Understanding and Predicting the Characteristics of Test Collections

Md Mustafizur Rahman[1], Mucahid Kutlu[2], and Matthew Lease[1]

[1] School of Information, University of Texas at Austin, TX, USA
{nahid,ml}@utexas.edu
[2] Dept. of Computer Eng., TOBB Economy and Tech. University, Ankara, Turkey
m.kutlu@etu.edu.tr

Abstract. Shared-task campaigns such as NIST TREC select documents to judge by pooling rankings from many participant systems. Therefore, the quality of the test collection greatly depends on the number of participants and the quality of submitted runs. In this work, we investigate i) how the number of participants, coupled with other factors, affects the quality of a test collection; and ii) whether the quality of a test collection can be inferred prior to collecting relevance judgments. Experiments on six TREC collections demonstrate that the required number of participants to construct a high-quality test collection varies significantly across different test collections due to a variety of factors. Furthermore, results suggest that the quality of test collections can be predicted.

Keywords: Evaluation; Test Collections; Pooling; Reusability

1 Introduction

The system-based IR evaluation relies on having a *test collection*, which consists of a document collection, a set of search topics, and relevance judgments for selected document-topic pairs. Typically a test collection is constructed by organizing a *shared-task* where the organizers (e.g., NIST TREC) usually provide a collection of documents and a set of topics developed by experts, and ask the participating groups to submit their ranked list of documents (i.e., *runs*) for each topic. Subsequently, the documents to be judged are selected from the ranked lists of participants using various methods such as *pooling* (Sparck Jones and Van Rijsbergen, 1975) and bandit techniques (Voorhees, 2018).

While shared-tasks are extremely useful to build test collections, the cost of test collections is still very high due to the huge amount of required human judgments. Besides, organizing the shared task itself is also difficult, slow, and expensive. Furthermore, documents selected for human annotation completely depend on the participating groups and their submitted runs. Therefore, if very few groups participate in a shared-task, the resultant test collection might not be of high quality as desired. Then, what should task organizers do when there are few participants? They might a) cancel the shared-task, or b) accept the risk of building a low-quality test collection and proceed as it is with all the costs of organizing the shared-task and collecting human annotation, or c) increase the publicity of the shared-task to attract more participants, or d) ask the existing participants to submit manual runs to increase the diversity of the results.

In this work, we conduct experiments to shed light on the impact of the number of participants on the quality of pooling-based test collections, particularly the *reusability* of a test collection. We investigate the following two key research questions. **RQ-1)** i) how does the number of participants coupled with other factors affect the quality of a test collection? **RQ-2)** Can we predict the quality of a test collection prior to collecting relevance judgments?

For RQ-1, our experiments vary the number of shared-task participants by down-sampling participants from past TREC tasks in order to construct simulated test collections. Then we analyze the impact of the interaction between the number of participants and other factors (e.g., the number of topics, collection size, etc.) on resultant test collection quality. For RQ-2, we develop a model to predict test collection quality using only the number of participants, the number of topics, collection size, and pool depth. We analyze the generalization performance of the prediction model using our designed "Leave-one-test-collection-out" setup[3].

Our analysis leads to the following recommendations for shared-tasks:

- With few participants, increasing the pool depth increases test collection quality more than increasing the number of topics.
- With few participants, organizers should maximally encourage manual runs.
- With larger document collections, it is particularly important that organizers attract strong participation in order to achieve test collection quality.
- To help guide track planning, organizers should consider using a predictive model such as ours to infer the expected quality of the test collection based on the planned pool depth and expected number of participants, so that they can adjust course as needed if the predicted quality is low.

2 Factors Impacting the Qualities of Test Collections

The quality of a test collection depends on a variety of factors, such as: i) the number of topics; ii) pool depth (assuming pool-based judging); iii) the number of participants; iv) the collection size; and v) the types of runs (e.g., manual runs and automatic runs) and their quality. One might also consider vi) the target evaluation metric (e.g., MAP@1000, NDCG@10) in assessing how well a test collection supports reliable evaluation for a given retrieval task, as measured by a particular metric. Since constructing a test collection is expensive, *rusability* is desirable. Reusability is often measured by how a given run contributing to the pool would have been assessed if excluded from the pool. In this study, we focus on reusability as a key measure of the test collection quality.

[3] In order to ensure the reproducibility of our findings, we will share our source code in the final version of the manuscript.

While considerable work has investigated how the quality of a test collection is impacted by the above-mentioned factors, prior studies have not explored how the number of participants interacts with other factors.

Number of topics. Sparck Jones and Van Rijsbergen (1976) suggest that 250 topics can be acceptable, but 1000 topics are needed for reliable evaluation. Voorhees (2000) performs an empirical analysis on the TREC6 test collection and shows that system rankings computed based on 5 or 10 topics are relatively unstable, whereas a set of 25 or more topics produces a stable ranking of IR systems. Buckley and Voorhees (2000) calculate the error rate of various evaluation measures and find that for reliable evaluation the required number of topics should be at least 25. Webber et al. (2008) recommend that a set of 150 topics is required to statistically distinguish the performance of one IR system from other IR systems. Zobel (1998) finds that a set of 25 topics can reasonably predict the performance of IR systems on a separate set of 25 topics.

Pool depth. Prior work has also studied the trade-off between collecting fewer judgments with more topics (i.e., Wide and Shallow (WaS) judging) vs. more judgments with fewer topics (i.e., Narrow and Deep (NaD) judging). Carterette et al. (2008) report on TREC Million Query track and conclude that WaS judging produces a more reliable evaluation of IR systems than NaD judging. Kutlu et al. (2018) find that NaD judging is preferred to WaS judging if we consider intelligent topic selection or other hidden costs of shallow judging, such as topic creation time and noisier judgments. Voorhees (2018) investigates the impact of varying pool depth on the reusability of a test collection.

Run types. To see how different types of runs (e.g., manual and automatic) impact collection quality, Büttcher et al. (2007) adapt the "leave-one-group-out"(Zobel, 1998) experiment by removing all unique documents contributed by the manual runs from the pool. The authors find that their setup ranks the runs differently than found in the original TREC 2006 Terabyte task.

Collection size. Hawking and Robertson (2003) study the effect of the document collection size on Very Large Collection track (Hawking et al., 1998) and observe a higher value of P@20 for runs in the larger collection. Interestingly, they find no change in value between the large and small document collection when the runs are ranked based on MAP.

3 The Impact of Varying the Number of Participants

Our experimental design for analyzing the impact of the number of groups is shown in **Algorithm 1**. First, we construct the original qrels (Q_o) using all participants from set G [**Line 2**]. Then we evaluate all runs using this original qrels (Q_o) in terms of a ranking metric (e.g., MAP@1000, NDCG@10, etc.) and store the ranking of runs in E [**Line 3**]. Next, we change the group number, g, from 1 to $|G|$ with an increment of 1 at each iteration to create test collection with varying number of participants [**Line 5**]. At each iteration, we randomly sample g number of groups ($\hat{G}_i$) from the set of groups G [**Line 7**] and construct the simulated test collection (i.e., qrels) Q_g using only the participants in $\hat{G}_i$ [**Line**

8]. Then, we evaluate all participating runs in set G by using simulated test collection Q_g in terms of a ranking metric (e.g., MAP@1000, NDCG@10, etc.) and store these new ranking of runs in E_g **[Line 9]**. We calculate the performance difference in terms of τ_{ap} (Yilmaz et al., 2008) and Max Drop (Voorhees, 2018) (i.e., the maximum drop in a run's rank, between the original ranking of runs E and the ranking of runs obtained via the respective simulated test collection E_g) **[Line 11]**. Note that we calculate average scores across different group samples for a particular parameter setup. In addition, we can also utilize Algorithm 1 to experiments with a varying number of topics and varying pool depth because it takes the set of topics T and pool depth P as inputs.

Algorithm 1: Experimental Design

Input : Set of groups G • Number of samples for groups N • Set of topics T • Pool depth P

Output: E, A set of performance score indexed by group number

1 $R \leftarrow$ Total number of runs from all groups in G
2 $Q_o \leftarrow Construct\_Qrels(G, T, P)$ ▷ official qrels
3 $E \leftarrow Evaluate\_runs(R, Q_o)$ ▷ Evaluate all runs with Q_o
4 $\hat{E} \leftarrow \emptyset$ ▷ keeps scores of systems with reduced qrels
5 **for** *group no* $g \leftarrow 1$ **to** $|G|$ **do**
6   **for** *sample number* $i \leftarrow 1$ **to** N **do**
7         $\hat{G}_i \leftarrow$ randomly sample g groups from G
8         $Q_g \leftarrow$ Construct_Qrels($\hat{G}_i, T, P$)
9         $E_g \leftarrow Evaluate\_runs(R, Q_g)$ ▷ Evaluate all runs using qrels Q_g
10         $\hat{E} \leftarrow \hat{E} \cup E_g$
11 **return** *Evaluate_Performance_Difference(E,$\hat{E}$)*

3.1 Datasets

We conduct our experiments on six TREC tracks and datasets: the 2013-2014 Web Tracks on ClueWeb12[4], the 2006 Terabyte track on Gov2[5], and the 2004 Robust Retrieval Task (Robust'04), the 1999 TREC-8 *ad hoc* track, the 1998 TREC-7 *ad hoc* track on TIPSTER disks 4-5[6] (excluding the *congressional record*). **Table 1** provides statistics about test collections we use. Later tracks have fewer participants than earlier tracks both in terms of the number of groups and the submitted runs. Later tracks also tend to use larger document collections, without commensurate increase in pool depth, leading to an increasing prevalence of relevant documents in judged pools, from ∼5% to ∼40%.

[4] `lemurproject.org/clueweb12` [5] `ir.dcs.gla.ac.uk/test_collections/gov2-summary.htm`
[6] `trec.nist.gov/data/docs_eng.html`

Table 1. Statistics about the test collections used in this study.

Track	#Groups	#Manual Runs	#Auto Runs	Pool Depth	Collection	#Topics	#Docs	#Judged	%Rel
WT'14	9	4	26	25	ClueWeb12	50	52,343,021	14,432	39.2%
WT'13	13	3	31	10 and 20	ClueWeb12	50	52,343,021	14,474	28.7%
TB'06	20	19	61	50	Gov2	50	25,205,179	31,984	18.4%
Adhoc'99	40	9	62	100	Disks45-CR	50	528,155	86,830	5.4%
Adhoc'98	41	16	68	100	Disks45-CR	50	528,155	80,345	5.8%
Robust'04	14	0	110	100	Disks45-CR	249	528,155	311,410	5.6%

3.2 Results and Discussion

Impact of Number of Topics. We first consider how the number of topics interacts with the number of groups in relation to test collection quality. To explore this, we study Robust'04 using various subsets of its 249 topics, randomly sampling m topics, with $m \in \{50, 100, 150, 200, 249\}$. Robust'04 topics can be categorized into 5 sets: 301-350 (Adhoc'97), 350-400 (Adhoc'98), 401-450 (Adhoc'99), 601-650 (Robust'03-Hard), and 651-700 (Robust'04). To ensure coverage over the different sets in our sampling, we apply stratified sampling. **Algorithm 1** implements our experimental setup for a given topic subset (we vary the topic subset outside of the algorithm). The pool depth is set to 100, and we evaluate MAP@1000 and NDCG@10.

Figure 1 shows τ_{ap} (larger is better) and Max Drop (smaller is better) on the Robust'04 test collection using all runs. Each line shows an average value of the computed performance metric (e.g., τ_{ap}) across 4 different random samples (i.e., N is set to 4 in Algorithm 1) of groups when m topics are randomly sampled from 249 topics. The first row presents τ_{ap} and Max Drop when the runs are ranked using MAP (i.e., MAP@1000) while the second row reports the same metric when runs are ranked using NDCG@10. We report the Area under the Curve (AUC) for each of the line plots and the Pearson correlation (ρ) between the number of groups and the corresponding performance metrics.

Let us first consider when runs are ranked by MAP (Figure 1, first row). For τ_{ap} and any number of groups g, we do not see any significant difference in AUC when we down-sample to 50, 100, 150, or 200 topics vs. all 249 topics. Furthermore, we achieve a τ_{ap} correlation of 0.9 using only 3 participating groups, irrespective of the number of topics. We observe the same outcome when using NDCG@10 (Figure 1, bottom row): there is no significant difference in AUC, and we only need 2 participating groups to achieve a τ_{ap} correlation of 0.9 or above. Since NDCG@10 is far shallower than MAP@1000, it is reasonable to observe that we might need fewer groups to achieve $\tau_{ap} >= 0.9$ using NDCG@10. On the other hand, Max Drop for MAP and NDCG shows a significant difference in AUC, when we down-sample topics. We do not see any decreasing pattern in AUC for MAP or NDCG as we increase the number of topics.

In these experiments, we see that for any given number of participating groups, increasing the number of topics does not improve test collection reusability. For both τ_{ap} and Max Drop metrics, this observation holds. However, we

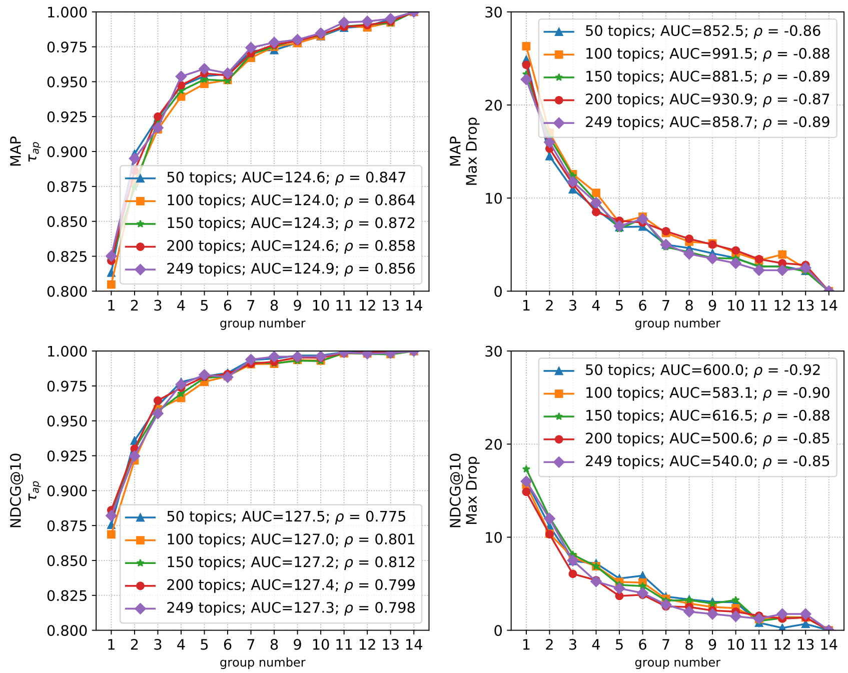

Fig. 1. τ_{ap} (first column), and Max Drop (second column) obtained by simulated test collections with a varying number of topics on the Robust'04 dataset. The x-axis represents the number of groups and the y-axis shows results when runs are ranked using MAP (top row) and NDCG@10 (bottom row).

must note the extent of our experiments. First, since we experiment using only Robust'04 (given its 249 topics), further analysis on other test collections is needed. Second, our experiments only vary the number of topics above 50, so results with more spartan topic sizes might vary. Third, we assume a fixed pool depth of 100. Fourth, Robust'04 contains only automatic runs; manual runs often find other unique relevant documents. Fifth, Robust'04 dataset is relatively small compared to a modern collection such as ClueWeb'12 (Table 1). Since larger collections tend to contain more relevant documents, increasing the number of topics for these larger document collections may be more valuable.

Impact of Pool Depth. In the previous experiment, we observe how varying the number of topics interacts with the varying number of participating groups to build a reusable test collection while keeping the pool depth fixed. In this experiment, we also change the pool depth along with the number of topics and the number of groups. The experimental setup for this experiment is the same as discussed in the previous experiment except we vary the pool depth p where p takes values from the set $\{20, 40, 60, 80, 100\}$.

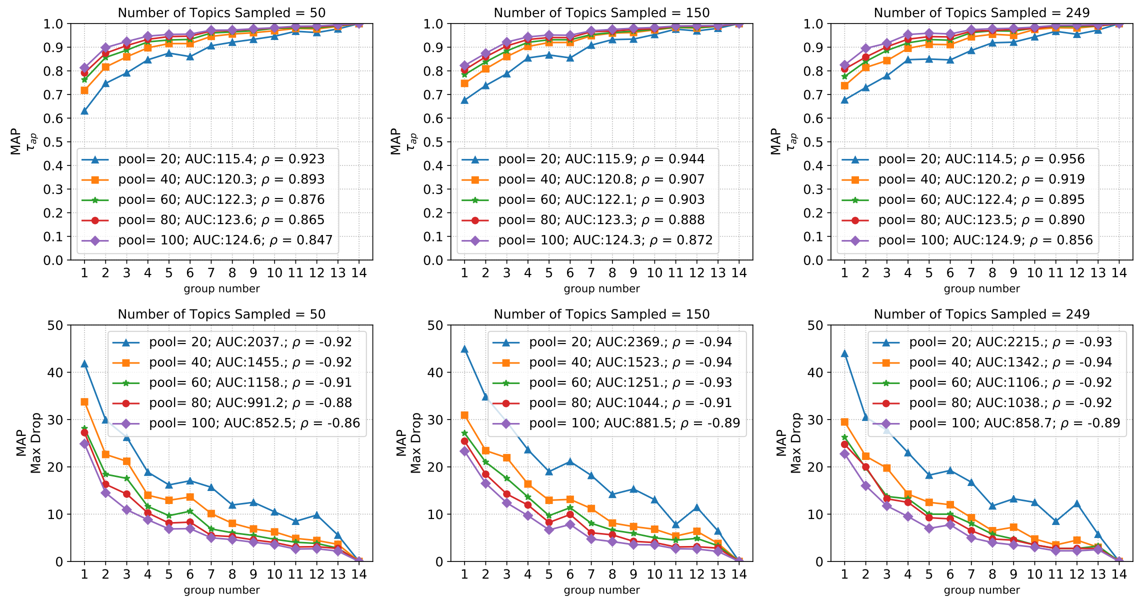

Fig. 2. τ_{ap} (first row), and Max Drop (second row) obtained by simulated test collections with a varying number of topics along with a varying pool depth on the Robust04 dataset. The x-axis represents the number of groups and the y-axis shows results when runs are ranked using MAP.

The results for this experiment are reported in **Figure 2**. The top row represents results for τ_{ap} whereas the bottom rows present results for Max Drop. The columns of Figure 2 indicate the number of topics sampled for each of those pool depth variations. Due to space constraints, we report the results where the runs are ranked using MAP. Please note that from the previous experiments, considering it's limitation discussed above, we already know that increasing the number of topics does not improve the reusability of a test collection when the pool depth is 100. The conclusion still holds at each varying pool depth reported in Figure 2.

By observing Figure 2, we find that for a fixed number of topics, increasing the pool depth improves the AUC in terms of τ_{ap} and lowers the value of AUC, which indicates a better quality test collection. The greatest improvement in AUC happens when we increase the pool depth from 20 to 40 in all topic sets we investigate, suggesting that it has a high return on investment. Based on our results, if we have enough evaluation budget, using a pool depth of at least 40 seems a reasonable choice.

Another interesting observation from Figure 2 is that how the number of groups interacts with the pool depth. For example, in the plot for 50 topics, when we have a pool depth of 20 only, we need at least 7 groups (half of the total number of groups in Robust'04 test collection), to achieve a τ_{ap} correlation of 0.9 or above. However, if the number of participants goes down to 3 groups (one-fifth of the total number of groups in Robust'04 test collection), we need a pool depth of 80 to achieve the same τ_{ap} correlation. The observation holds for all other varying numbers of topics sampled in Figure 2.

Based on the above discussion, we can conclude that the number of participants is the most important factor for the quality of test collections. Therefore, shared-task organizers should pay attention to the publicity of shared tasks. However, if we have a few participating groups, for a given judging budget, rather than increasing the number of topics, we should increase pool depth in order to have a reusable test collection. It should be noted this conclusion is subjected to all the limitations discussed in the previous experiment except for the pool depth.

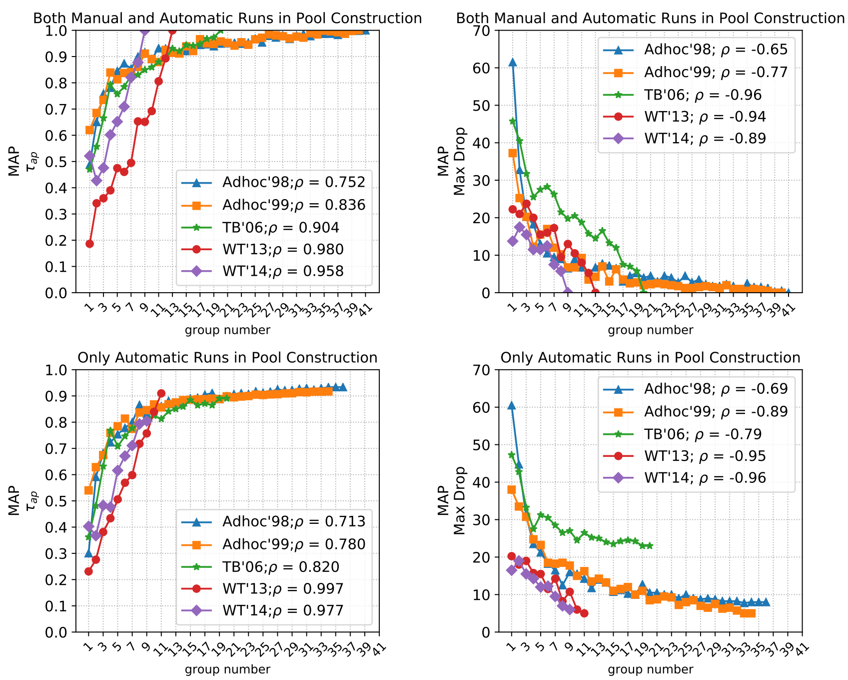

Fig. 3. τ_{ap}, and Max Drop obtained by simulated test collections with a varying number of groups with manual runs (top row) and without manual runs (bottom row) on the five TREC datasets. The x-axis represents the number of groups. Runs are ranked using MAP.

Impact of the Document Collection Size. We conduct experiments on five different test collections, namely Adhoc'99, Adhoc'98, TB'06, WT'13, and

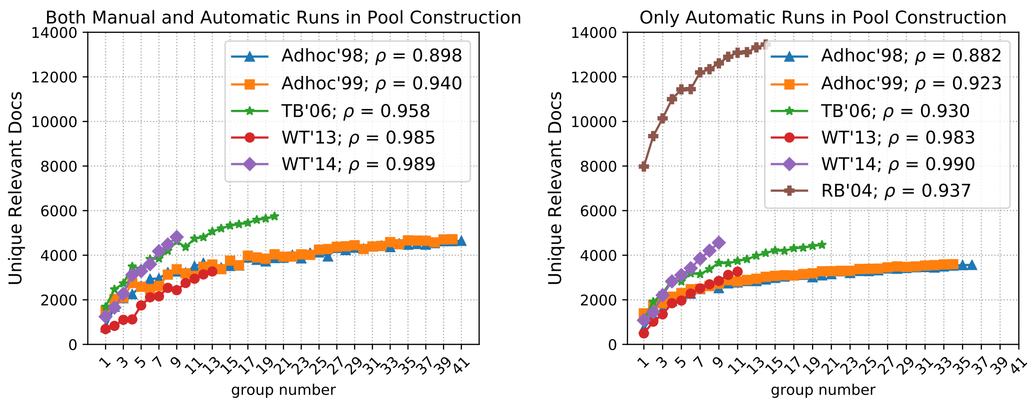

Fig. 4. The number of unique relevant documents obtained by simulated test collections on six different test collections. Left plot does not include Robust'04 (RB'04) dataset as it contains only automatic runs. In the right plot, Robust'04 test collection has a higher number of unique relevant documents because it has 249 topics whereas the other datasets have 50 topics only.

WT'14, which contain both types of runs (i.e., manual and automatic runs), as seen in Table 1. We vary the number of groups and report respective τ_{ap}, Max Drop scores (See **Figure 3**), and the number of relevant documents (See **Figure 4**), following Algorithm 1. We do not report AUC since the AUC is not comparable across different collections. The top row of Figure 3 considers both manual and automatic runs in the construction of a test collection and the bottom row only considers the automatic runs. We present results here using only MAP due to space constraints.

From the Table 1, we can see that the size of document collection for Adhoc'99 and Adhoc'98 test collection is very small ($\approx$ 0.5 Million) compared to the size of document collection ($\approx$ 52 Million) for WT'13 and WT'14 test collections. Although the number of unique relevant documents found depends on the number of topics, pool depth, and the number of participating groups, it is intuitive that for a fixed number of topics and pool depth, a larger document collection usually has a higher number of unique relevant documents. Therefore, the size of the document collection coupled with a varying number of participating groups will affect the reusability of a test collection. By observing Figure 3 (top row), we can see that for Adhoc'98 and Adhoc'99 collections with both manual and automatic runs, we can achieve a τ_{ap} of 0.9 or above when there are 8 and 9 participating groups, respectively (i.e., approximately 20% of the original number of participating groups for the respective test collections). However, for WT'13 and WT'14 datasets, we need 12 and 8 groups (Figure 3, top row), respectively, which is around on average 90% of the original number of participating groups for those respective test collections.

Although there is no acceptable range of values for Max Drop, if we consider that the 90% of the original number of groups participate for WT'13 and WT'14 test collections, we find the Max Drop is in between 6 and 10. Regarding Pearson correlation (ρ) scores computed between the number of groups and the corresponding τ_{ap} and Max Drop, we observe higher Pearson correlation values for WT'13 and WT'14 collections than other collections. This is because τ_{ap} keeps increasing and Max Drop keeps decreasing with an increasing number of groups in WT'13 and WT'14 test collections. On the other hand, both Adhoc'99 and Adhoc'98 collections have low Pearson correlation scores because τ_{ap} increases and Max Drop decreases when the number of groups is increased from 2 to 10. However, after having 10 groups, τ_{ap} and Max Drop scores become almost stable for these two collections.

In summary, we can conclude that we would need a higher number of participating groups to have a reusable collection if the underlying document collection is very large. This is probably due to the fact that runs in these larger document collections would return more unique-relevant documents than runs in other test collections (Figure 3). Therefore, a run might be highly affected if it does not contribute to the pool. As another hypothesis, the groups participating in the recent shared-tasks are able to develop a more diverse set of IR runs than the groups participated in earlier shared-tasks due to the progress in the field of IR.

However, it should be noted that the finding reported in this experiment has certain limitations. Firstly, all five test collections used in this analysis have only 50 topics. Secondly, for WT'13 and WT'14 collections, the employed pool depth is very shallow (Table 1). A deeper pool depth and a higher number of topics might provide us a different conclusion than the one stated here.

Impact of Manual Runs. In order to see the impact of manual runs on test collection reusability, we remove all manual runs from the simulated test collections and conduct the same experiments as described above. **Figure 3**, bottom row presents the results for this particular setup. Before going into further details, we should notice from **Figure 4** that the number of unique-relevant documents noticeably reduces when we do not use any manual run in AdHoc'98, AdHoc'99, and TB'06 test collections, confirming that manual runs usually provide more unique-relevant documents than automatic runs. We also observe that WT'13 and WT'14 test collections are less affected in terms of the number of unique relevant documents by not having as many manual runs as other collections (See Table 1).

Comparing results between the top and bottom rows of Figure 3, we find that not any using manual run increases the required number of participating groups to achieve a τ_{ap} correlation of 0.9 or above. For example, for Adhoc'98, and Adhoc'99 collections, we need at least 50% of the original number of participants (Table 3, bottom row) to achieve a τ_{ap} correlation of 0.9 or above which is 20% (Table 3, top row) when we include manual runs in the test collections. For TB'06 and WT'13 collections, we actually need 100% of the original number of participants. Our observations are also similar for Max Drop scores. This suggests that the unique-relevant documents detected by manual runs could not have been detected by any of the automatic runs, thereby affecting the ranking of manual runs during the evaluation.

The experimental evidence informs us that if we have a very few participating groups, it is always better to have as much as possible manual runs to develop a reusable test collection. Again, we should note that this finding is limited by the facts that we have only 50 topics and the pool depth is very shallow for the WT'13 and WT'14 test collections. Further experiments on other test collections might suggest differently.

4 Predicting the Qualities of Test Collections

We investigate whether it is possible to forecast the quality of a test collection even before gathering the ranked lists of participants. Our rationale is that the shared-task organizers can act accordingly based on the predicted quality of a test collection even before spending budget on collecting relevance judgments. In this study, we focus on predicting τ_{AP} as a measure of reusability.

To generate data for our model, we use the same simulated test collections constructed from Algorithm 1 and employ MAP to compute τ_{ap}. We utilize the following features for the prediction model: i) the number of participating groups (G), ii) the number of topics (T), iii) the pool depth (P) and iv) the size of the document collection (C). Then we fit a Multiple Linear Regression model: $\hat{y} = W_0 + W_1 * G + W_2 * T + W_3 * P + W_3 * C$ on the training data to predict τ_{ap}. Here W's are the learned weights for the features of the model. As a performance measure of our prediction model, we report Mean Squared Error (MSE) $= \sum_{i=1}^{n} \frac{(\hat{y_i}-y_i)*(\hat{y_i}-y_i)}{n}$, where y_i, and $\hat{y_i}$ are the predicted and true target value of τ_{ap} respectively and n is the total number of data points. A lower value of MSE indicates a better model.

To understand the generalization performance of the prediction model, we employ "*Leave-one-test-collection-out*" (LOTO) strategy. In this LOTO setup, in turn, we hold out one test collection and utilize the remaining test collections from the set of test collections to train the prediction model, and then we test the predictive performance of the trained model on the held-out test collection. For example, in the first row of **Table 2**, we utilize {Adhoc'99, TB'06, WT'13

Table 2. Performance of the multiple linear regression model using Leave-one-test-collection-out setup on five different test collections.

Training Set	Testing Set	Intersection of Document Collection between Train & Test Set	MSE τ_{ap}
{Adhoc'99 , TB'06 , WT'13 , WT'14}	Adhoc'98	Yes	0.005
{Adhoc'98 , TB'06 , WT'13 , WT'14}	Adhoc'99	Yes	0.003
{Adhoc'98 , Adhoc'99 , WT'13 , WT'14}	TB'06	No	0.171
{Adhoc'98 , Adhoc'99 , TB'06 , WT'14}	WT'13	Yes	0.060
{Adhoc'98 , Adhoc'99 , TB'06 , WT'13}	WT'14	Yes	0.055

, WT'14} test collections as training set and Adhoc'98 test collection as testing set for our model. Since Adhoc'98 in the test set shares the same document collection (Table 1) with Adhoc'99 from the train set, the 3rd column of 1st row indicates "Yes". In contrast, TB'06 in the testing set (3rd row) does not share the document collection with any of the test collections utilized in the training set and thus the 3rd column of the 3rd row indicates "No". MSE for predicting τ_{ap} is reported in 4th column of Table 2.

From Table 2, we observe that the model can predict τ_{ap} with a very high accuracy (MSE ≤ 0.06) for all five test collections except for TB'06. This can be because a) runs in Adhoc'98 and Adhoc'99 might be similar due to closeness of two shared-tasks in terms of time, and b) both have the same document collection (Column 3 of Table 2). We can apply the same reasoning for WT'13 and WT'14 test collections.

In summary, our prediction performance improves when we use test collections from the same document collection for both training and testing sets of the model. In practice, our prediction model can be especially useful after its first year as TREC usually continues a track for more than one year. Thus, the results in the first year can be used to forecast the quality of the test collections in the following years for the same track. Test collection construction parameters (i.e., topic set size and pool depth) can be set based on the predictions.

5 Conclusion and Future Work

In this work, we investigate how varying the number of participating groups coupled with other factors in a shared-task affects the reusability of a test collection. Our main findings based on our extensive experiments conducted on six TREC test collections are as follows. Firstly, when we have a very few participating groups in a shared-task, increasing the pool depth provides a more reusable test collection than increasing the number of topics. Secondly, the size of the document collection and the types of runs play an important role when the number of participants is very few. Thirdly, the reusability of a test collection can be predicted with high accuracy when the test collections used for training and testing the model have the same underlying document collection. We believe that our findings will be useful for shared-task organizers to take necessary actions while organizing the event.

There are many possible future directions for this work. For example, our experimental analysis is conducted on pooling-based test collections. However, we plan to extend our work on analyzing test collections constructed using other techniques. For example, the TREC 2017 Common Core track (Allan et al., 2017) test collection is constructed using a bandit-based technique (Losada et al., 2016). In addition, we also plan to address the mentioned limitations of our analysis by using more test collections, preferable recent ones in which deep learning based IR models are used. Future work might also explore the prediction of test collection quality when the ranked lists of documents are submitted but relevance judgments are not collected yet. This scenario will enable using different features such as the number of unique documents retrieved by groups, rank correlation of ranked lists, and others with more sophisticated machine learning models.

Bibliography

James Allan, Donna Harman, Evangelos Kanoulas, Dan Li, Christophe Van Gysel, and Ellen M Voorhees. 2017. TREC 2017 Common Core Track Overview.. In *TREC*.

Chris Buckley and Voorhees. 2000. Evaluating Evaluation Measure Stability. (2000), 33–40. https://doi.org/10.1145/345508.345543

Stefan Büttcher, Charles LA Clarke, Peter CK Yeung, and Ian Soboroff. 2007. Reliable information retrieval evaluation with incomplete and biased judgements. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 63–70.

Ben Carterette, Virgil Pavlu, Evangelos Kanoulas, Javed A. Aslam, and James Allan. 2008. Evaluation over thousands of queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '08*. ACM Press, Singapore, Singapore, 651. https://doi.org/10.1145/1390334.1390445

David Hawking, Nick Craswell, and Paul Thistlewaite. 1998. Overview of TREC-7 very large collection track. *NIST SPECIAL PUBLICATION SP* (1998), 93–106.

David Hawking and Stephen Robertson. 2003. On collection size and retrieval effectiveness. *Information retrieval* 6, 1 (2003), 99–105.

Mucahid Kutlu, Tamer Elsayed, and Matthew Lease. 2018. Intelligent topic selection for low-cost information retrieval evaluation: A New perspective on deep vs. shallow judging. *Information Processing & Management* 54, 1 (Jan. 2018), 37–59. https://doi.org/10.1016/j.ipm.2017.09.002

David E. Losada, Javier Parapar, and Álvaro Barreiro. 2016. Feeling lucky?: multi-armed bandits for ordering judgements in pooling-based evaluation. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing - SAC '16*. ACM Press, Pisa, Italy, 1027–1034. https://doi.org/10.1145/2851613.2851692

K Sparck Jones and C Van Rijsbergen. 1975. Report on the Need for and Provision of an" Ideal. *Information Retrieval Test Collection* (1975).

Karen Sparck Jones and Cornelis Joost Van Rijsbergen. 1976. Information retrieval test collections. *Journal of documentation* 32, 1 (1976), 59–75.

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

Voorhees. 2018. On Building Fair and Reusable Test Collections Using Bandit Techniques. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. ACM, New York, NY, USA, 407–416. https://doi.org/10.1145/3269206.3271766

William Webber, Alistair Moffat, and Justin Zobel. 2008. Statistical power in retrieval experimentation. In *Proceeding of the 17th ACM conference on*

Information and knowledge mining - CIKM '08. ACM Press, Napa Valley, California, USA, 571. https://doi.org/10.1145/1458082.1458158

Emine Yilmaz, Javed A Aslam, and Stephen Robertson. 2008. A new rank correlation coefficient for information retrieval. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 587–594.

Justin Zobel. 1998. How reliable are the results of large-scale information retrieval experiments?. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 307–314.