Information Processing and Management 54 (2018) 37–59

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Intelligent topic selection for low-cost information retrieval evaluation: A New perspective on deep vs. shallow judging

Mucahid Kutlu*[,a], Tamer Elsayed[a], Matthew Lease[b]

[a] Dept. of Computer Science and Engineering, Qatar University, Qatar
[b] School of Information, University of Texas at Austin, USA

ABSTRACT

While *test collections* provide the cornerstone for Cranfield-based evaluation of information retrieval (IR) systems, it has become practically infeasible to rely on traditional *pooling* techniques to construct test collections at the scale of today's massive document collections (e.g., ClueWeb12's 700M+ Webpages). This has motivated a flurry of studies proposing more cost-effective yet reliable IR evaluation methods. In this paper, we propose a new *intelligent topic selection* method which reduces the number of search topics (and thereby costly human relevance judgments) needed for reliable IR evaluation. To rigorously assess our method, we integrate previously disparate lines of research on intelligent topic selection and *deep* vs. *shallow* judging (i.e., whether it is more cost-effective to collect many relevance judgments for a few topics or a few judgments for many topics). While prior work on intelligent topic selection has never been evaluated against shallow judging baselines, prior work on deep vs. shallow judging has largely argued for shallowed judging, but assuming random topic selection. We argue that for evaluating any topic selection method, ultimately one must ask whether it is actually useful to select topics, or should one simply perform shallow judging over many topics? In seeking a rigorous answer to this over-arching question, we conduct a comprehensive investigation over a set of relevant factors never previously studied together: 1) method of topic selection; 2) the effect of topic familiarity on human judging speed; and 3) how different topic generation processes (requiring varying human effort) impact (i) budget utilization and (ii) the resultant quality of judgments. Experiments on NIST TREC Robust 2003 and Robust 2004 test collections show that not only can we reliably evaluate IR systems with fewer topics, but also that: 1) when topics are intelligently selected, deep judging is often more cost-effective than shallow judging in evaluation reliability; and 2) topic familiarity and topic generation costs greatly impact the evaluation cost vs. reliability trade-off. Our findings challenge conventional wisdom in showing that deep judging is often preferable to shallow judging when topics are selected intelligently.

1. Introduction

Test collections provide the cornerstone for system-based evaluation of information retrieval (IR) algorithms in the Cranfield paradigm (Cleverdon, 1959). A test collection consists of: 1) a *collection* of documents to be searched; 2) a set of pre-defined user *search topics* (i.e., a set of topics for which some users would like to search for relevant information, along with a concise articulation of each topic as a *search query* suitable for input to an IR system); and 3) a set of human *relevance judgments* indicating the relevance of

* Corresponding author.
E-mail addresses: mucahidkutlu@qu.edu.qa (M. Kutlu), telsayed@qu.edu.qa (T. Elsayed), ml@utexas.edu (M. Lease).

http://dx.doi.org/10.1016/j.ipm.2017.09.002
Received 2 March 2017; Received in revised form 17 July 2017; Accepted 14 September 2017

collection documents to each search topic. Such a test collection allows empirical A/B testing of new search algorithms and community benchmarking, thus enabling continuing advancement in the development of more effective search algorithms. Because exhaustive judging of all documents in any realistic document collection is cost-prohibitive, traditionally the top-ranked documents from many systems are *pooled*, and only these top-ranked documents are judged. Assuming the *pool depth* is sufficiently large, the reliability of incomplete judging by pooling is well-established (Sanderson, 2010).

However, if insufficient documents are judged, evaluation findings could be compromised, e.g., by erroneously assuming unjudged documents are not relevant when many actually are relevant (Buckley, Dimmick, Soboroff, & Voorhees, 2006). The great problem today is that: 1) today's document collections are increasingly massive and ever-larger; and 2) realistic evaluation of search algorithms requires testing them at the scale of document collections to be searched in practice, so that evaluation findings in the lab carry-over to practical use. Unfortunately, larger collections naturally tend to contain many more relevant (and seemingly-relevant) documents, meaning human *relevance assessors* are needed to judge the relevance of ever-more documents for each search topic. As a result, evaluation costs have quickly become cost prohibitive with traditional pooling techniques (Sanderson, 2010). Consequently, *a key open challenge in IR is to devise new evaluation techniques to reduce evaluation cost while preserving evaluation reliability*. In other words, how can we best spend a limited IR evaluation budget?

A number of studies have investigated whether it is better to collect many relevance judgments for a few topics – i.e., *Narrow and Deep* (NaD) judging – or a few relevance judgments for many topics – i.e., *Wide and Shallow* (WaS) judging, for a given evaluation budget. For example, in the TREC Million Query Track (Carterette, Pavlu, Kanoulas, Aslam, & Allan, 2009), IR systems were run on ~ 10K queries sampled from two large query logs, and shallow judging was performed for a subset of topics for which a human assessor could ascribe some intent to the query such that a topic description could be back-fit and relevance determinations could be made. Intuitively, since people search for a wide variety of topics expressed using a wide variety of queries, it makes sense to evaluate systems across a similarly wide variety of search topics and queries. Empirically, large variance in search accuracy is often observed for the same system across different topics (Banks, Over, & Zhang, 1999), motivating use of many diverse topics for evaluation in order to achieve stable evaluation of systems. Prior studies have reported a fairly consistent finding that WaS judging tends to provide more stable evaluation for the same human effort vs. NaD judging (Bodoff & Li, 2007; Carterette & Smucker, 2007; Sanderson & Zobel, 2005). While this finding does not hold in all cases, exceptions have been fairly limited. For example, Carterette, Pavlu, Kanoulas, Aslam, and Allan 2008 achieve the same reliability using 250 topics with 20 judgments per topic (5000 judgments in total) as 600 topics with 10 judgments per topic (6000 judgments in total). A key observation we make in this work is noting that all prior studies comparing NaD vs. WaS judging assume that search topics are selected randomly.

Another direction of research has sought to carefully choose which search topics are included in a test collection (i.e., *intelligent topic selection*) so as to minimize the number of search topics needed for a stable evaluation. Since human relevance judgments must be collected for any topic included, using fewer topics directly reduces judging costs. NIST TREC test collections have traditionally used 50 search topics (manually selected from a larger initial set of candidates), following a simple, effective, but costly topic creation process which includes collecting initial judgments for each candidate topic and manual selection of final topics to keep (Voorhees, 2001). Buckley and Voorhees 2000 report that at least 25 topics are needed for stable evaluation, with 50 being better, while Zobel 1998 showed that one set of 25 topics predicted relative performance of systems fairly well on a different set of 25 topics. Guiver, Mizzaro, and Robertson 2009 conducted a systematic study showing that evaluating IR systems using the "right" subset of topics yields very similar results vs. evaluating systems over all topics. However, they did not propose a method to find such an effective topic subset in practice. Most recently, Hosseini, Cox, Milic-Frayling, Shokouhi, and Yilmaz 2012 proposed an iterative algorithm to find effective topic subsets, showing encouraging results. A key observation we make is that prior work on intelligent topic selection has not evaluated against shallow judging baselines, which tend to be the preferred strategy today for reducing IR evaluation cost. We argue that one must ask whether it is actually useful to select topics, or should one simply perform WaS judging over many topics?

Our Work. In this article, we propose a new *intelligent topic selection* method which reduces the number of search topics (and thereby costly human relevance judgments) needed for reliable IR evaluation. To rigorously assess our over-arching question of whether topic selection is actually useful in comparison to WaS judging approaches, we integrate previously disparate lines of research on intelligent topic selection and NaD vs. WaS judging. Specifically, we investigate a comprehensive set of relevant factors never previously considered together: 1) method of topic selection; 2) the effect of topic familiarity on human judging speed; and 3) how different topic generation processes (requiring varying human effort) impact (i) budget utilization and (ii) the resultant quality of judgments. We note that prior work on NaD vs. WaS judging has not considered cost ramifications of how judging depth impacts judging speed (i.e., assessors becoming faster at judging a particular topic as they become more familiar with it). Similarly, prior work on NaD vs. WaS judging has not considered topic construction time; WaS judging of many topics appears may be far less desirable if we account for traditional NIST TREC topic construction time (Voorhees, 2016). As such, our findings also further inform the broader debate on NaD vs. WaS judging assuming random topic selection.

We begin with our first research question **RQ-1**: *How can we select search topics that maximize evaluation validity given document rankings of multiple IR systems for each topic?* We propose a novel application of *learning-to-rank* (L2R) to topic selection. In particular, topics are selected iteratively via a greedy method which optimizes accurate ranking of systems (Section 4.3). We adopt MART (Friedman, 2001) as our L2R model, though our approach is largely agnostic and other L2R models might be used instead. We define and extract 63 features for this topic selection task which represent the interaction between topics and ranking of systems (Section 4.3.1). To train our model, we propose a method to automatically generate useful training data from existing test collections (Section 4.3.2). By relying only on pre-existing test collections for model training, we can construct a new test collection without any prior relevance judgments for it, rendering our approach more generalizable and useful. We evaluate our approach on NIST TREC

Robust 2003 (Voorhees, 2003) and Robust 2004 (Voorhees, 2004) test collections, comparing our approach to recent prior work (Hosseini et al., 2012) and *random* topic selection (Section 5). Results show consistent improvement over baselines, with greater relative improvement as fewer topics are used.

In addition to showing improvement of our topic selection method over prior work, as noted above, we believe it is essential to assess intelligent topic selection in regard to the real over-arching question: what is the best way to achieve cost-effective IR evaluation? Is intelligent topic selection actually useful, or should we simply do WaS judging over many topics? To investigate this, we conduct a comprehensive analysis involving a set of focused research questions not considered by the prior work, all utilizing our intelligent topic selection method:

- **RQ-2: When topics are selected intelligently, and other factors held constant, is WaS judging (still) a better way to construct test collections than NaD judging?**When intelligent topic selection is used, we find that NaD judging often achieves greater evaluation reliability than WaS judging for the same budget when topics are selected intelligently, contrasting with popular wisdom today favoring WaS judging.
- **RQ-3 (Judging Speed and Topic Familiarity): Assuming WaS judging leads to slower judging speed than NaD judging, how does this impact our assessment of intelligent topic selection?** Past comparisons between NaD vs. WaS judging have typically assumed constant judging speed (Carterette & Smucker, 2007; Sanderson & Zobel, 2005). However, data reported by Carterette et al. (Carterette et al., 2009) suggests that assessors may judge documents faster as they judge more documents for the same topic (likely due to increased topic familiarity). Because we can collect more judgments in the same amount of time with NaD vs. WaS judging, we show that NaD judging achieves greater relative evaluation reliability than shown in prior studies, which did not consider the speed benefit of deep judging.
- **RQ-4 (Topic Development Time): How does topic development time in the context of NaD vs. WaS judging impact our assessment of intelligent topic selection?** Prior NaD vs. WaS studies have typically ignored non-judging costs involved in test collection construction. While Carterette et al. 2008 consider topic development time, the 5-minute time they assumed is roughly two orders of magnitude less than the 4 hours NIST has traditionally taken to construct each topic (Voorhees, 2016). We find that WaS judging is preferable than NaD judging for short topic development times (specifically ≤ 5 minutes in our experiments). However, as the topic development cost further increases, NaD judging becomes increasingly preferable.
- **RQ-5 (Judging Error): Assuming short topic development times reduce judging consistency, how does this impact our assessment of intelligent topic selection in the context of NaD vs. WaS judging?** Several studies have reported calibration effects impacting the decisions and consistency of relevance assessors (Sanderson, Scholer, & Turpin, 2010; Scholer, Kelly, Wu, Lee, & Webber, 2013). While NIST has traditionally included an initial “burn-in” judging period as part of topic generation and formulation (Voorhees, 2016), we posit that drastically reducing topic development time (e.g., from 4 hours (Voorhees, 2016) to 2 minutes (Carterette et al., 2009)) could negatively impact topic quality, leading to less well-defined topics and/or calibrated judges, and thereby less reliable judgments. As suggestive evidence, McDonnell, Lease, Kutlu, and Elsayed 2016 report high judging agreement in reproducing a “standard” NIST track, but high and inexplicable judging disagreement on TREC’s Million Query track (Carterette et al., 2009), which lacked any burn-in period for judges and had far shorter topic generation times. To investigate this, we simulate increased judging error as a function of lower topic generation times. We find that it is better to invest a portion of our evaluation budget to increase quality of topics, instead of collecting more judgments for low-quality topics. This also makes NaD judging preferable in many cases, due to increased topic development cost.

Contributions. Our five research questions address the over-arching goal and challenge of minimizing IR evaluation cost while ensuring validity of evaluation. Firstly, we propose an intelligent topic selection algorithm, as a novel application of learning-to-rank, and show its effectiveness vs. prior work. Secondly, we go beyond prior work on topic selection to investigate whether it is actually useful, or if one should simply do WaS judging over many topics rather than topic selection? Our comprehensive analysis over several factors not considered in prior studies shows that intelligent topic selection is indeed useful, and contrasts current wisdom favoring WaS judging.

The remainder of this article is organized as follow. Section 2 reviews the related work on topic selection and topic set design. Section 3 formally defines the topic selection problem. In Section 4, we describe our proposed L2R-based approach in detail. Section 5 presents our experimental evaluation. Finally, Section 6 summarizes the contributions of our work and suggests potential future directions.

2. Related work

Constructing test collections is expensive in human effort required. Therefore, researchers have proposed a variety of methods to reduce the cost of creating test collections. Proposed methods include: developing new evaluation measures and statistical methods for the case of incomplete judgments (Aslam, Pavlu, & Yilmaz, 2006; Buckley & Voorhees, 2004; Sakai, 2007; Yilmaz & Aslam, 2006; 2008), finding the best sample of documents to be judged for each topic (Carterette, Allan, & Sitaraman, 2006; Cormack, Palmer, & Clarke, 1998; Jones & van Rijsbergen, 1975; Moffat, Webber, & Zobel, 2007; Pavlu & Aslam, 2007), inferring relevance judgments (Aslam & Yilmaz, 2007), topic selection (Guiver et al., 2009; Hosseini et al., 2012; Hosseini, Cox, Milic-Frayling, Vinay, & Sweeting, 2011; Mizzaro & Robertson, 2007), evaluation with no human judgments (Nuray & Can, 2006; Soboroff, Nicholas, & Cahan, 2001), crowdsourcing (Alonso & Mizzaro, 2009; Grady & Lease, 2010), and others. The reader is referred to (Moghadasi, Ravana, & Raman, 2013) and (Sanderson, 2010) for a more detailed review of prior work on methods for low-cost IR evaluation.

2.1. Topic selection

To the best of our knowledge, Mizzaro and Robertson 2007's study was the first seeking to select the best subset of topics for evaluation. They first built a system-topic graph representing the relationship between topics and IR systems, then ran the HITS algorithm on it. They hypothesized that topics with higher 'hubness' scores would better distinguish between systems. However, Robertson 2011 experimentally showed that their hypothesis was not true.

Guiver et al. 2009 experimentally showed that if we choose the right subset of topics, we can achieve a ranking of systems that is very similar to the ranking when we employ all topics. However they did not provide a solution to find the right subset of topics. This study has motivated other researchers to investigate this problem. Berto, Mizzaro, and Robertson 2013 stressed generality and showed that a carefully selected good subset of topics to evaluate a set of systems can be also adequate to evaluate a different set of systems. Hauff, Hiemstra, Azzopardi, and De Jong 2010 reported that using the easiest topics based on Jensen-Shannon Divergence approach did not work well to reduce the number of topics. Hosseini et al. 2011 focused on selecting the subset of topics to extend an existing collection in order to increase its re-usability. Culpepper, Mizzaro, Sanderson, and Scholer. 2014 investigated how the capability of topics to predict overall system effectiveness has changed over the years in TREC test collections. Kazai and Sung 2014 reduced the cost using dissimilarity based query selection for preference based IR evaluation.

The closest study to our own is (Hosseini et al., 2012), which employs an adaptive algorithm for topic selection. It selects the first topic randomly. Once a topic is selected, the relevance judgments are acquired and used to assist with the selection of subsequent topics. Specifically, in the following iterations, the topic that is predicted to maximize the current Pearson correlation is selected. In order to do that, they predict relevance probabilities of *qrels* for the remaining topics using a Support Vector Machine (SVM) model trained on the judgments from the topics selected thus far. Training data is extended at each iteration by adding the relevance judgments from each topic as it is selected in order to better select the next topic.

Further studies investigated topic selection for other purposes, such as creating low-cost datasets for training learning-to-rank algorithms (Mehrotra & Yilmaz, 2015), system rank estimation (Hauff, Hiemstra, De Jong, & Azzopardi, 2009), and selecting training data to improve supervised data fusion algorithms (Lin & Cheng, 2011). These studies do not consider topic selection for low-cost evaluation of IR systems.

2.2. How many topics are needed?

Past work has investigated the ideal size of test collections and how many topics are needed for a reliable evaluation. While traditional TREC test collections employ 50 topics, a number of researchers claimed that 50 topics are not sufficient for a reliable evaluation (Jones & van Rijsbergen, 1975; Sakai, 2016c; Urbano, Marrero, & Martín, 2013; Voorhees, 2009). Many researchers reported that wide and shallow judging is preferable than narrow and deep judging (Bodoff & Li, 2007; Carterette & Smucker, 2007; Sanderson & Zobel, 2005). Carterette et al. 2008 experimentally compared deep vs. shallow judging in terms of budget utilization. They found that 20 judgments with 250 topics was the most cost-effective in their experiments. Urbano et al. 2013 measured the reliability of TREC test collections with regard to generalization and concluded that the number of topics needed for a reliable evaluation varies across different tasks. Urbano 2016 analyzed different test collection reliability measures with a special focus on the number of topics.

In order to calculate the number of topics required, Webber, Moffat, and Zobel 2008 proposed adding topics iteratively until desired statistical power is reached. Sakai proposed methods based on two-way ANOVA 2014c, confidence interval 2014b, and *t* test and one-way ANOVA 2014a. In his follow-up studies, Sakai investigated the effect of score standardization 2016b in topic set design 2016a and provided guidelines for test collection design for a given fixed budget 2016c. Sakai, Shang, Lu, and Li 2015 applied the method of Sakai 2016c to decide the number of topics for evaluation measures of a Short Text Conversation task[1]. Sakai and Shang 2016 explored how many topics and IR systems are needed for a reliable topic set size estimation. While these studies focused on calculating the number of topics required, our work focuses on how to select the best topic set for a given size in order to maximize the reliability of evaluation. We also investigate further considerations impacting the debate over shallow vs. deep judging: familiarization of users to topics, and the effect of topic development costs on the budget utilization and the quality of judgments for each topic.

Lewandowski 2015 investigated the problem of evaluating commercial search engines by sampling queries based on their distribution in query logs. In contrast, our work does not rely on any prior knowledge about the popularity of topics in performing topic selection.

2.3. Topic familiarity vs. judging speed

Carterette et al. 2009 reported that as the number of judgments per topic increases (when collecting 8, 16, 32, 64 or 128 judgments per topic), the median time to judge each document decreases respectively: 15, 13, 15, 11 and 9 seconds. This suggests that assessors become more familiar with a topic as they judge more documents for it, and this greater familiarity yields greater judging speed. However, prior work comparing deep vs. shallow judging did not consider this, instead assuming that judging speed is constant regardless of judging depth. Consequently, our experiments in Section 5.5 revisit this question, considering how faster

[1] http://ntcir12.noahlab.com.hk/stc.htm

judging with greater judging depth per topic may impact the tradeoff between deep vs. shallow judging in maximizing evaluation reliability for a given budget in human assessor time.

2.4. Topic development cost vs. judging consistency

Past work has utilized a variety of different processes to develop search topics when constructing test collections. These different processes explicitly or implicitly enact potentially important trade-offs between human effort (i.e. cost) vs. quality of the resultant topics developed by each process. For example, NIST has employed a relatively costly process in order to ensure creation of very high quality topics (Voorhees, 2016):

> For the traditional ad hoc tasks, assessors generally came to NIST with some rough ideas for topics having been told the target document collection. For each idea, they would create a query and judge about 100 documents (unless at least 20 of the first 25 were relevant, in which case they would stop at 25 and discard the idea). From the set of candidate topics across all assessors, NIST would select the final test set of 50 based on load-balancing across assessors, number of relevant found, eliminating duplication of subject matter or topic types, etc. The judging was an intrinsic part of the topic development routine because we needed to know that the topic had sufficiently many (but not too many) relevant in the target document set. (These judgments made during the topic development phase were then discarded. Qrels were created based only on the judgments made during the official judgment phase on pooled participant results.) We used a heuristic that expected one out of three original ideas would eventually make it as a test set topic. Creating a set of 50 topics for a newswire ad hoc collection was budgeted at about 175–225 assessor hours, which works out to about 4 hours per final topic.

In contrast, the TREC Million Query (MQ) Track used a rather different procedure to develop topics. In the 2007 MQ Track (Allan et al., 2007), 10,000 queries were sampled from a large search engine query log. The assessment system showed 10 randomly selected queries to each assessor, who then selected one and converted it into a standard TREC topic by back-fitting a topic description and narrative to the selected query. Carterette et al. 2008 reported that the median time of developing a topic was roughly 5 minutes. In the 2008 MQ Track (Allan, Aslam, Carterette, Pavlu, & Kanoulas, 2008), assessors could refresh list of candidate 10 queries if they did not want to judge any of the candidates listed. Carterette et al. 2009 reported that *median* time for viewing a list of queries was 22 seconds and back-fitting a topic description was 76 seconds. On average, each assessor viewed 2.4 lists to develop each topic. Therefore, the cost of developing a topic was roughly $2.4*22 + 76 \approx 129$ seconds, or 2.1 minutes.

The examples above show a vast range of topic creation times: from 4 hours to 2 minutes per topic. Therefore, in Section 5.5, we investigate deep vs. shallow judging when cost of developing topics is also considered.

In addition to considering topic construction time, we might also consider whether aggressive reduction in topic creation time might also have other unintended, negative impacts on topic quality. For example, Scholer et al. 2013 reported calibration effects change judging decisions as assessors familiarize themselves with a topic. Presumably NIST's 4 h topic creation process provides judges ample time to familiarize themselves with a topic, and as noted above, judgments made during the topic development phase are then discarded. In contrast, it seems MQ track assessors began judging almost immediately after selecting a query for which to back-fit a topic, and with no initial topic formation period for establishing the topic and discarding initial topics made during this time. Further empirical evidence suggesting quality concerns with MQ track judgments was also recently reported by McDonnell et al. 2016, who described a detailed judging process they employed to reproduce NIST judgments. While the authors reported high agreement between their own judging and crowd judging vs. NIST on the 2009 Web Track, for NIST judgments from the 2009 MQ track, the authors and crowd judges were both consistent while disagreeing often with NIST judges. The authors also reported that even after detailed analysis of the cases of disagreement, they could not find a rationale for the observed MQ track judgments. Taken in sum, these findings suggest that aggressively reducing topic creation time may negatively impact the quality of judgments collected for that topic. For example, while an assessor is still formulating and clarify a topic for himself/herself, any judgments made at this early stage of topic evolution may not be self-consistent with judgments made once the topic is further crystallized. Consequently, in Section 5.5 we revisit the question of deep judging of few topics vs. shallow judging of many topics, assuming that low topic creation times may also mean less consistent judging.

3. Problem definition

In this section, we define the topic selection problem. We assume that we have a TREC-like setup: a document collection has already been acquired, a large pool of topics and ranked lists of IR systems for each topic are also available. Our goal is to select a certain number of topics from the topic pool such that evaluation with those selected topics yields the most similar ranking of the IR systems to the "*ground-truth*". We assume that the ground-truth ranking of the IR systems is the one when we use all topics in the pool for evaluation.

We can formulate this problem as follows. Let $T = \{t_1, t_2, \ldots, t_N\}$ denote the pool of N topics, $S = \{s_1, s_2, \ldots, s_K\}$ denote the set of K IR systems to be evaluated, and $R_{<S,\, T,\, e>}$ denote the ranking of systems in S when they are evaluated based on evaluation measure e over the topic set T (notation used in equations and algorithms is shown in Table 1). We aim to select a subset $P \subset T$ of M topics that *maximizes* the correlation (as a measure of similarity between two ranked lists) between the ranking of systems over P (i.e., considering only M topics and their corresponding relevance judgments) and the ground-truth ranking of systems (over T). Mathematical definition of our goal is as follows:

Table 1
Notation used in equations and algorithms.

Symbol	Name
T	Topic pool
S	IR systems participated to the pool of the corresponding test collection
$R_{<S,\ T,\ e>}$	Ranking of systems in S when they are evaluated based on evaluation measure e over the topic set T
N	Size of topic pool
M	Number of topics to be selected
D_{t_c}	The document pool for topic t_c
$L_{sj(t_c)}$	The ranked list resulting from system s_j for the topic t_c

$$\max_{P\subset T,\ |P|=M} corr(R_{<S,P,e>}, R_{<S,T,e>}) \tag{1}$$

where *corr* is a ranking similarity function, *e.g.,* Kendall-τ (Kendall, 1938).

4. Proposed approach

The problem we are tackling is challenging since we do not know the actual performance of systems (i.e. their performance when all topics are employed for evaluation) and we would like to find a subset of topics that achieves similar ranking to the *unknown* ground-truth.

To demonstrate the complexity of the problem, let us assume that we obtain the judgments for all topic-document pairs (i.e., we know the ground-truth ranking). In this case, we have to check $\binom{N}{M}$ possibilities of subsets in order to find the optimal one (i.e., the one that produces a ranking that has the maximum correlation with the ground-truth ranking). For example, if $N = 100$ and $M = 50$, we need to check around 10^{29} subsets of topics. Since this is computationally intractable, we need an approximation algorithm to solve this problem. Therefore, we first describe a greedy oracle approach to select the best subset of topics when we already have the judgments for all query-document pairs (Section 4.1). Subsequently, we discuss how we can employ this greedy approach when we do not already have the relevance judgments (Section 4.2). Finally, we introduce our L2R-based topic selection approach (Section 4.3).

4.1. Greedy approach

We first explore a greedy oracle approach that selects topics in an iterative way when relevance judgments are already obtained. Instead of examining all possibilities, at each iteration, we select the ’*best*’ topic (among the currently non-selected ones) that, when added to the currently-selected subset of topics, will produce the ranking that has the maximum correlation with the ground-truth ranking of systems.

Algorithm 1illustrates this oracle greedy approach. First, we initialize set of selected topics (P) and set of candidate topics to be selected ($\overline{P}$) (Line 1). For each candidate topic t in $\overline{P}$, we rank the systems over the selected topics P in addition to t ($R_{<S,\ P\cup\{t\},\ e>}$), and calculate the Kendall’s τ achieved with this ranking (Lines 3–4). We then pick the topic achieving the highest Kendall-τ score among other candidates (Line 5) and update P and $\overline{P}$ accordingly (Lines 6–7). We repeat this process until we reach the targeted subset size M (Lines 2–7).

While this approach has $O(M \times N)$ complexity (which is clearly much more efficient compared to selecting the optimal subset), it is also impractical due to leveraging the real judgments (which we typically do not have in advance) in order to calculate the ground-truth ranking and thereby Kendall-τ scores.

4.2. Performance prediction approach

One possible way to avoid the need for the actual relevance judgments is to predict the performance of IR systems using automatic evaluation methods (Nuray & Can, 2006; Soboroff et al., 2001) and then rank the systems based on their predicted performance. For example, Hosseini et al. 2012 predict relevance probability of document-topic pairs by employing an SVM classifier and select topics in a greedy way similar to Algorithm 1. We use their selection approach as a baseline in our experiments (Section 5).

4.3. Proposed Learning-to-Rank Approach

In this work, we formulate the topic selection problem as a learning-to-rank (L2R) problem. In a typical L2R problem, we are given a query q and a set of documents D, and a model is learned to rank those documents in terms of relevance with respect to q. The model is trained using a set of queries and their corresponding labeled documents. In our context, we are given a set of currently-selected topic set P (analogous to the query q) and the set of candidate topics $\overline{P}$ to be selected from (analogous to the documents D), and we aim to train a model to rank the topics in $\overline{P}$ based on the expected effectiveness of adding each to P. The training samples used to train the model are tuples of the form (P, t, $corr$($R_{<S,\ P\cup\{t\},\ e>}$, $R_{<S,\ T,\ e>}$)), where the measured correlation is used to label topic t with respect to P. Notice that the correlation is computed using the true relevance judgments in the training data. This enables us to

```
$P \leftarrow \emptyset$; $\bar{P} \leftarrow T$                                ▹ selected set $P$ is initially empty
while $|P| < M$ do
    for each topic $t \in \bar{P}$ do
        $\tau_t \leftarrow corr(R_{<S,P \cup \{t\},e>},\ R_{<S,T,e>})$
    topic $t^* \leftarrow \underset{t \in \bar{P}}{max}(\tau_t)$                ▹ choose $t^*$ yielding the best correlation
    $P \leftarrow P \cup \{t^*\}$                                        ▹ add $t^*$ to the selected set $P$
    $\bar{P} \leftarrow \bar{P} - \{t^*\}$
```

Algorithm 1. A Greedy Oracle Approach for Topic Selection.

```
$P \leftarrow \emptyset; \bar{P} \leftarrow T$    ▹ selected set $P$ is initially empty
**while** $|P| < M$ **do**
    **for each** topic $t \in \bar{P}$ **do**
        $v_t \leftarrow f(t, P, \bar{P})$    ▹ compute feature vector for $t$
    $l \leftarrow L2R(\{(t, v_t)\} \mid \forall t \in \bar{P})$    ▹ apply L2R on feature vectors
    topic $t^* \leftarrow first(l)$    ▹ choose $t^*$ in the first rank
    $P \leftarrow P \cup \{t^*\}$    ▹ add $t^*$ to the selected set $P$
    $\bar{P} \leftarrow \bar{P} - \{t^*\}$    ▹ Remove $t^*$ from the candidate set $\bar{P}$
```

Algorithm 2. L2R-based Topic Selection.

Table 2
Topic-based Features.

Feature	Description
$f_{\overline{w}}$	Average sampling weight of documents
f_{σ_w}	Standard deviation of weight of documents
$f_{\overline{\tau}}$	Average τ score for ranked lists pairs
f_{σ_τ}	Standard deviation of τ scores for ranked lists pairs
$f_{\$}$	Judgment cost of the topic
$f_{\sigma_\$}$	Standard deviation of judgment costs of system pairs
$f_{\sigma_{QPP}}$	Standard deviation of estimated performance of systems

use the wealth of existing test collections to acquire data for training our model, as explained in Section 4.3.2.

We apply this L2R problem formulation to the topic selection problem using our greedy approach. We use the trained L2R model to rank the candidate topics and then select the first-ranked one. The algorithm is shown in Algorithm 2. At each iteration, a feature vector v_t is computed for each candidate topic t in $\overline{P}$ using a feature extraction function f (Lines 3–4), detailed in Section 4.3.1. The candidate topics are then ranked using our learned model (Line 5) and the topic in the first rank is picked (Line 6). Finally, the topic sets P and $\overline{P}$ are updated (Lines 7–8) and a new iteration is started, if necessary.

4.3.1. Features

In this section, we describe the features we extract in our L2R approach for each candidate topic. Hosseini et al. 2012 mathematically show that, in the greedy approach, the topic selected at each iteration should be different from the already-selected ones (i.e., topics in P) while being representative of the non-selected ones (i.e., topics in $\overline{P}$). Therefore, the extracted set of features should cover the candidate topic as well as the two sets P and $\overline{P}$. Features should therefore capture the interaction between the topics and the IR systems in addition to the diversity between the IR systems in terms of their retrieval results.

We define two types of feature sets. *Topic-based* features are extracted from an individual topic while *set-based* features are extracted from a set of topics by aggregating the topic-based features extracted from each of those topics.

The topic-based features include 7 features that are extracted for a given candidate topic t_c and are listed in Table 2. For a given set of topics (e.g., currently-selected topics P), we extract the set-based features by computing both average and standard deviation of each of the 7 topic-based features extracted from all topics in the set. This gives us 14 set-based features that can be extracted for a set of topics. We compute these 14 features for each of the following sets of topics:

- currently-selected topics (P)
- not-yet-selected topics ($\overline{P}$)
- selected topics with the candidate topic ($P \cup \{t_c\}$)
- not-selected topics excluding the candidate topic ($\overline{P} - \{t_c\}$)

In total, we have 63 features for each data record representing a candidate topic: $14 \times 4 = 56$ features for the above groups + 7 topic-based features. We now describe the seven topic-based features that are at the core of the feature set.

- **Average sampling weight of documents ($f_{\overline{w}}$)**: In the statAP sampling method (Pavlu & Aslam, 2007), a weight is computed for each document based on where it appears in the ranked lists of all IR systems. Simply, the documents at higher ranks get higher weights. The weights are then used in a non-uniform sampling strategy to sample more documents relevant to the corresponding topic. We compute the average sampling weight of all documents that appear in the pool of the candidate topic t_c as follows:

$$f_{\overline{w}}(t_c) = \frac{1}{|D_{t_c}|} \sum_{d \in D_{t_c}} w(d, S) \tag{2}$$

  where D_{t_c} is the document pool for topic t_c and $w(d, S)$ is the weight of document d over the IR systems S. High $f_{\overline{w}}$ values mean that the systems have common documents at higher ranks for the corresponding topic, whereas lower $f_{\overline{w}}$ values indicate that the systems return significantly different ranked lists or have only the documents at lower ranks in common.
- **Standard deviation of weight of documents (f_{σ_w})**: Similar to $f_{\overline{w}}$, we also compute the standard deviation of the sampling weights of documents for the candidate topic as follows:

$$f_{\sigma_w}(t_c) = \sigma\{w(d, S) \mid \ \forall\ d \in D_{t_c}\} \tag{3}$$

- **Average** τ score for ranked lists pairs ($f_{\overline{\tau}}$): This feature computes Kendall's τ correlation between ranked lists of each pair of the IR systems and then takes the average (as shown in Eq. 4) in order to capture the diversity of the results of the IR systems. The depth of the ranked lists is set to 100. In order to calculate the Kendall's τ score, the documents that appear in one list but not in the other are concatenated to the other list so that both ranked lists contain the same documents. If there are multiple documents to be concatenated, the order of the documents in the ranked list is preserved during concatenation. For instance, if system B returns documents {a,b,c,d} and system R returns {e,a,f,c} for a topic, then the concatenated ranked lists of B and R are {a,b,c,d,e,f} and {e,a,f,c,b,d}, respectively.

$$f_{\bar{\tau}}(t_c) = \frac{1}{2|S| - 1} \sum_{i=1}^{|S|-1} \sum_{j=i+1}^{|S|} corr\left(L_{si(t_c)}, L_{sj(t_c)}\right) \tag{4}$$

where $L_{s_j(t_c)}$ represents the ranked list resulting from system s_j for the topic t_c.

- **Standard deviation of** τ scores for ranked lists pairs (f_{σ_τ}): This feature computes the standard deviation of the τ scores of the pairs of ranked lists as follows:

$$f_{\sigma_\tau}(t_c)=\sigma\{corr\left(L_{s_i(t_c)}, L_{s_j(t_c)}\right) \mid \ \forall\, i, j \le |S|,\, i \ne j\} \tag{5}$$

- **Judgment cost of the topic ($f_\$$):** This feature estimates the cost of judging the candidate topic as the number of documents in the pool at a certain depth. If IR systems return many different documents, then the judging cost increases; otherwise, it decreases due to having many documents in common. We set pool depth to 100 and normalize costs by dividing by the maximum possible cost (i.e., 100 × $|S|$).

$$f_\$(t_c) = \frac{D_{t_c}}{|S| \times 100} \tag{6}$$

- **Standard deviation of judgment costs of system pairs ($f_{\sigma_\$}$):** The judgment cost depends on systems participating in the pool. We construct the pool separately for each pair of systems and compute the standard deviation of the judgment cost across pools as follows:

$$f_{\sigma_\$}(t_c) = \sigma\{|L_{s_i(t_c)} \cup L_{s_j(t_c)}| \mid \ \forall\, i, j \le |S|,\, i \ne j\} \tag{7}$$

- **Standard deviation of estimated performance of systems ($f_{\sigma_{QPP}}$):** We finally compute standard deviation of the estimated performances of the IR systems for the topic t_c using a query performance predictor (QPP) (Cummins, Jose, & O'Riordan, 2011). The QPP is typically used to estimate the performance of a single system and is affected by the range of retrieval scores of retrieved documents. Therefore, we normalize the document scores using min-max normalization before computing the predictor.

$$f_{\sigma_{QPP}}(t_c)=\sigma\{QPP(s_i,\, t_c) \mid \ \forall\, i \le |S|\} \tag{8}$$

where $QPP(s_i,\ t_c)$ is the performance predictor applied on system s_i given topic t_c.

4.3.2. Generating training data

Our proposed L2R approach ranks topics based on their effectiveness when added to some currently-selected set of topics. This makes creating the training data for the model a challenging task. First, there are countless number of possible scenarios (i.e., different combinations of topic sets) that we can encounter during the topic selection process. Second, the training data should specify which topic is more preferable for a given scenario.

We developed a method to generate training data by leveraging existing test collections for which we have both relevance judgments and document rankings from several IR systems (e.g., TREC test collections). We first simulate a scenario in which a subset of topics has already been selected. We then rank the rest of the topics based on the correlation with the ground-truth ranking when each topic is added to the currently-selected subset of topics. We repeat this process multiple times and vary the number of already-selected topics in order to generate more diverse training data. The algorithm for generating training data from one test collection is given in Algorithm 3. The algorithm could also be applied to several test collections in order to generate larger training data.

The algorithm first determines the ground-truth ranking of IR systems using all topics in the test collection (Line 1). It then starts the process of generating the data records for each possible topic subset size for the targeted test collection (Line 2). For each subset size i, we repeat the following procedure W times (Line 3); in each, we randomly select i topics, assuming that these represent the currently-selected subset of topics P (Line 4). For each topic t of the non-selected topics $\overline{P}$, we rank the systems in case we add t to P and calculate the Kendall's τ score achieved in that case (Lines 6–9). This gives us how effective each of the candidate topics would be in the IR evaluation for this specific scenario (*i.e.*, when those i topics are already selected). This also allows us to make a comparison between topics and rank them in terms of their effectiveness. In order to generate labels that can be used in L2R methods, we map each τ score to a value within a certain range. We first divide the range between maximum and minimum τ scores into K equal bins and then assign each topic to its corresponding bin based on its effectiveness. For example, let $K = 10$,$T_{\max} = 0.9$, and $T_{\min} = 0.7$. The τ ranges for labeling will be $0 = [0.7 - 0.72)$,$1 = [0.72 - 0.74)$, ... , $9 = [0.88 - 0.9]$. Topics are then labeled from 0 to $(K - 1)$ based on their assigned bin. For example, if we achieve $\tau = 0.73$ score for a particular topic, then the label for the corresponding data record will be 1. Finally, we compute the feature vector for each topic, assign the labels, and output the data records for the current scenario (Lines 10–13). We repeat this process W times (Line 3) to capture more diverse scenarios for the given topic subset size. We can further increase the size of the generated training data by applying the algorithm on different test collections and merging the resulting data.

```
$R^* \leftarrow R_{<S,T,e>}$                                   ▹ Determine the ground-truth ranking of systems
for i = 0 to $|T|$ - 2 do                                      ▹ Vary the size of already-selected topic set
    for j = 1 to W do                                          ▹ Repeat the process $W$ times
        $P \leftarrow random(T, i)$                            ▹ Randomly select $i$ topics
        $\bar{P} \leftarrow T - P$
        for each topic $t \in \bar{P}$ do
            $R_t \leftarrow R_{<S,P \cup t,e>}$
            $\tau_t \leftarrow corr(R_t, R^*)$
        $\tau_{\max} \leftarrow \underset{t \in \bar{P}}{max}(\tau_t)$; $\tau_{\min} \leftarrow \underset{t \in \bar{P}}{min}(\tau_t)$
        for each topic $t \in \bar{P}$ do
            $v_t \leftarrow f(t, P, \bar{P})$                  ▹ Compute feature vector for $t$
            $l_t \leftarrow \lfloor \frac{K*(\tau_t - \tau_{\min})}{(\tau_{\max} - \tau_{\min})} \rfloor$        ▹ Find which partition $\tau_t$ is in.
            Output $< l_t, v_t >$
```

Algorithm 3. Generating training data from one test collection of T topics and S IR systems.

Table 3
Test Collections Used in Experiments.

Collection	Topics	Dataset	Runs	Judgments	# Relevant
TREC-9 Web Track	451–500	WT10g	104	70070	2617
TREC-2001 Web Track	501–550	WT10g	97	70400	3363
Robust2003	601–650, 50 difficult topics from 301 to 450	TREC disks 4&5 minus Congressional Records	78	128796	6074
Robust2004	301–450, 601–700	TREC disks 4&5 minus Congressional Records	110	311410	17412

5. Evaluation

In this section, we evaluate our proposed L2R topic selection approach with respect to our research questions and baseline methods. Section 5.1 details our experimental setup, including generation of training data and tuning of our L2R model parameters. We present results of our topic selection experiments (RQ-1) in Section 5.2. We report ablation analysis of our features in Section 5.3 and discuss the evaluation of the parameters of our approach in Section 5.4. In Section 5.5, we report the results of our experiments for intelligent topic selection with a fixed budget (RQ-2) and considering different parameters in the debate of NaD vs. WaS judging: varying judging speed (RQ-3), topic generation time (RQ-4), and judging error (RQ-5).

5.1. Setup

We adopt the MART (Friedman, 2001) implementation in the RankLib library[2] as our L2R model[3]. To tune MART parameters, we partition our data into disjoint training, tuning, and testing sets. We assume that the ground-truth ranking of systems is given by MAP@100.

Test Collections. We consider two primary criteria in selecting test collections to use in our experiments: (1) the collection should contain many topics, providing a fertile testbed for topic selection experimentation, and (2) the set of topics used in training, tuning, and testing should be disjoint to avoid over-fitting. To satisfy these criteria, we adopt the TREC-9 (Hawking, 2000) and TREC-2001 (Hawking & Craswell, 2002) Web Track collections, as well as TREC-2003 (Voorhees, 2003) and TREC-2004 (Voorhees, 2004) Robust Track collections. Details of these test collections are presented in Table 3. Note that all four collections target ad-hoc retrieval. We use TREC-9 and TREC-2001 test collections to generate our training data.

Robust2003 and Robust2004 collections are particularly well-suited to topic selection experimentation since they have relatively more topics (100 and 249, respectively) than many other TREC collections. However, because topics of Robust2003 were repeated in Robust2004, we define a new test collection subset which excludes all Robust2003 topics from Robust2004, referring to this subset as Robust2004$_{149}$. We use Robust2003 and Robust2004$_{149}$ collections for tuning and testing. When testing on Robust2003, we tune parameters on Robust2004$_{149}$, unless otherwise noted. Similarly, when testing on Robust2004$_{149}$, we tune parameters on Robust2003.

Generation of Training Data. We generate 100K data records for each topic set size from 0 to 49 (i.e., $N = 50$ and $W = 100K$ in Algorithm 3) for TREC-9 and TREC-2001 and remove the duplicates. The label range is set to 0–49 (i.e., $K = 50$ in Algorithm 3) since each of TREC-9 and TREC-2001 has 50 topics. We merge the data records generated from each test collection to form our final training data. We use this training data in our experiments unless otherwise stated.

Parameter Tuning. To tune parameters of MART, we fix the number of trees to 50 and vary the number of leaves from 2 to 50 with a step-size of 2. For each of those 25 considered configurations, we build a L2R model and select 50 topics (the standard number of topics in TREC collections) using the tuning set. At each iteration of the topic selection process, we rank the systems based on the topics selected thus far and calculate Kendall's τ rank correlation vs. the ground-truth system ranking. Finally, we selected the parameter configuration which achieves the highest average τ score while selecting the first 50 topics.

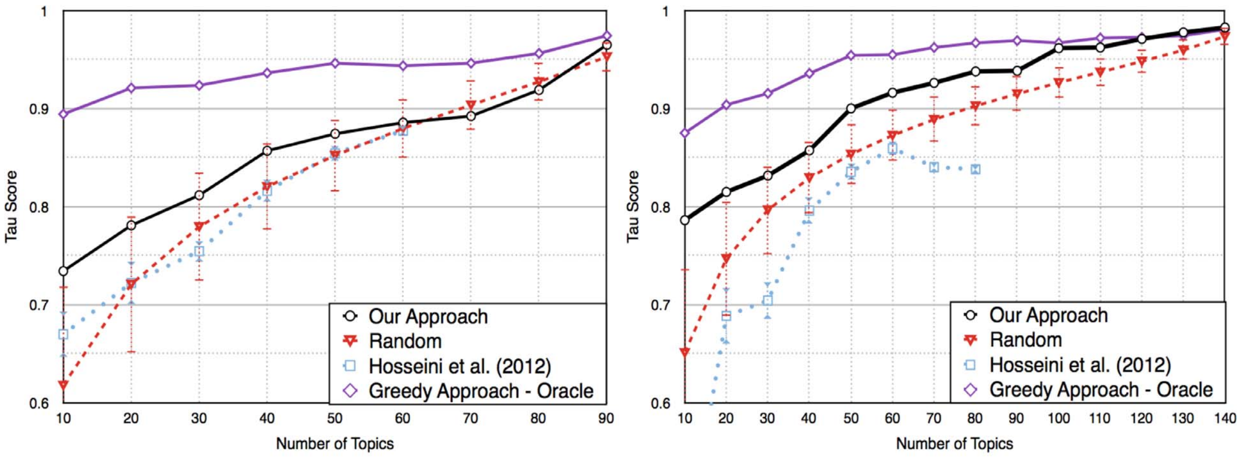
Evaluation Metrics. We adopt MAP@100 and statAP (Pavlu & Aslam, 2007) in order to measure the effectiveness of IR systems. In computing MAP, we use the full pool of judgments for each selected topic. In computing statAP, the number of sampled documents varies in each experiment and are reported in the corresponding sections. Because statAP is stochastic, we repeat the sampling 20 times and report average results.

Baselines. We compare our approach to two baselines:

- **Baseline 1: Random**. For a given topic subset size M, we randomly select topics R times and calculate the average Kendall's τ score achieved over the R trials. We set R to 10K for MAP and 1K for statAP (due to its higher computation cost than MAP).
- **Baseline 2:** Hosseini et al. 2012. We implemented their method using WEKA library (Hall et al., 2009) since no implementation is available from the authors. The authors do not specify parameters used in their linear SVM model, so we adopt default parameters of WEKA's sequential minimal optimization implementation for linear SVMs. Due to its stochastic nature, we run it 50 times and report the average performance.

[2] https://sourceforge.net/p/lemur/wiki/RankLib/

[3] We also focused on other L2R models but MART yielded the best results in our initial experiments when developing our method.



(a) Robust2003

(b) Robust2004$_{149}$

Fig. 1. Selecting a fixed number of topics, using full pool judgments per topic, and evaluating with MAP.

In addition to these two baselines, we also compare our approach to the **greedy oracle approach** defined in Section 4.1 (See Algorithm 1). This serves as a useful oracle upper-bound, since in practice we would only collect judgments for a topic after it was selected.

5.2. Selecting a fixed number of topics

In our first set of experiments, we evaluate our proposed L2R topic selection approach vs. baselines in terms of Kendall's τ rank correlation achieved as a function of number of topics (**RQ-1**). We assume the full pool of judgments are collected for each selected topic and evaluate with MAP.

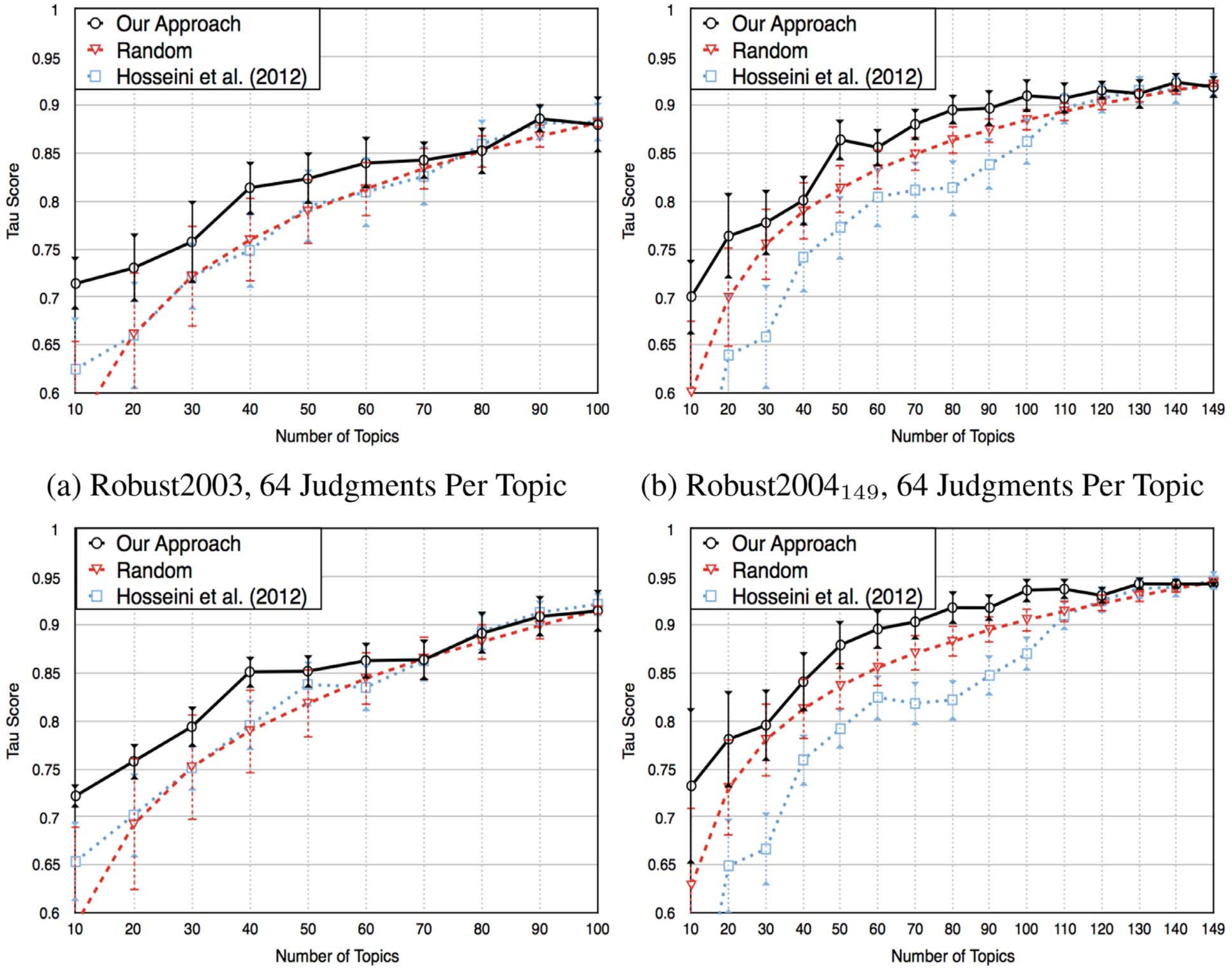
Fig. 1shows results on Robust2003 and Robust2004$_{149}$ collections. Given the computational complexity of Hosseini et al. 2012's method, which re-trains the classifier at each iteration, we could only select 63 topics for Robust2003 and 77 topics for Robust2004$_{149}$ after 2 days of execution[4], so its plots terminate early. The upper-bound *Greedy Oracle* is seen to achieve 0.90 τ score (a traditionally-accepted threshold for acceptable correlation (Voorhees, 2000)) with only 12 topics in Robust2003 and 20 topics in Robust2004$_{149}$. Our proposed L2R method strictly outperforms baselines for Robust2004$_{149}$ and outperforms baselines for Robust2003 except when 70% and 80% of topics are selected. Relative improvement over baselines is seen to increase as the number of topics is reduced. This suggests that our L2R method becomes more effective as either fewer topics are used, or as more topics are available to choose between when selecting a fixed number of topics.

In our next experiment, instead of assuming the full document pool is judged for each selected topic, we consider a more parsimonious judging condition in which statAP is used to select only 64 or 128 documents to be judged for each selected topic. The average τ scores for each method are shown in Fig. 2. The vertical bars represent the standard deviation across trials. Overall, similar to the first set of experiments, our approach outperforms the baselines in almost all cases and becomes more preferable as the number of selected topics decreases. Similar to the previous experiment with full pooling, our L2R approach performs relatively weakest on Robust 2003 when 70 or 80 topics are selected. In this case, our L2R approach is comparable to random selection (with slight increase over it), whereas with the previous experiment we performed slightly worse than random for 70 or 80 topics on this collection.

We were surprised to see Hosseini et al. 2012's topic selection method performing worse than random in our experiments, contrary to their reported results. Consequently, we investigated this in great detail. In comparing results of our respective random baselines, we noted that our own random baseline performed $\tau \approx 0.12$ better on average than theirs over the 20 results they report (using 10, 20, 30, ... , 200 topics), despite our carefully following their reported procedure for implementing the baseline. To further investigate this discrepancy in baseline performance, we also ran our random baseline on TREC-8 and compared our results with those reported by Guiver et al. 2009. Our results were quite similar to Guiver et al.'s. Hosseini et al. 2012 kindly discussed the issue with us, and the best explanation we could find was that they took, "special care when considering runs from the same participant", so perhaps different preprocessing of participant runs between our two studies may contribute to this empirical discrepancy.

Overall, our approach outperforms the baselines in almost all cases demonstrated over two test collections. While the baseline methods do not require any existing test collections for training, the existing wealth of test collections produced by TREC and other shared task campaigns make our method feasible. Moreover, our experiments show that we can leverage existing test collections in building models that are useful for constructing other test collections. This suggests that there are common characteristics across different test collections that can be leveraged even in other scenarios that are out of the scope of this work, such as the prediction of system rankings in a test collection using other test collections.

[4] Two days is the time limit for executing programs on the computing cluster we used for experiments.



(a) Robust2003, 64 Judgments Per Topic

(b) Robust2004$_{149}$, 64 Judgments Per Topic

Fig. 2. Selecting a fixed number of topics and a limited number of documents to judge per topic (using StatAP).

5.3. Feature ablation analysis

In this experiment, we conduct a feature ablation analysis to study the impact of each core feature and also each group of features on the performance of our approach.

We divide our feature set into mutually-exclusive subsets in two ways: core-feature-based subsets, and topic-group-based subsets. Each of the core-feature-based subsets consists of all features related to one of our 7 core features (defined in Table 2). That yields 9 features in each of these subsets; we denote each of them by $\{f\}$, where f represents a core feature. In the other way, we define 5 groups of the topics: the candidate topic t_c (which has 7 core features) and four other groups of topics defined in Section 4.3.1 (each has a subset of features using average and standard deviation of the 7 core features, yielding a total of 14 features). We denote each of these feature subsets by $F(g)$, where g represents a group of topics.

In our ablation analysis, we apply leave-one-subset-out method in which we exclude one subset of the features at a time and follow the same experimental procedure with the previous experiments using the remaining features. We evaluate the effectiveness of systems using MAP. For each subset of features, we report the average Kendall's τ correlation over all possible topic set sizes (1 to 100 for Robust2003 and 1 to 149 for Robust2004$_{149}$) to see its effect on the performance. The results are shown in Table 4.

The table shows four interesting observations. First, $\{f_{\sigma_\$}\}$ and $\{f_{\sigma_w}\}$ are the most effective among the core-feature-based subsets, while $F(P \cup \{t_c\})$ and $F(t_c)$ are the most effective among the topic-group-based subsets, when testing on Robust2003 and Robust2004$_{149}$ respectively. Second, $\{f_{\sigma_\tau}\}$ has the least impact in both test collections. Third, the feature subset of the candidate topic $F(P \cup \{t_c\})$ is the best on overage over all subsets, which is expected as it solely focuses on the topic we are considering to add to the currently-selected topics. Finally, testing on both test collections, we achieve the best performance when we use *all* features.

5.4. Robustness and parameter sensitivity

The next set of experiments we report assess our L2R method's effectiveness across different training datasets and

Table 4
Feature ablation analysis. The percentages in parenthesizes show how much the performance is decreased by removing the corresponding subset of features.

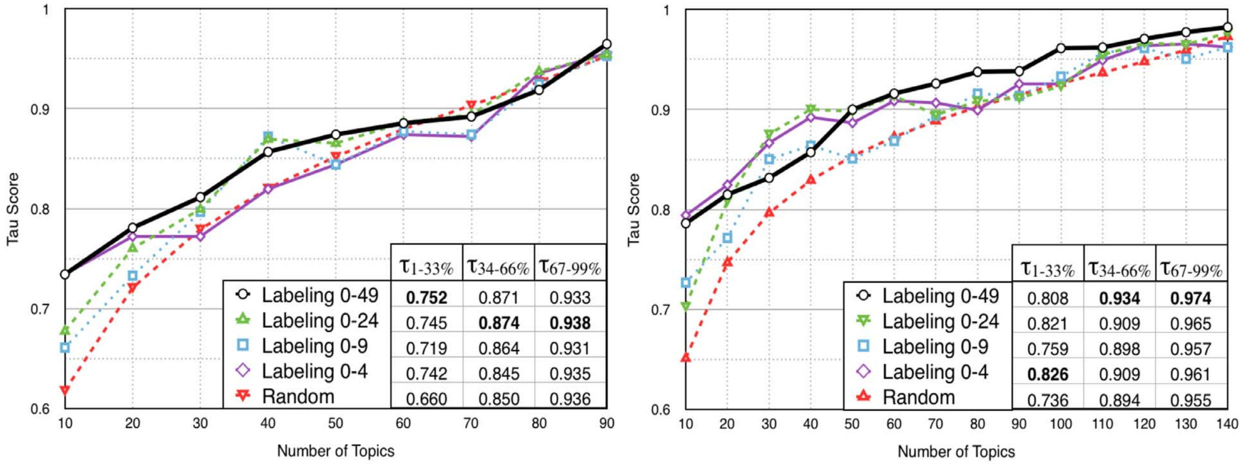
Feature Set	Number of Features	Average Kendall's τ	
		Robust2003	Robust2004$_{149}$
All	63	0.8535	0.9047
All - $\{f_{\overline{w}}\}$	54	0.8339 (-2%)	0.8958 (-1%)
All - $\{f_{\sigma_w}\}$	54	0.8441 (-1%)	0.8717 (-4%)
All - $\{f_{\overline{\tau}}\}$	54	0.8423 (-1%)	0.8796 (-3%)
All - $\{f_{\sigma_\tau}\}$	54	0.8465 (-1%)	0.9003 (-0.5%)
All - $\{f_{\$}\}$	54	0.8351 (-2%)	0.8829 (-2%)
All - $\{f_{\sigma_\$}\}$	54	0.7929 (-7%)	0.8834 (-2%)
All - $\{f_{\sigma_{QPP}}\}$	54	0.8118 (-5%)	0.8852 (-2%)
All - $F(t_c)$	56	0.7969 (-7%)	0.8485 (-6%)
All - $F(P)$	49	0.8310 (-3%)	0.8743 (-3%)
All - $F(\overline{P})$	49	0.8297 (-3%)	0.8882 (-2%)
All - $F(P \cup \{t_c\})$	49	0.7873 (-8%)	0.9017 (-0.3%)
All - $F(\overline{P} - \{t_c\})$	49	0.8432 (-1%)	0.8897 (-2%)

parameterizations. We evaluate the effectiveness of systems using MAP. In addition to presenting results for all topics, we also compute the average τ score over 3 equal-sized partitions of the topics. For example, in Robust2004, we calculate the average τ scores for each of the following partitions: 1–50 (denoted by $\tau_{1-33\%}$), 51–100 (denoted by $\tau_{34-66\%}$) and 101–149 (denoted by $\tau_{67-100\%}$). These results are presented in a table within each figure.

Effect of Label Range in Training Set: As explained in Section 4.3.2, we can assign labels to data records in various ranges. In this experiment, we vary the label range parameter (K in Line 12 of Algorithm 3) and compare the performance of our approach with the corresponding training data on Robust2003 and Robust2004$_{149}$ test collections. The results are shown in Fig. 3. It is hard to draw a clear conclusion since each labeling range has varying performances in different cases. For instance, when we use 5 labels only (i.e., Labeling 0–4), it has very good performance with few selected topics. As the number of topics increases, its performance becomes very close to the random approach. Considering the results in Robust2003 and Robust2004$_{149}$ together, using 50 labels (i.e., labeling 0–49) gives more consistent and better results than others. Using 25 labels are better than using 10 or 5 labels, in general. Therefore, we observe that *using fine grained labels yields better results with our L2R approach.*

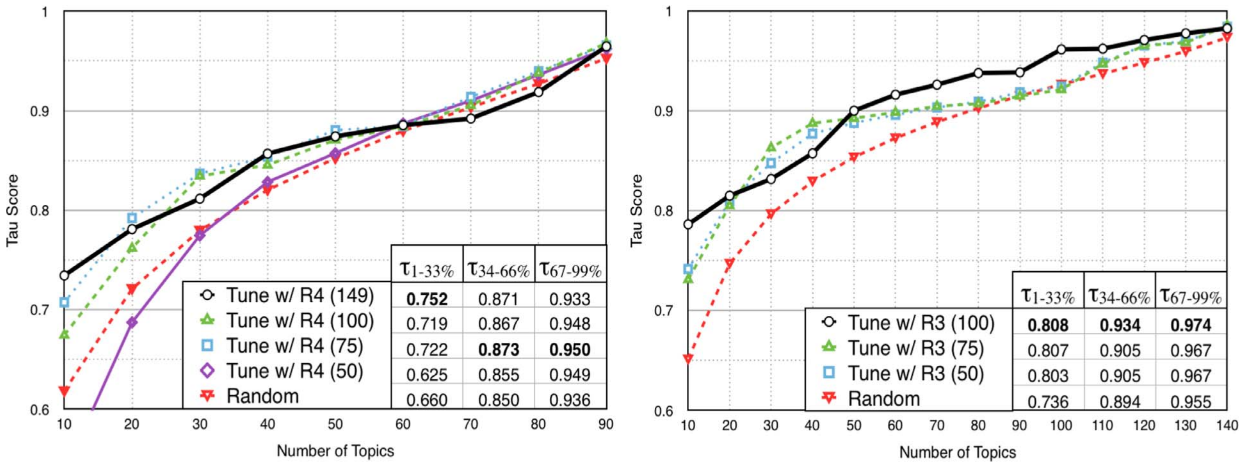
Effect of Size of *Tuning* Dataset: In this experiment, we evaluate how robust our approach is to having fewer topics available for tuning. For this experiment, we randomly select 50 and 75 topics from Robust2003 and remove the not-selected ones from the test collection. We refer to these reduced tuning sets as R3(50) and R3(75). We use these reduced tuning sets for testing on Robust2004$_{149}$. When we test on Robust2003, we perform a similar approach. That is, we randomly select 50, 75, and 100 topics from Robust2004$_{149}$ and follow the same procedure. We repeat this process 5 times and calculate the average τ score achieved.

The results are presented in Fig. 4. The vertical bars represent the standard deviation across 5 different trials. As expected, over Robust2004$_{149}$, we achieve the best performance when we tune with all 100 topics (i.e., actual Robust2003); employing 75 topics is slightly better than employing 50 topics. Over Robust2003, when the number of selected topics is $\leq$ 33% of the whole topic pool



(a) Robust2003 (b) Robust2004$_{149}$

Fig. 3. Effect of label ranges in training data for our learning-to-rank model.



	τ1-33%	τ34-66%	τ67-99%
Tune w/ R4 (149)	**0.752**	0.871	0.933
Tune w/ R4 (100)	0.719	0.867	0.948
Tune w/ R4 (75)	0.722	**0.873**	**0.950**
Tune w/ R4 (50)	0.625	0.855	0.949
Random	0.660	0.850	0.936

(a) Robust2003

	τ1-33%	τ34-66%	τ67-99%
Tune w/ R3 (100)	**0.808**	**0.934**	**0.974**
Tune w/ R3 (75)	0.807	0.905	0.967
Tune w/ R3 (50)	0.803	0.905	0.967
Random	0.736	0.894	0.955

(b) Robust2004$_{149}$

Fig. 4. Effect of test collections used for parameter tuning.

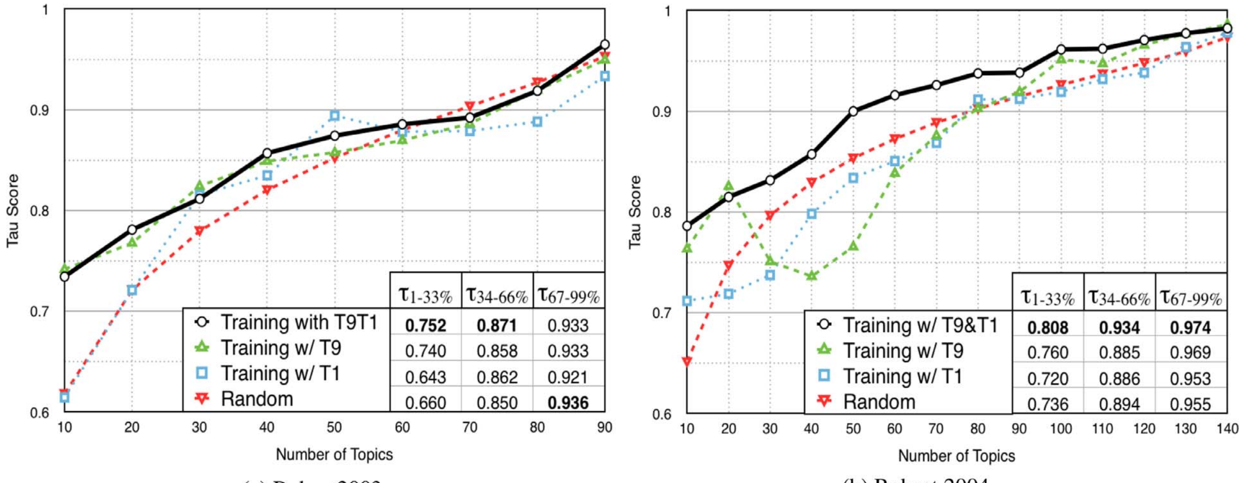
size, tuning with 149 topics gives the best results. For the rest of the cases, tuning with 75 topics gives slightly better results than others. As expected, tuning with only 50 topics yields the worst results in general. Intuitively, *using test collections with more tuning topics is seen to yield better results.*

Effect of Test Collections Used in *Training* : In this experiment, we fix the training data set size, but vary the test collections used for generating the training data. For the experiments so far, we had generated 100K data records for each topic set size from 0 to 49 with TREC-9 and TREC-2001 and subsequently combined both (yielding 200K records in total). In this experiment, in addition to this training data, we generate 200K data records for each topic set size from 0 to 49 with TREC-9 and TREC-2001, and use them separately. That is, we have 3 different datasets (namely, T9&T1, T9 and T1) and each dataset has roughly the same number of data records. The results are shown in Fig. 5. As expected, using more test collections leads to better and more consistent results. Therefore, instead of simply generating more data records from the same test collection, *diversifying the test collections in present in the training data appears to increase our L2R method's effectiveness*.

5.5. Topic selection with a fixed budget

Next, we seek to compare narrow and deep (NaD) vs. wide and shallow (WaS) judging when topics are selected intelligently (**RQ-2**), considering also familiarization of assessors to topics (**RQ-3**), the effect of topic generation cost (**RQ-4**) and judging error (**RQ-5**). We evaluate the performance of the methods using statAP (Pavlu & Aslam, 2007). The budget is distributed equally among topics. In each experiment, we exhaust the full budget for the selected topics, i.e., as the number of topics increases, the number of judgments per topic decreases, and vice-versa.

Effect of Familiarization to Topics when Judging: As discussed in Section 2.3, Carterette et al. 2009 found that as the number



	τ1-33%	τ34-66%	τ67-99%
Training with T9T1	**0.752**	**0.871**	0.933
Training w/ T9	0.740	0.858	0.933
Training w/ T1	0.643	0.862	0.921
Random	0.660	0.850	**0.936**

(a) Robust2003

	τ1-33%	τ34-66%	τ67-99%
Training w/ T9&T1	**0.808**	**0.934**	**0.974**
Training w/ T9	0.760	0.885	0.969
Training w/ T1	0.720	0.886	0.953
Random	0.736	0.894	0.955

(b) Robust 2004$_{149}$

Fig. 5. Effect of test collections used for generating training data.

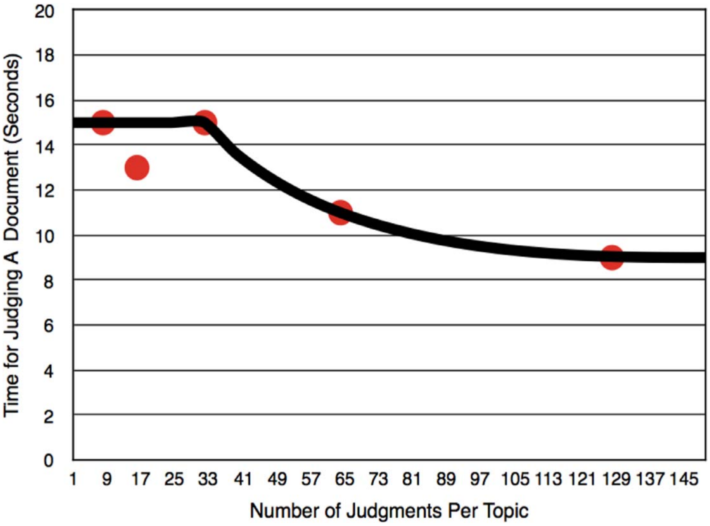


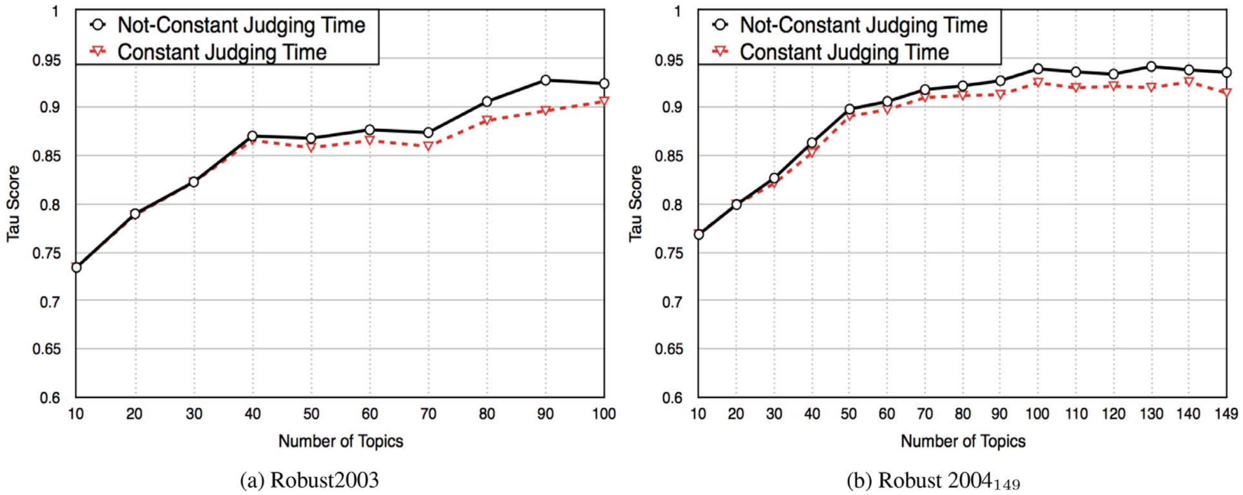
Fig. 6. Illustration of Eq. 9 modeling the judging time per document as a function of judgments per topic, as fit to actual data points reported by (Carterette et al., 2009).

of judgments per topic increases (when collecting 8, 16, 32, 64 or 128 judgments per topic), the median time to judge each document decreases respectively: 15, 13, 15, 11 and 9 seconds. Because prior work comparing NaD vs. WaS judging did not consider variable judging speed as a function of topic depth, we revisit this question, considering how faster judging with greater judging depth per topic may impact the tradeoff between deep vs. shallow judging in maximizing evaluation reliability for a given assessment time budget.

Using Carterette et al. 2009's data points, we fit a piece-wise judging speed function (Eq. 9) to simulate judging speed as a function of judging depth as illustrated in Fig. 6. According to this model, judging a single document takes 15 seconds if there are 32 or fewer judgments per topic (i.e., as the assessor "warms up"). After 32 judgments, the assessors become familiar with the topic and start judging faster. Because judging speed cannot increase forever, we assume that after 128 judgments, judging speed becomes stable at 9 seconds per judgment.

$$f(x) = \begin{cases} 15, & \text{if } x \le 32 \\ 8.761 + 16.856 \times e^{-0.0316 \times x}, & \text{if } 32 < x < 127 \\ 9, & \text{otherwise} \end{cases} \tag{9}$$

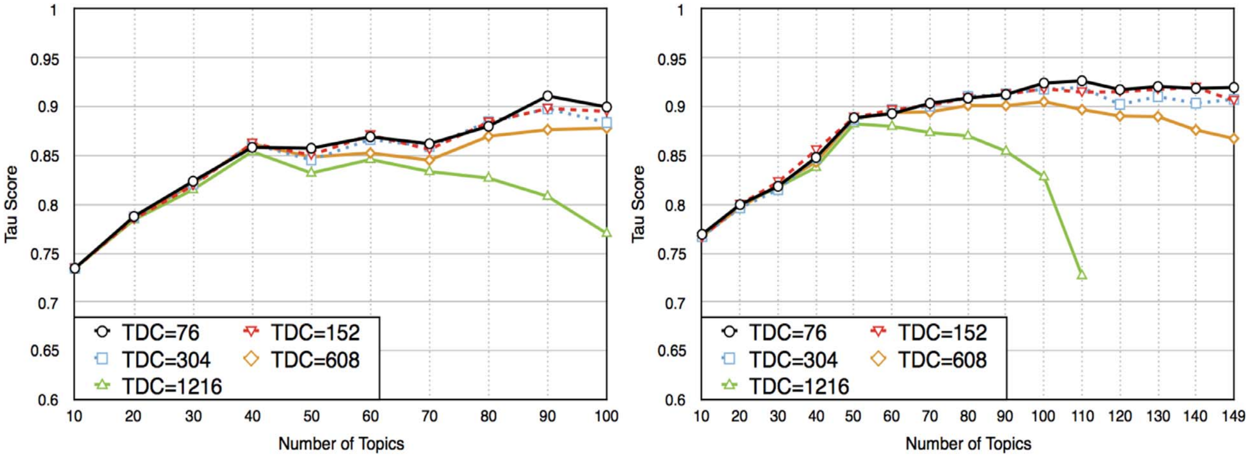
For the constant judging case, we set the judging speed to 15 seconds per document. For instance, if our total budget is 100 hours and we have 100 topics, then we spend 1 h per topic. If judging speed is constant, we judge $60 \text{ min} \times 60\frac{sec}{min} \div 15\frac{seconds}{judgment} = 240$ *judgments* for each topic. However, if judging speed increases according to our model, we can judge a larger set of 400 documents per topic in the same one hour.



(a) Robust2003

(b) Robust 2004_{149}

Fig. 7. Constant vs. Non-Constant Judging Speed. Topic selection is performed with our method. For each case, we run 20 times and report average τ score.



(a) Robust 2003

(b) Robust 2004_{149}

Fig. 8. Effect of Topic Generation Cost (times listed in seconds). For all cases, we apply statAP and sample the documents 20 times. We assume that each judgment requires 15 seconds. We use our method to select the topics.

We set our budget to 40 hours for both test collections. We initially assume that developing the topics has no cost. Results are shown in Fig. 7. We can see that the additional judgments due to faster judging results in a higher τ score after 30 topics, and its effect increases as the number of topics increases. Since the results are significantly different (p value of paired t-test is 0.0075 and 0.0001 in experiments with Robust2003 and Robust2004_{149}, respectively), results suggest that *topic familiarity (i.e., judging speed) impacts optimization of evaluation budget for deep vs. shallow judging.*

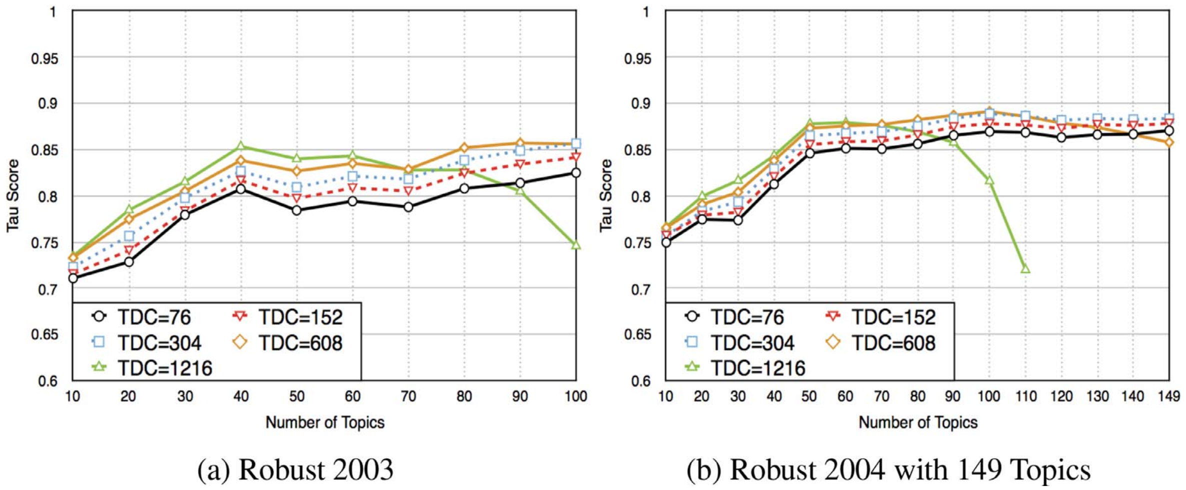
Effect of Topic Development Cost: As discussed in Section 2, topic development cost (TDC) can vary significantly. In order to understand TDC's effect, we perform intelligent topic selection with our L2R method and vary TDC from 76 seconds (i.e., time needed to convert a query to topic, as reported in (Carterette et al., 2009)) to 2432 seconds (i.e., 32 × 76) in geometric order while fixing the budget to 40 hours for both test collections. Note that TREC spends 4 hours to develop a final topic (Voorhees, 2016), which is almost 5 times more than 2432 seconds. We assume that judging speed is constant (i.e., 15 seconds per judgment). For instance, if TDC is 76 seconds and we select 50 topics, then we subtract $50 \times 76 = 3800$ seconds from the total budget and use the remaining for document judging.

The results are shown in Fig. 8. When the topic development cost is ≤ 152 seconds, results are fairly similar. However, when we spend more time on the topic development, after selecting a number of topics, τ scores achieved start decreasing due to insufficient budget left for judging the documents. In Robust2004_{149}, the total budget is not sufficient to generate more than 118 topics when TDC is 1216 seconds. Therefore, no judgment can be collected when the number of topics is 120 or higher. Considering the results for Robust2004_{149} test collection, when TGC is 304 seconds (which is close to that mentioned in (Carterette et al., 2008)), we are able to achieve better performance with 80–110 topics instead of employing all topics. When TDC is 608 seconds, using only 50 topics achieves higher τ score than employing all topics (0.888 vs. 0.868). Overall, the results suggest that *as the topic development cost increases, NaD judging becomes more cost-effective than WaS judging.*

Another effect of topic development cost can be observed in the *reliability of the judgments*, as discussed in Section 2. In this experiment, we consider a scenario where the assessors rely on the topic definitions and poorly-defined topics can cause inconsistent judgments. In order to simulate this scenario, we assume that 8% of judgments are inconsistent when $TDC = 76$ seconds. The accuracy of judgments increases by 2% as TDC doubles. So when TDC = 1216 seconds, assessors can achieve perfect judgments. Note that the assessors in this scenario are much more reliable than what is reported in (McDonnell et al., 2016). In order to implement this scenario, we randomly flip over judgments of *qrels* based on the corresponding accuracy of judging. The ground-truth rankings of the systems are based on the original judgments. We set the total budget to 40 hours and assume that judging a single document takes 15 seconds. We use our method to select the topics. We repeat the process 50 times and report the average.

The results are shown in Fig. 9. In Robust 2003, the achieved Kendall's τ scores increase as TDC increases from 76 to 608 seconds due to more consistent judging (opposite of what we observe in Fig. 8). In Robust2004_{149}, when the number of topics is 100 or less, Kendall's τ score increases as TDC increases from 76 to 608. However, when the number of topics is more than 100, τ scores achieved with $TDC = 608$ start decreasing due to insufficient amount of budget for judging. We observe a similar pattern in Robust2003 with $TDC = 1216$. We achieve the highest τ scores with $TDC = 1216$ and 60 or fewer topics, but the performance starts decreasing later on as more topics are selected. In general, the results suggest that poorly-defined topics should be avoided if they have negative effect on the consistency of the relevance judgments. However, spending more time to develop high quality topics can significantly increase the cost. Therefore, *NaD becomes preferable over WaS when we target constructing high quality topics*.

Varying budget: In this set of experiments, we vary our total budget from 20 to 40 hours. We assume that the assessors judge faster as they judge more documents, up to a point, based on our model given in Eq. 9. We also assume that topic development cost is 76 seconds.



(a) Robust 2003

(b) Robust 2004 with 149 Topics

Fig. 9. Effect of topic development cost with imperfect judgments. For all cases, we apply statAP and sample the documents 20 times. We repeat the process 50 times and report the average overall. We assume that each judgment requires 15 seconds. We use our method to select the topics.

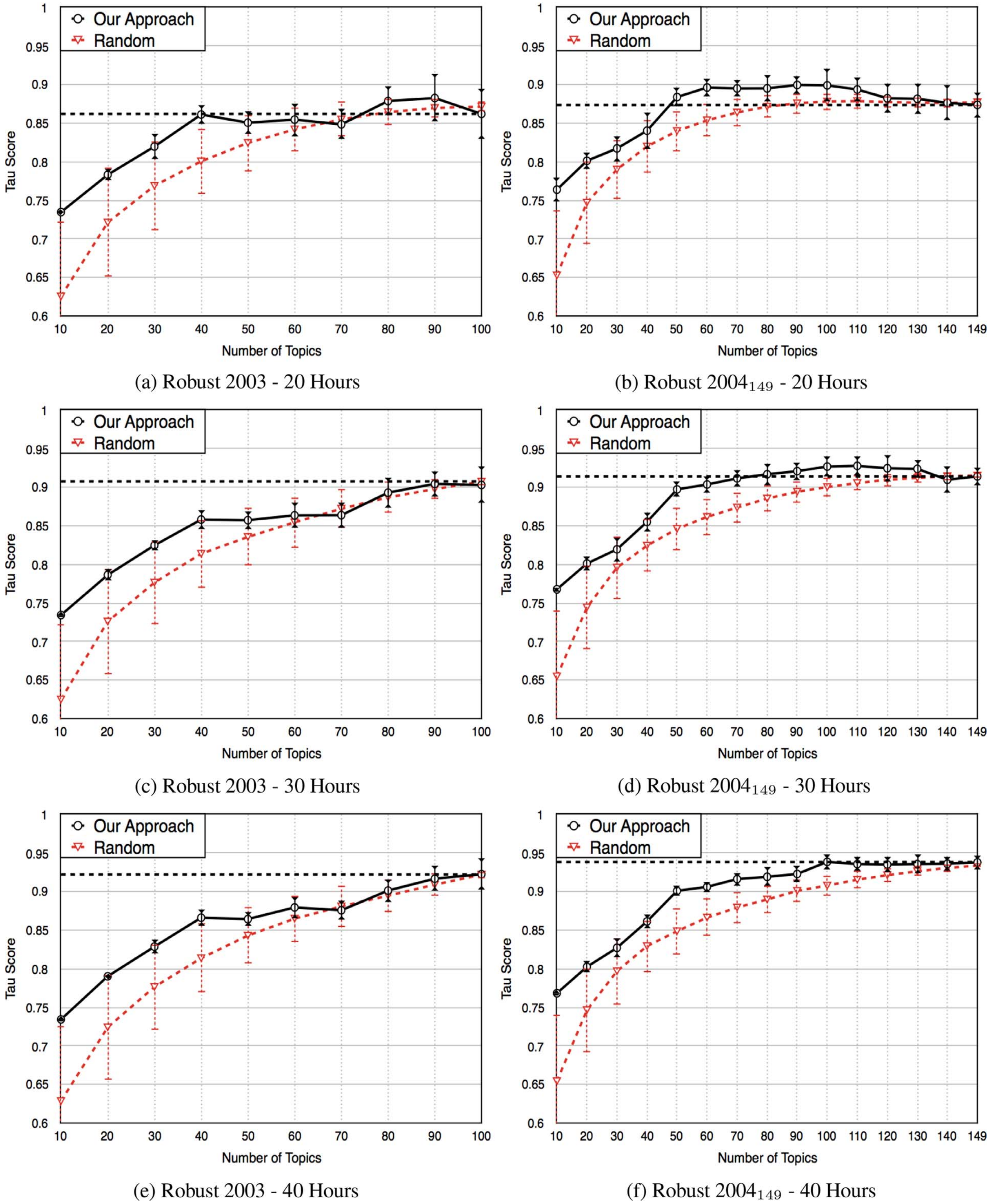
The results are shown in Fig. 10. In Robust2004$_{149}$, our approach performs better than the random method in all cases. In Robust2003, our approach outperforms the random method in the selection of the first 50 topics, while both perform similarly when we select more than 50 topics. Regarding WaS vs. NaD judging debate, when our budget is 20 hours, we are able to achieve higher τ scores by reducing the number of topics in both test collections. In Robust2004$_{149}$, when our budget is 30 hours, using 90–130 topics leads to higher τ scores than using all topics. When our budget is 40 hours, we are able to achieve similar τ scores by reducing the number of topics to 100. However, τ scores achieved by the random method monotonically increase as the number of topics increases (except 20-hours budget scenario with Robust2004$_{149}$ in which using 100–120 topics achieves very slightly higher τ scores than using all topics). That is to say that, *WaS judging leads to a better ranking of systems if we select the topics randomly, as reported by other studies (Bodoff and Li, 2007; Sanderson and Zobel, 2005; Webber et al., 2008). However, if we select the topics intelligently, we can achieve a better ranking by using fewer number of topics for a given budget.*

Re-usability of Test Collections: In this experiment, we compare our approach with the random topic selection method in terms of re-usability of the test collections with the selected topics. We again set topic development cost to 76 seconds and assume non-constant judging speed. We vary the total budget from 20 hours to 40 hours, as in the previous experiment. In order to measure the re-usability of the test collections, we adopt the following process. For each topic selection method, we first select the topics for the given topic subset size. Using only the selected topics, we then apply a leave-one-group-out method (Voorhees, 2001): for each group, we ignore the documents which only that group contributes to the pool and sample documents based on remaining documents. Then, the statAP score is calculated for the runs of the corresponding group. After applying this for all groups, we rank the systems based on their statAP scores and calculate Kendall's τ score compared to the ground-truth ranking of the retrieval systems. We repeat this process 20 times for our method and 5000 times for random method by re-selecting the topics.

The results are shown in Fig. 11. The vertical bars represent the standard deviation and the dashed horizontal line represents the performance when we employ all topics. There are several observations we can make from the results. First, our proposed method yields more *re-usable* test collections than random method in almost all cases. As the budget decreases, our approach becomes more effective in order to construct reusable test collections. Second, in all budget cases for both test collections, we can reach same/similar re-usability scores with fewer topics. Lastly, the τ scores achieved by the random topic selection method again monotonically increases as the number of topics increases in almost all cases. However, by intelligently reducing the number of topics, we can increase the reusability of the test collections in all budget scenarios for Robust2004$_{149}$ and 20-hours-budget scenario for Robust2003. Therefore, *the results suggest that NaD judging can yield more reusable test collections than WaS judging, when topics are selected intelligently*.

6. Conclusion

While the Cranfield paradigm (Cleverdon, 1959) for systems-based IR evaluations has demonstrated remarkably longevity, it has become increasingly infeasible to rely on TREC-style pooling to construct test collections at the scale of today's massive document collections. In this work, we proposed a new *intelligent topic selection* method which reduces the number of search topics (and thereby costly human relevance judgments) needed for reliable IR evaluation. To rigorously assess our method, we integrated previously disparate lines of research on intelligent topic selection and NaD vs. WaS judging. While prior work on intelligent topic selection has never been evaluated against shallow judging baselines, prior work on deep vs. shallow judging has largely argued for shallow judging, but assuming random topic selection. Arguing that ultimately one must ask whether it is actually useful to select topics, or should one simply perform shallow judging over many topics, we presented a comprehensive investigation over a set of relevant factors never previously studied together: 1) method of topic selection; 2) the effect of topic familiarity on human judging speed; and



(a) Robust 2003 - 20 Hours

(b) Robust 2004_{149} - 20 Hours

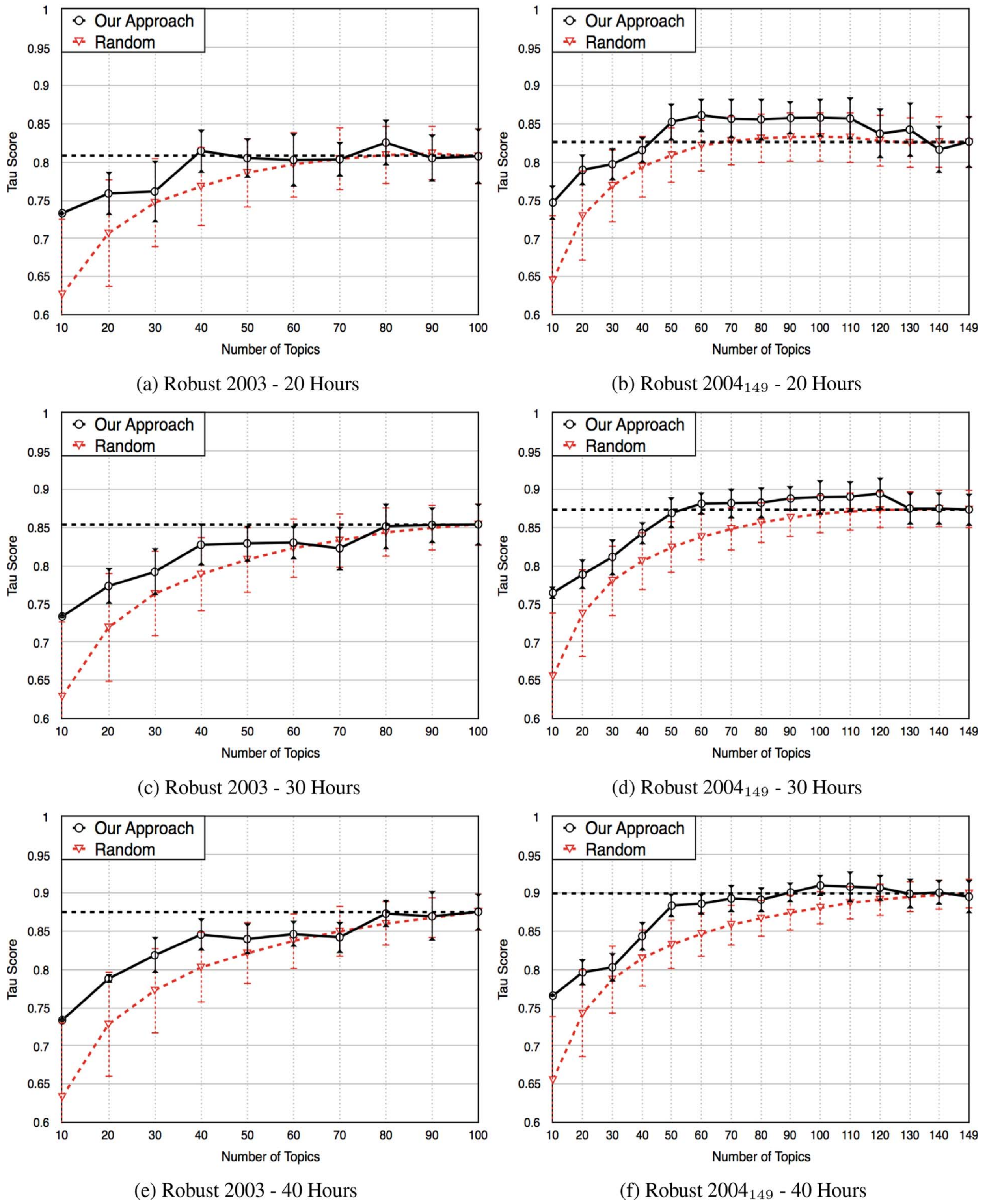
(c) Robust 2003 - 30 Hours

(d) Robust 2004_{149} - 30 Hours

(e) Robust 2003 - 40 Hours

(f) Robust 2004_{149} - 40 Hours

Fig. 10. Topic selection with different test collections when evaluation metric is statAP and we have a fixed budget. For random method, the number of query selection trials is 1000. For both methods, we run statAP 20 times and took the average of all trials. We assume that each judgment requires 15 seconds. The dashed horizontal line represents the performance when we employed all topics.



(a) Robust 2003 - 20 Hours

(b) Robust 2004_{149} - 20 Hours

(c) Robust 2003 - 30 Hours

(d) Robust 2004_{149} - 30 Hours

(e) Robust 2003 - 40 Hours

(f) Robust 2004_{149} - 40 Hours

Fig. 11. Re-usability Performance. For our method, we run statAP 20 times and take the average of tau scores achieved. For random approach, we select queries 5000 times and calculate statAP once for each query set. Topic generation cost is set to 76 seconds and we assume that the judging speed increases as they judge more documents. The vertical bars represents the standard deviation. The dashed horizontal line represents the performance when we employed all topics.

3) how different topic generation processes (requiring varying human effort) impact (i) budget utilization and (ii) the resultant quality of judgments.

Experiments on NIST TREC Robust 2003 and Robust 2004 test collections show that not only can we reliably evaluate IR systems with fewer topics, but also that: 1) when topics are intelligently selected, deep judging is often more cost-effective than shallow judging in evaluation reliability; and 2) topic familiarity and topic generation costs greatly impact the evaluation cost vs. reliability trade-off. Our findings challenge conventional wisdom in showing that deep judging is often preferable to shallow judging when topics are selected intelligently.

More specifically, the main findings from our study are as follows. First, in almost all cases, our proposed approach selects better topics yielding more reliable evaluation than the baselines. Second, shallow judging is preferable than deep judging if topics are selected randomly, confirming findings of prior work. However, when topics are selected intelligently, deep judging often achieves greater evaluation reliability for the same evaluation budget than shallow judging. Third, assuming that judging speed increases as more documents for the same topic are judged, increased judging speed has significant effect on evaluation reliability, suggesting that it should be another parameter to be considered in deep vs. shallow judging trade-off. Fourth, as topic generation cost increases, deep judging becomes preferable to shallow judging. Finally, assuming that short topic generation times reduce the quality of topics, and thereby consistency of relevance judgments, it is better to increase quality of topics and collect fewer judgments instead of collecting many judgments with low-quality topics. This also makes deep judging preferable than shallow judging in many cases, due to increased topic generation cost.

As future work, we plan to investigate the effectiveness of our topic selection method using other evaluation metrics, and conduct qualitative analysis to identify underlying factors which could explain *why* some topics seem to be better than others in terms of predicting the relative average performance of IR systems. We are inspired here by prior qualitative analysis seeking to understand what makes some topics harder than others (Harman & Buckley, 2009). Such deeper understanding could provide an invaluable underpinning to guide future design of topic sets and foster transformative insights on how we might achieve even more cost-effective yet reliable IR evaluation.

Acknowledgments

This work was made possible by NPRP grant# NPRP 7-1313-1-245 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors. We thank the Texas Advanced Computing Center (TACC) at the University of Texas at Austin for computing resources enabling this research.

References

Allan, J., Aslam, J. A., Carterette, B., Pavlu, V., & Kanoulas, E. (2008). *Million query track 2008 overview. Proceedings of the seventeenth text retrieval conference*.

Allan, J., Carterette, B., Aslam, J. A., Pavlu, V., Dachev, B., & Kanoulas, E. (2007). *Million query track 2007 overviewTechnical Report.* DTIC Document.

Alonso, O., & Mizzaro, S. (2009). *Can we get rid of trec assessors? using mechanical turk for relevance assessment. Proceedings of the SIGIR 2009 workshop on the future of ir evaluation15. Proceedings of the SIGIR 2009 workshop on the future of ir evaluation* 16.

Aslam, J. A., Pavlu, V., & Yilmaz, E. (2006). *A statistical method for system evaluation using incomplete judgments. Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval.* ACM541–548.

Aslam, J. A., & Yilmaz, E. (2007). *Inferring document relevance from incomplete information. Proceedings of the sixteenth ACM conference on conference on information and knowledge management.* ACM633–642.

Banks, D., Over, P., & Zhang, N.-F. (1999). Blind men and elephants: Six approaches to trec data. *Information Retrieval, 1*(1–2), 7–34.

Berto, A., Mizzaro, S., & Robertson, S. (2013). *On using fewer topics in information retrieval evaluations. Proceedings of the 2013 conference on the theory of information retrieval.* ACM9.

Bodoff, D., & Li, P. (2007). *Test theory for assessing ir test collections. Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval.* ACM367–374.

Buckley, C., Dimmick, D., Soboroff, I., & Voorhees, E. (2006). *Bias and the limits of pooling. Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval.* ACM619–620.

Buckley, C., & Voorhees, E. M. (2000). *Evaluating evaluation measure stability. Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval.* ACM33–40.

Buckley, C., & Voorhees, E. M. (2004). *Retrieval evaluation with incomplete information. Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval.* ACM25–32.

Carterette, B., Allan, J., & Sitaraman, R. (2006). *Minimal test collections for retrieval evaluation. Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval.* ACM268–275.

Carterette, B., Pavlu, V., Kanoulas, E., Aslam, J. A., & Allan, J. (2008). *Evaluation over thousands of queries. Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval.* ACM651–658.

Carterette, B., Pavlu, V., Kanoulas, E., Aslam, J. A., & Allan, J. (2009). *If i had a million queries. European conference on information retrieval.* Springer288–300.

Carterette, B., & Smucker, M. D. (2007). *Hypothesis testing with incomplete relevance judgments. Proceedings of the sixteenth ACM conference on conference on information and knowledge management.* ACM643–652.

Cleverdon, C. W. (1959). *The evaluation of systems used in information retrieval. Proceedings of the international conference on scientific information1. Proceedings of the international conference on scientific information* National Academy of Sciences Washington, DC,687–698.

Cormack, G. V., Palmer, C. R., & Clarke, C. L. (1998). *Efficient construction of large test collections. Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval.* ACM282–289.

Culpepper, J. S., Mizzaro, S., Sanderson, M., & Scholer, F. (2014). *Trec: topic engineering exercise. Proceedings of the 37th international ACM sigir conference on research & development in information retrieval.* ACM1147–1150.

Cummins, R., Jose, J., & O'Riordan, C. (2011). *Improved query performance prediction using standard deviation. Proceedings of the 34th international ACM sigir conference on research and development in information retrieval.* ACM1089–1090.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of statistics,* 1189–1232.

Grady, C., & Lease, M. (2010). *Crowdsourcing document relevance assessment with mechanical turk. Proceedings of the NAACL HLT workshop on creating speech and language data with Amazon's mechanical turk.* Association for Computational Linguistics172–179.

Guiver, J., Mizzaro, S., & Robertson, S. (2009). A few good topics: Experiments in topic set reduction for retrieval evaluation. *ACM Transactions on Information Systems (TOIS), 27*(4), 21.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: An update. *ACM SIGKDD explorations newsletter, 11*(1), 10–18.

Harman, D., & Buckley, C. (2009). Overview of the reliable information access workshop. *Information Retrieval, 12*(6), 615.

Hauff, C., Hiemstra, D., Azzopardi, L., & De Jong, F. (2010). *A case for automatic system evaluation. European conference on information retrieval*. Springer153–165.

Hauff, C., Hiemstra, D., De Jong, F., & Azzopardi, L. (2009). *Relying on topic subsets for system ranking estimation. Proceedings of the 18th ACM conference on information and knowledge management*. ACM1859–1862.

Hawking, D. (2000). *Overview of the trec-9 web track. Trec*.

Hawking, D., & Craswell, N. (2002). Overview of the trec-2001 web track. *NIST special publication,* 61–67.

Hosseini, M., Cox, I. J., Milic-Frayling, N., Shokouhi, M., & Yilmaz, E. (2012). *An uncertainty-aware query selection model for evaluation of ir systems. Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval*. ACM901–910.

Hosseini, M., Cox, I. J., Milic-Frayling, N., Vinay, V., & Sweeting, T. (2011). *Selecting a subset of queries for acquisition of further relevance judgements. Conference on the theory of information retrieval*. Springer113–124.

Jones, K. S., & van Rijsbergen, C. J. (1975). *Report on the need for and provision of an "ideal" information retrieval test collection (british library research and development report no. 5266)*. 43.

Kazai, G., & Sung, H. (2014). *Dissimilarity based query selection for efficient preference based ir evaluation. European conference on information retrieval*. Springer172–183.

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika, 30*(1/2), 81–93.

Lewandowski, D. (2015). Evaluating the retrieval effectiveness of web search engines using a representative query sample. *Journal of the Association for Information Science and Technology, 66*(9), 1763–1775.

Lin, T.-C., & Cheng, P.-J. (2011). *Query sampling for learning data fusion. Proceedings of the 20th ACM international conference on information and knowledge management*. ACM141–146.

McDonnell, T., Lease, M., Kutlu, M., & Elsayed, T. (2016). *Why is that relevant? collecting annotator rationales for relevance judgments. Proceedings of the 4th AAAI conference on human computation and crowdsourcing (HCOMP)*. AAAI139–148.

Mehrotra, R., & Yilmaz, E. (2015). *Representative & informative query selection for learning to rank using submodular functions. Proceedings of the 38th international ACM sigir conference on research and development in information retrieval*. ACM545–554.

Mizzaro, S., & Robertson, S. (2007). *Hits hits trec: exploring ir evaluation results with network analysis. Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*. ACM479–486.

Moffat, A., Webber, W., & Zobel, J. (2007). *Strategic system comparisons via targeted relevance judgments. Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*. ACM375–382.

Moghadasi, S. I., Ravana, S. D., & Raman, S. N. (2013). Low-cost evaluation techniques for information retrieval systems: A review. *Journal of Informetrics, 7*(2), 301–312.

Nuray, R., & Can, F. (2006). Automatic ranking of information retrieval systems using data fusion. *Information Processing & Management, 42*(3), 595–614.

Pavlu, V., & Aslam, J. (2007). *A practical sampling strategy for efficient retrieval evaluationTechnical Report*. College of Computer and Information Science, Northeastern University.

Robertson, S. (2011). *On the contributions of topics to system evaluation. European conference on information retrieval*. Springer129–140.

Sakai, T. (2007). *Alternatives to bpref. Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*. ACM71–78.

Sakai, T. (2014a). *Designing test collections for comparing many systems. Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. ACM61–70.

Sakai, T. (2014b). *Designing test collections that provide tight confidence intervals. Forum on information technology 2014 rd3*.

Sakai, T. (2014c). *Topic set size design with variance estimates from two-way anova. EVIA@ NTCIR*. Citeseer.

Sakai, T. (2016a). *The effect of score standardisation on topic set size design. Information retrieval technology*. Springer16–28.

Sakai, T. (2016b). *Simple and effective approach to score standardisation. Proceedings of the 2016 ACM on international conference on the theory of information retrieval*. ACM95–104.

Sakai, T. (2016c). Topic set size design. *Information Retrieval Journal, 19*(3), 256–283.

Sakai, T., & Shang, L. (2016). *On estimating variances for topic set size design. Proceedings of the seventh international workshop on evaluating information access (EVIA 2016)*.

Sakai, T., Shang, L., Lu, Z., & Li, H. (2015). *Topic set size design with the evaluation measures for short text conversation. Asia information retrieval symposium*. Springer319–331.

Sanderson, M. (2010). *Test collection based evaluation of information retrieval systems*. Now Publishers Inc.

Sanderson, M., Scholer, F., & Turpin, A. (2010). *Relatively relevant: Assessor shift in document judgements. Proceedings of the Australasian document computing symposium*60–67.

Sanderson, M., & Zobel, J. (2005). *Information retrieval system evaluation: effort, sensitivity, and reliability. Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval*. ACM162–169.

Scholer, F., Kelly, D., Wu, W.-C., Lee, H. S., & Webber, W. (2013). *The effect of threshold priming and need for cognition on relevance calibration and assessment. Proceedings of the 36th international ACM sigir conference on research and development in information retrieval*. ACM623–632.

Soboroff, I., Nicholas, C., & Cahan, P. (2001). *Ranking retrieval systems without relevance judgments. Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*. ACM66–73.

Urbano, J. (2016). Test collection reliability: A study of bias and robustness to statistical assumptions via stochastic simulation. *Information Retrieval Journal, 19*(3), 313–350.

Urbano, J., Marrero, M., & Martín, D. (2013). *On the measurement of test collection reliability. Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval*. ACM393–402.

Voorhees, E. M. (2000). Variations in relevance judgments and the measurement of retrieval effectiveness. *Information processing & management, 36*(5), 697–716.

Voorhees, E. M. (2001). *The philosophy of information retrieval evaluation. Workshop of the cross-language evaluation forum for european languages*. Springer355–370.

Voorhees, E. M. (2003). *Overview of trec 2003. Trec*1–13.

Voorhees, E. M. (2004). *Overview of the trec 2004 robust track. Trec4*.

Voorhees, E. M. (2009). *Topic set size redux. Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval*. ACM806–807.

Voorhees, E. M. (2016). Personal communication

Webber, W., Moffat, A., & Zobel, J. (2008). *Statistical power in retrieval experimentation. Proceedings of the 17th ACM conference on information and knowledge management*. ACM571–580.

Yilmaz, E., & Aslam, J. A. (2006). *Estimating average precision with incomplete and imperfect judgments. Proceedings of the 15th ACM international conference on information and knowledge management*. ACM102–111.

Yilmaz, E., & Aslam, J. A. (2008). Estimating average precision when judgments are incomplete. *Knowledge and Information Systems, 16*(2), 173–211.

Zobel, J. (1998). *How reliable are the results of large-scale information retrieval experiments? Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval*. ACM307–314.