection (JTREC7) as the ideal collection whose documents are composed of all judged documents of TREC7, and whose runs are computed from the original runs by removing all unjudged entries and by ranking the remaining documents according to their original order. The new JTREC7 collection, for which each retrieved document of each run has been judged, is composed of 80,345 documents taken from TREC7, and 103 runs containing an average of 411 retrieved documents per topic. Analogously, starting from TREC8 we define JTREC8, another fully-judged collection composed by 86,830 documents and 129 runs containing an average of 431 retrieved documents per topic. Additionally, we define a variant of JTREC7 named JTREC7BPT, which includes the best run per team, resulting in 41 runs (i.e., steps of the continuous evaluation).

We experimentally compare the following pooling strategies in the context of a continuous IR evaluation simulated over the JTREC7 collection:

[5] `http://km.aifb.kit.edu/ws/semsearch11/`

fix-depth pooling, random sampling pooling [3], selective pooling [11], and the new strategies we introduced in Section 3.[6]

4.2 Continuous Evaluation Statistics

4.2.1 Evaluation Metrics

Since the number of available relevance judgements increases at each step of a continuous evaluation campaign, the values of the evaluation metrics used to rank the IRS become increasingly accurate. For example, Figure 2 (top) shows the value of AP as a function of the number of judgements j for five different IRSs in JTREC7. In general, the bigger the value of j, the lower the resulting AP (since its denominator, i.e., the number of relevant documents, steadily increases). As expected, these variations on the metrics used to rank IRSs lead to a variation on the accuracy of the ranking. Figure 2 (bottom) shows, for each step s of a continuous evaluation campaign, the evolution of accuracy of the ranking computed at step s using AP. The accuracy is calculated by means of Kendall's correlation between the ranking produced by computing the AP of the IRSs at step s by using the judgements available at that point in time and the ranking obtained by computing the AP of the same systems but with all the relevance judgements. While the variability of the rankings is high for the first few steps, the rankings become relatively stable after 20 steps (when passing from two to three system it is more likely for the statistics to change than when passing from twenty to twenty-one), hence increasing the values of Kendall's τ). Notice that Figure 2 (bottom) does not reflect the fact that a system before obtaining new judgements can be unfairly ranked lower than after having obtained them.

4.2.2 Continuous Evaluation Gauges

Figure 3 shows how the variance of the Fairness Scores increases as we perform more steps of a continuous evaluation campaign for both sampling-based pooling and fix-depth pooling with Fairness. As expected, the fairness-aware algorithm treats the runs in a fairer way, i.e., we observe higher FS values as we progress in the campaign. In addition, Table 1 shows the final difference between the maximum and minimum values of the Fairness Score among all the runs in the continuous evaluation. As we can see, when we follow Algorithm 1, the final difference is reduced. The minimum is attained by using the fair variant of the fix-depth pooling strategy.

4.3 Pooling Strategies in a Continuous Evaluation Setting

We now analyze the pooling strategies presented in Section 3.5 by measuring how well they select the documents.

[6] We have made available the raw data we used for our experimental evaluation, our raw crowdsourcing results, as well as a set of tools to handle continuous evaluation campaigns using TREC-like evaluation collections at `http://exascale.info/ceval`.

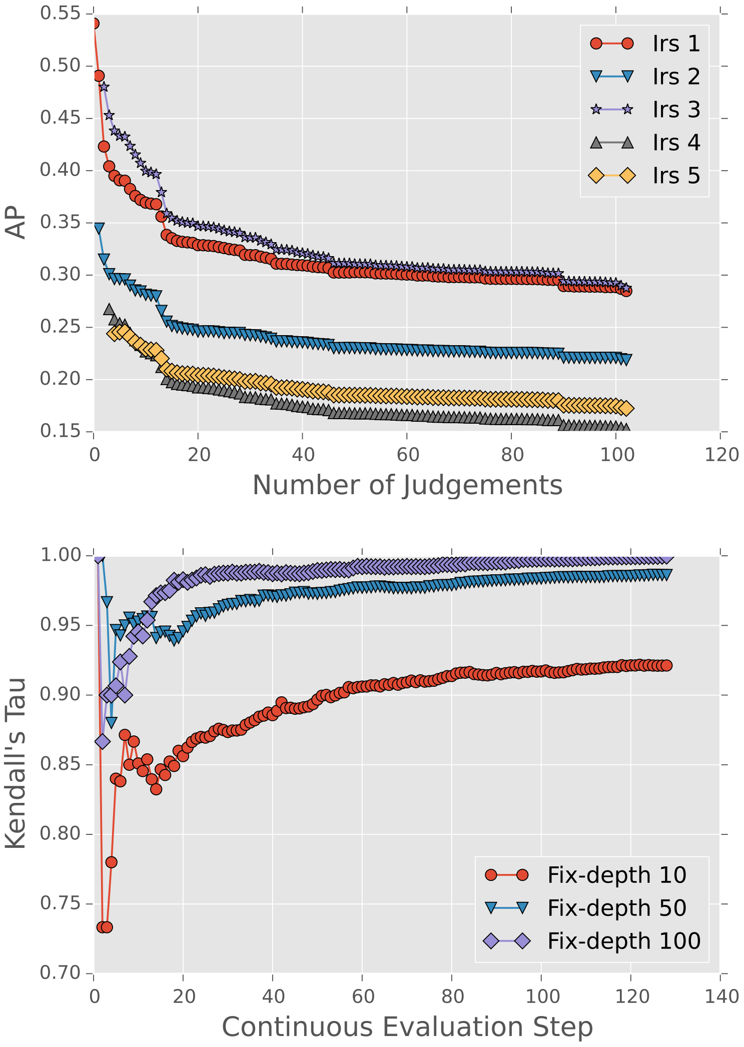

Fig. 2: Evolution of AP values (top) and Kendall's correlation (bottom) during a simulation of a continuous evaluation campaign based on JTREC7. Runs join the campaign based on the lexicographic order of their filename.

4.3.1 Effectiveness of Pooling Strategies

We measure the effectiveness of the pooling strategies described in Section 3 by analyzing the evolution of ranking correlation over the steps of a *simulated* continuous evaluation campaign and the number of selected documents to judge. In all cases we used AP in order to compute the rankings except for RBP-pooling and Aslam-related poolings in which RBP and a variant of AP

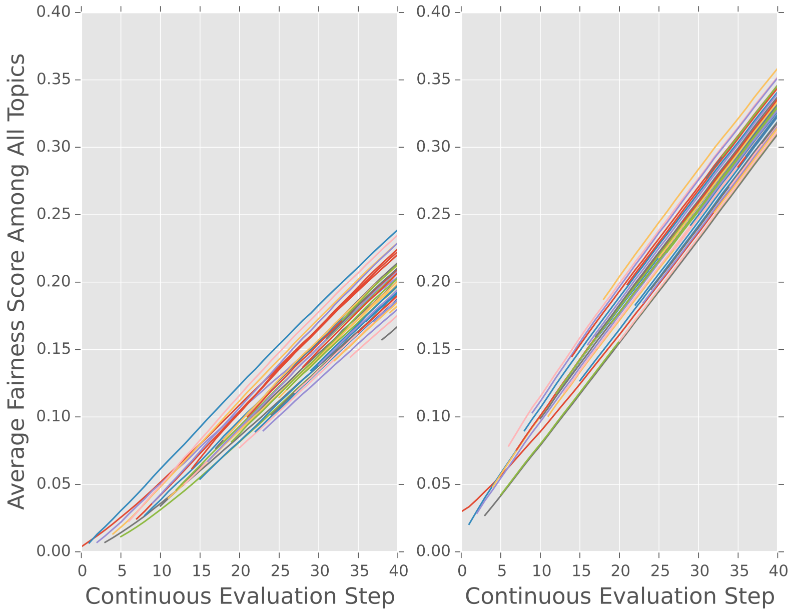

Fig. 3: Fairness Scores evolution for each run participating to a continuous evaluation using sampling-based pooling (left) and fix-depth pooling with Fairness (right) on a sample of 50 randomly selected permutations of JTREC7BPT runs. Each permutation represents a possible order in which the systems join the campaign. At each step 10 new relevance judgements are obtained for each topic.

Pooling approach	Max-Min Fairness Score
Sampling-based	0.95
Opportunistic $t = 0.25$	0.86
Fix-depth	0.76
Sampling-based w/ Fairness	0.23
Fix-depth w/ Fairness	0

Table 1: Max-Min Fairness Scores after a continuous evaluation with and without Alg. 1 with 50 judgement tokens per topic.

[3] are used, respectively. We set $p = 0.8$ for both the computation of RBP and for computing the documents to be judged using RBP-pooling. As for opportunistic pooling, we compare four values of the t threshold on the maximum improvement on the effectiveness of the best system at a certain step: 0.25, 0.3, 0.5, and 0.75. The ranking correlation is computed as described in Section 4.2.1. Figure 4 summarizes our evaluation. A clear limitation of this

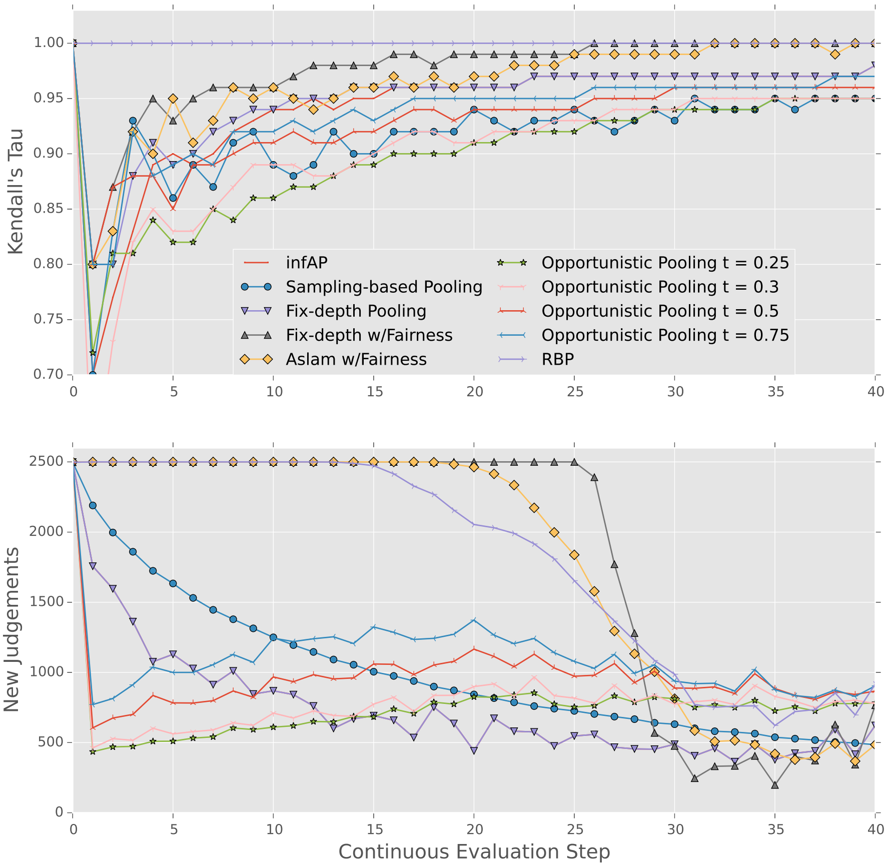

Fig. 4: Effectiveness evaluation of pooling algorithms on 20 random samples of JTREC7BPT runs with 50 new judgedments per topic for each new system participating in the campaign. Ranking correlation with the real ranking (top) and number of judgement per step (bottom). In the first figure three outliers were removed in order to increase readability.

experiment is that we work with only a simulation of a continuous evaluation, that is, we simulate new relevance judgements by using the original assessments done by the TREC assessors and not elicited by people in different points in time.

We observe that: 1) The most effective approaches in terms of correlation to the real ranking (top plot), namely RBP-pooling, Fix-depth w/Fairness and Sampling w/Fairness, are also those that make more relevance judgements per step, and thus more judgements in total. For both approaches, the drop in the number of judgements is due to the fact that all the collection (JTREC7BPT) was judged. 2) RBP-pooling requires a number of judgements comparable to that of Sampling w/Fairness and Fix-depth w/Fairness but the correlation of

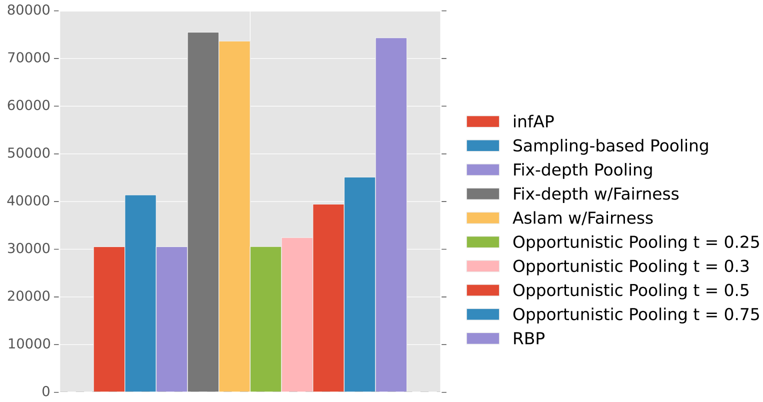

Fig. 5: Total number of judged documents per pooling algorithms.

the ranking produced by RBP with the real ranking is always perfect. Notice that in the case of RBP-pooling we computed the correlation between a ranking produced by RBP with partial judgements and a ranking produces by RBP with all judgements. The geometric weighting used in RBP plays a central role putting most of the weight on top positions and being much less sensitive to variations at lower positions. 3) Using infAP [39] as a measure to estimate AP *before* performing the judgements selected at a given step gives a very good indication of what the IRSs ranking will be after having performed the judgements according to AP. 4) Fix-depth pooling and opportunistic pooling with $t = 0.25$ make a very similar number of judgements (30,538 and 30,568, resp.), however, the former correlates better with the real ranking than the latter. We believe that such a difference in effectiveness is due to the fact that fix-depth pooling selects top documents to be judged, and those documents influence more AP. Opportunistic pooling, on the other hand, tends to avoid selecting those documents if they are shared with other runs (we expect good runs to have many relevant documents shared in top positions). 5) Sampling-based pooling and opportunistic pooling with $t = 0.5$ both require a comparable number of judgements (41,396 and 39,458, resp.) but differ in ranking correlation. We believe that a similar explanation to that used for 3) could clarify this difference in effectiveness: the sampling technique we used samples documents from all the runs, such that the algorithm may select many low ranked documents even if top ranked documents have a higher probability to be picked. The Kendall's tau correlation of Carterette et al.'s pooling strategy with the ideal ranking at the end of a continuous evaluation campaign on JTREC7BPT is 0.25.

In the above experiment we fixed the p parameter of RBP-pooling to 0.8. We study the influence of modeling a more persistent user (that is, a user who is more prone to analyze lower ranked results). Figure 6 shows the behavior of RBP-pooling over a continuous evaluation campaign with varying number of judgements and values of p. As expected, a higher p makes the metrics more sensitive to changes in the lower part of the rankings thus increasing the

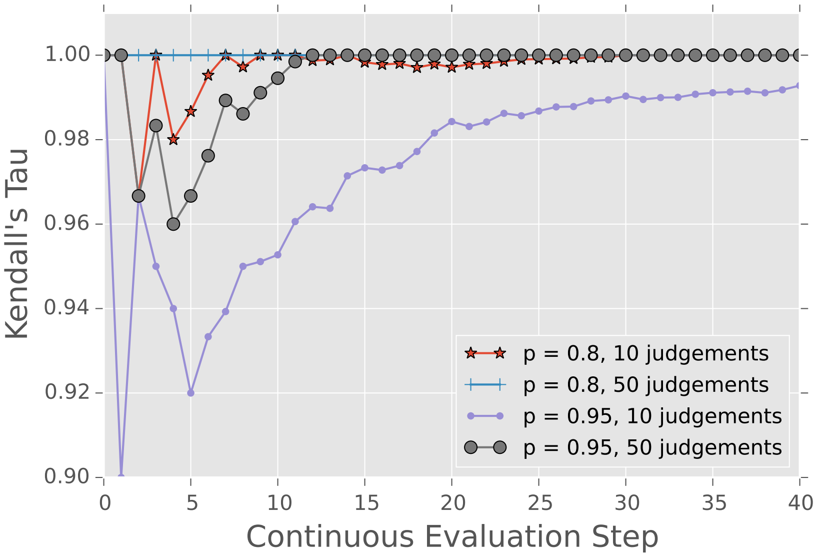

Fig. 6: Evolution of Kendall's correlation during a continuous evaluation campaign on JTREC7BPT. The chart highlights the effect of using different values of the p parameter of RBP and different numbers of judgements per topic at each step.

need for additional relevance judgements in order to produce a more accurate ranking. Anyway, the lower correlation we obtained is 0.9 which is a remarkable result due mostly to the fact that most of the probability mass is concentrated on the top-50 ranked documents. This implies that changes on the relevance of the lower scored documents do not greatly affect RBP.

4.4 Real Deployment of a Continuous Evaluation Campaign

As pointed out in Section 3.3, continuous evaluation campaigns may leverage crowdsourcing in order to obtain relevance judgements over time. To demonstrate the feasibility of such an approach, we run a continuous evaluation for the IRSs participating in the SemSearch 2011 competition. We crowdsourced relevance judgements for SemSearch11 following the same HIT design and using AMT settings[7] similar to the ones originally chosen by the SemSearch 2011 organizers [16]. We run a continuous evaluation by grouping together all runs submitted by the same research group in one evaluation step.[8] As four groups

[7] Three judgements per document made by workers from the U.S. and aggregated with majority vote.

[8] We suppose that during a continuous evaluation initiative each group would have submitted all its runs together as it is quite likely they come from the same system tuned with different values for its parameters.

Measure	1st step (3 runs)	2nd step (5 runs)	3rd step (8 runs)	4th step (10 runs)
τ vs SemSearch11 fix-depth pooling	-0.33	0.6	0.86	0.87
τ vs SemSearch11 sampling	1	0.4	0.79	0.6
τ vs previous step fix-depth pooling	-	0.33	0.8	1
τ vs previous step sampling	-	-0.33	0.4	1
Simulation on SemSearch11 judgements				
τ vs previous step	-	1	0.6	0.8571

Table 2: Kendall τ correlation with SemSearch11 results and stability of the ranking at each step of the continuous evaluation.

submitted runs, we obtain four steps in our continuous evaluation. Additionally, to correctly run a continuous evaluation, we make sure that no crowd worker participates in two different evaluation steps, since evaluations at different points in time are typically handled by different crowds. The judgements were collected in different points in time, as reported in the following:

- First step and CJS: Taken from SemSearch11.
- Second step: May 8–13, 2012.
- Third step: May 14–19, 2012.
- Fourth step: May 18–23, 2012.

The short time span we use to collect relevance judgements (15 days) and the limited number of IRSs taken into consideration are limitations of this evaluation, since we expect a real continuous evaluation to last for some years and to involve many systems. However, the data we collected gives insights on the diversity of the assessments and on the economical feasibility of the proposed methodology.

We analyze the results along two dimensions: How the IRSs ranking compares to the original final SemSearch 2011 results, and how stable the IRSs ranking is across the steps of the continuous evaluation, that is, how much the ranking changes as compared to the previous step measured by Kendall's τ (we call this ranking stability).

Looking at the correlation with the original SemSearch11 ranking (see Table 2), we observe that the best correlation is obtained with fix-depth pooling since it is the same strategy as the one used by the SemSearch11 organizers. Looking at the ranking stability, we see that in all cases more evaluation steps make IRSs ranking more stable. The minimum ranking stability ($\tau = 0.6$) occurs at the third step of our continuous evaluation, when the new judgements produced by the third team reveal that the run ranked 3rd in the 2nd step of

Measure	1st step (3 runs)	2nd step (5 runs)	3rd step (8 runs)	4th step (10 runs)
τ vs SemSearch11 fix-depth pooling	-0.33	0.4	0.71	0.73
τ vs SemSearch11 sampling	-0.33	0.4	0.71	0.82
τ vs previous step fix-depth pooling	-	1	1	0.86
τ vs previous step sampling	-	1	1	1

Table 3: Kendall τ correlation with SemSearch11 results and stability of the ranking at each step of the continuous evaluation when using CJS to mitigate the assessment diversity of the crowd.

the evaluation is actually more effective than the run ranked 1st at that step. Nevertheless, it is worth noticing that the differences between the scores of the two systems were not statistical significant (t-test: $p = 0.9310$ at the second step and $p = 0.9861$ at the third step).

To deal with the assessment diversity of different crowds, as explained in Section 3.6.1, we defined the CJS containing topic-result pairs from the first step of the continuous evaluation and built an AMT Qualification Test that each worker has to perform before starting to judge documents for the current IRS. To balance the judgements we define a parametric function

$$weight(w) = \begin{cases} \alpha & \text{if } (w(i) - cjs(i)) < 0 \\ -\alpha & \text{if } (w(i) - cjs(i)) > 0 \\ 0 & \text{if } (w(i) - cjs(i)) = 0 \end{cases} \tag{3}$$

where $w(i)$ is the judgement given by worker w to the topic/document i, $cjs(i)$ its original judgement, and α is a parameter in $[0, 1]$ that defines how we treat *strict* and *tolerant* workers (so, $weigth(w) = \alpha > 0$ for strict workers and $weight(w) = -\alpha < 0$ for tolerant workers). In our experiments we set $\alpha = 0.5$.

An interesting comparison can be made between the correlation values of the rankings obtained by the raw judgements of different crowds and those obtained by considering CJS to adjust the values (see Table 3). As we can observe, there is a clear improvement in the stability of the ranking over the evaluation steps since we adjust the latter crowds to be more similar to the former ones.

Table 4 shows the rate of documents considered as relevant versus the number of judgements at each step of the evaluation. As we can see, the non-adjusted crowd behaves in a more tolerant way (i.e., more results are considered relevant) as compared to the adjusted crowd where judgements on CJS are compared to those of SemSearch11 to downscale the new judgements.

Evaluation Step:	1	2	3	4
Raw Crowd	0.098	0.091	0.078	0.092
CJS-balanced Crowd	0.095	0.068	0.063	0.073

Table 4: Relevant/Judged rate over the steps of the continuous evaluation of SemSearch11.

5 Discussion

In this section we discuss about few key points which are necessary to deal with when deploying a continuous evaluation campaign in practice. In particular we focus on issues on integrating relevance judgements, on constructing the CJS, we discuss the economical viability of a continuous evaluation and we conclude the section describing how we envision a "More continuous" continuous evaluation.

5.1 Integrating Relevance Judgements

Other aspects of the integration of relevance judgements are to be considered when running a continuous evaluation campaign.

In particular, inconsistencies may arise as a consequence of having several judges and of continuously judging documents on the same corpus in different points in time. We observed that, the more a continuous evaluation advances, the fewer new relevant documents can be found and the more lenient the judges (exposed to many non-relevant documents) may be, thus possibly introducing inconsistencies (cfr. last paragraph of Section 2). We believe that the problem could be attenuated by using more complex weighting functions possibly depending on the number of relevant documents found until now and/or on the step in the continuous evaluation.

Inconsistencies can also arise because during time things change and the answer to a certain topic may be different (e.g., "Who is the president of Italy?"). In this case we believe that the training of the judges and/or the design of the HIT might play a central role: the judge has to know when the topic was chosen in order to provide assessments that are coherent to those composing the dataset. If we assume to crowdsource relevance judgements one option could be to ask the workers to take a particular HIT only if we deem them old enough to answer; however, in current crowdsourcing platforms it is not possible to impose constraints on the age of the workers. Another viable option could be to add an age requirement in the description of the task but anyway we could not be sure of such a requirements being respected. A more reliable solution would be to use push crowdsourcing [15], in which we can leverage the background of the workers and push the tasks to the those workers we deem adequate. Anyway, these methods can only mitigate the effect of time since it may by difficult for the judges to collocate themselves in the right time period necessary to make the requested relevance judgements. In our future work we will focus on understanding how the design of the judging interface and the provision of additional contextual information helping the judge remembering the times in which the collection was built affects crowdsourced relevance judgements.

Finally, inconsistencies can arise simply because different judges interpret in different ways the intent of a keyword query or because they use a different interpretation of relevance. As for the first case, Verbene *et al.* [34] show that keywords are not enough to transmit to the judge the searcher's intent and that, in general, external judges can only correctly extrapolate the topic and the spatial sensitivity of the original search intent. This aspect also has to be taken into consideration when merging different kinds of relevance judgements, for example when mixing TREC-like judgements made by the creators of the topics with crowdsourced judgements. In particular, the resulting test collection might, at some point, have a large number of relevance assessments made by a few people, and a small number made by a lot of different people. We think that a longer description of the topic (e.g., the "narrative" field of TREC topics) can help the judge in deciding about the relevance of the document. We suspect, however, that there might be a trade-off between the precision of the description of the topic and the quality of the judgements obtained: crowd workers are often less prone to read long instructions. It is also worth noticing that filtering the crowd workers by country might help dealing with linguistic and cultural issues (this option is supported by AMT).

5.2 Building the CJS

In our work we defined the judgements composing the CJS to be selected by the organizers of the campaign, possibly with the help of the creators of the topics. Optimally selecting the best judgements to include in the CJS and/or how to adjust the relevance scores provided by the judges is an open research question which will require further investigation. We can envision several way of populating the CJS: The most trivial (and possibly the less valid) option would be not to involve the creators of the topics, crowdsource the first relevance judgements, and select the documents with higher agreement. A more sophisticated method could consist in asking the creators of the topics to provide positive and negative examples of documents for each topic where the negative examples cover different interpretations of the query. For example, the INEX Entity Ranking track [14] topics contain example relevant results as defined by the topic creator. Doing this would possibly make the size of the CJS too large but probably still manageable if the HITs are organized by topic. We believe that it would also be interesting to use per-topic qualification tests in order to accept workers with the same "point of view" as the topic creator.

5.3 Economical Viability

Another question we want to tackle is whether or not it is economically viable to use crowdsourcing to create additional relevance judgements for a new run in a continuous evaluation setting. While the cost of crowdsourcing the relevance judgements for an entire TREC collection may be too high for a single research group (around 10,000 USD according to [2]), the per-run cost in a fair, continuous IR evaluation setting is much more affordable.

As shown in Figure 4 (bottom), in a fix-depth pooling setting the judgement cost per run on average decreases as we add more runs. We observe that the first runs are more expensive to evaluate, which leads to the conclusion that the first steps of a continuous evaluation should be carried out in the context of a classic TREC-like initiative. Assuming that the first 20 runs participated in the evaluation initiative, we compute the cost of creating the remaining judgements for each run assuming that three workers are asked to judge each document and are paid $0.10 for each relevance judgement (this is the standard setting used by most of the approaches we refer to in Section 2). With such settings, the average cost per run is $22 for pool depth 10, $90 for pool depth 50, and $160 for pool depth 100, which in our opinion would be acceptable for most research groups proposing a new system. It is also worth noticing that in some cases it is possible, during the organization of the campaign, to decide to lower the accuracy of the score in order to reduce the cost of relevance evaluations. An example is the RBP metrics by Moffat et al. [28].

5.4 More Continuous Continuous Evaluations

An extended version of the continuous evaluation methodology we proposed in this paper is not bound to a fixed document collection, to a fixed set of topics, or to a fixed set of judgements as all its components change continuously. Notice that by adding new topics the inaccuracies of the old topics get amortized over a growing number of newer topics and, as shown by Sanderson and Zobel in [31], contributes to a more reliable evaluation. One of the crucial points implied by this definition is that it should be always possible to obtain new runs of all the IRSs participating the evaluation, thus requiring effort both of the participants (maintaining a possibly long term access point to their system or providing the organizers with their system) and of the organizers (maintaining a directory of systems ready to run, developing new topics, etc.) depending on how centralized is the campaign. Microsoft already implemented a system that was able to run all the IRSs participating the entity Recognition and Disambiguation Challenge Workshop held at SIGIR 2014. Continuous access to running IRSs is also adopted by the living-lab approach [5], however this process is highly centralized and not really continuous as it does not allow systems to join the evaluation after the end of the initiative. We think that such an approach would not accommodate the needs of many researchers since it requires a constant effort from the organizers.

Other issues we envision regarding the process of integrating new topics, extending the corpus, and dealing with the conflicts of interests that may arise, are, for example, when a research group is responsible for collecting and auditing judgements in which a relevant outcome is likely to favor their own system at the expense of previously-scored ones. In this work we decided to take a conservative approach by proposing an evaluation methodology with fixed topics, "static" systems and honest participants in order keep our methodology simple and focused.

6 Conclusions

Crowdsourcing has already been recognized as a viable alternative to expert assessment when creating evaluation collections. In this paper, we showed how to effectively update IR evaluation collections *continuously* as new systems appear and get evaluated. We also introduced a series of metrics to monitor and compare the IRSs rankings over the course of a continuous evaluation campaign (optimistic and pessimistic effectiveness, opportunistic number of relevant documents), and to ensure that all systems are treated fairly (fairness score). We use those metrics to create fairness-aware pooling strategies and to define opportunistic pooling.

Our approach can be readily applied both on new and on preexisting document collections, as long as the system runs and relevance judgements are available. To demonstrate our approach, we have created and made available a set of tools that ease the setup and the handling of continuous evaluation campaigns using TREC-like evaluation collections.

As future work, we plan to better simulate a continuous evaluation by actually obtaining relevance judgements continuously by using different crowds in order to better analyze the impact of time and of different people on the relevance judgements. Moreover, we envisage to adapt our approach to evolving collections, e.g., to let researcher re-evaluate their techniques iteratively over updated version of Web crawls.

Also, we deem interesting analyzing a context in which one wants to pool for extending the completeness of the test collection instead than for fairness among the systems: if treating the system fairly is not a priority, it is possible to judge deeper into the rankings of, for example, the higher-performing runs.

References

1. O. Alonso and R. A. Baeza-Yates. Design and implementation of relevance assessments using crowdsourcing. In *European Conference on Information Retrieval, ECIR*, pages 153–164, 2011.
2. O. Alonso and S. Mizzaro. Using crowdsourcing for trec relevance assessment. *Information Processing & Management*, 48(6):1053–1066, Nov. 2012.
3. J. Aslam and V. Pavlu. A practical sampling strategy for efficient retrieval evaluation. *Working Draft, http://www.ccs.neu.edu/home/jaa/papers/drafts/statAP.html*, 2007.
4. J. A. Aslam, V. Pavlu, and E. Yilmaz. A statistical method for system evaluation using incomplete judgments. In *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, pages 541–548, 2006.
5. K. Balog, L. Kelly, and A. Schuth. Head First: Living Labs for Ad-hoc Search Evaluation. In *International Conference on Information and Knowledge Management, CIKM*, pages 1815–1818, 2014.
6. R. Blanco, H. Halpin, D. M. Herzig, P. Mika, J. Pound, H. S. Thompson, and D. T. Tran. Repeatable and reliable search system evaluation using crowdsourcing. In *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, pages 923–932, 2011.
7. R. Blanco, H. Halpin, D. M. Herzig, P. Mika, J. Pound, H. S. Thompson, and T. Tran. Repeatable and reliable semantic search evaluation. *Journal of Web Semantics, JWS*, 21:14–29, 2013.
8. C. Buckley, D. Dimmick, I. Soboroff, and E. M. Voorhees. Bias and the limits of pooling for large collections. *Information Retrieval Journal, IRJ*, 10(6):491–508, 2007.
9. C. Buckley and E. Voorhees. Retrieval evaluation with incomplete information. In *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, pages 25–32. ACM, 2004.
10. S. Büttcher, C. L. A. Clarke, P. C. K. Yeung, and I. Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. In *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, pages 63–70, 2007.
11. B. Carterette, J. Allan, and R. K. Sitaraman. Minimal test collections for retrieval evaluation. In *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, pages 268–275, 2006.
12. B. Carterette, E. Kanoulas, V. Pavlu, and H. Fang. Reusable test collections through experimental design. In *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, pages 547–554, 2010.
13. C. Cleverdon. Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems. *College of Aeronautics, Cranfield, England*, 1962.
14. G. Demartini, T. Iofciu, and A. P. de Vries. Overview of the INEX 2009 entity ranking track. In *Focused Retrieval and Evaluation, 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX*, pages 254–264, 2009.
15. D. E. Difallah, G. Demartini, and P. Cudré-Mauroux. Pick-A-Crowd: Tell Me What You Like, and Ill Tell You What to Do. In *International World Wide Web Conference, WWW*, pages 367–374. International World Wide Web Conferences Steering Committee, May 2013.
16. H. Halpin, D. M. Herzig, P. Mika, R. Blanco, J. Pound, H. S. Thompson, and D. T. Tran. Evaluating Ad-Hoc Object Retrieval. In *Evaluation of Semantic Technologies (IWEST 2010) at ISWC2010*, 2010.
17. M. Hosseini, I. J. Cox, N. Milic-Frayling, G. Kazai, and V. Vinay. On aggregating labels from multiple crowd workers to infer relevance of documents. In *European Conference on Information Retrieval, ECIR*, pages 182–194, 2012.
18. K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems, TOIS*, 20(4):422–446, Oct. 2002.
19. K. Jones and C. Van Rijsbergen. *Report on the Need for and Provision of an Ideal Information Retrieval Test Collection.* British Library Research and Development reports. 1975.
20. G. Kazai. In search of quality in crowdsourcing for search engine evaluation. In *European Conference on Information Retrieval, ECIR*, pages 165–176, 2011.

21. G. Kazai, J. Kamps, M. Koolen, and N. Milic-Frayling. Crowdsourcing for book search evaluation: impact of hit design on comparative system ranking. In *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, pages 205–214, 2011.
22. G. Kazai, J. Kamps, and N. Milic-Frayling. Worker types and personality traits in crowdsourcing relevance labels. In *International Conference on Information and Knowledge Management, CIKM*, pages 1941–1944, 2011.
23. M. Lesk and G. Salton. Relevance assessments and retrieval system evaluation. *Information Storage and Retrieval*, 4(4):343 – 359, 1968.
24. C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008.
25. R. McCreadie, C. Macdonald, and I. Ounis. Crowdsourcing Blog Track Top News Judgments at TREC. In *Crowdsourcing for Search and Data Mining (CSDM) at WSDM 2011*, pages 23–26, 2011.
26. S. Mizzaro. Relevance: The whole history. *Journal of the Association for Information Science and Technology, JASIST*, 48(9):810–832, 1997.
27. A. Moffat, W. Webber, and J. Zobel. Strategic System Comparisons via Targeted Relevance Judgments. In *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, pages 375–382, 2007.
28. A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems, TOIS*, 27(1):2:1–2:27, Dec. 2008.
29. Y. Moshfeghi, M. Matthews, R. Blanco, and J. M. Jose. Influence of timeline and named-entity components on user engagement. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), LNCS*, volume 7814 LNCS, pages 305–317, 2013.
30. V. Pavlu, S. Rajput, P. B. Golbus, and J. A. Aslam. Ir system evaluation using nugget-based test collections. In *International Conference on Web Search and Data Mining, WSDM*, pages 393–402, New York, NY, USA, 2012. ACM.
31. M. Sanderson and J. Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, page 162, New York, New York, USA, Aug. 2005. ACM Press.
32. F. Scholer, D. Kelly, W.-C. Wu, H. S. Lee, and W. Webber. The effect of threshold priming and need for cognition on relevance calibration and assessment. In *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, page 623632. ACM, 2013.
33. M. Smucker, G. Kazai, and M. Lease. Overview of the trec 2012 crowdsourcing track. In *NIST Text Retrieval Conference, TREC*, 2013.
34. S. Verberne, M. V. D. Heijden, M. Hinne, M. Sappelli, S. Koldijk, E. Hoenkamp, and W. Kraaij. Reliability and Validity of Query Intent Assessments. *Journal of the Association for Information Science and Technology, JASIST*, 64(August):2224–2237, 2013.
35. E. Voorhees and D. Harman. Overview of the Seventh Text REtrieval Conference (TREC-7). *NIST Special Publication 500-242*, 1998.
36. E. Voorhees and D. Harman. Overview of the Eighth Text REtrieval Conference (TREC-8). *NIST Special Publication 500-246*, 1999.
37. E. M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, pages 315–323, New York, NY, USA, 1998. ACM.
38. W. Webber and L. A. F. Park. Score adjustment for correction of pooling bias. In *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, pages 444–451, 2009.
39. E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *International Conference on Information and Knowledge Management, CIKM*, pages 102–111, 2006.
40. E. Yilmaz, E. Kanoulas, and J. A. Aslam. A simple and efficient sampling method for estimating AP and NDCG. In *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, pages 603–610, 2008.
41. J. Zobel. How reliable are the results of large-scale information retrieval experiments? In *International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, pages 307–314, New York, NY, USA, 1998. ACM.