Dynamic Test Collections: Measuring Search Effectiveness on the Live Web

Ian Soboroff
National Institute of Standards and Technology
Gaithersburg, Maryland
ian.soboroff@nist.gov

ABSTRACT

Existing methods for measuring the quality of search algorithms use a static collection of documents. A set of queries and a mapping from the queries to the relevant documents allow the experimenter to see how well different search engines or engine configurations retrieve the correct answers. This methodology assumes that the document set and thus the set of relevant documents are unchanging. In this paper, we abandon the static collection requirement. We begin with a recent TREC collection created from a web crawl and analyze how the documents in that collection have changed over time. We determine how decay of the document collection affects TREC systems, and present the results of an experiment using the decayed collection to measure a live web search system. We employ novel measures of search effectiveness that are robust despite incomplete relevance information. Lastly, we propose a methodology of "collection maintenance" which supports measuring search performance both for a single system and between systems run at different points in time.

Categories and Subject Descriptors: H.3.4 [Information Storage and Retrieval]: *Systems and Software—Performance evaluation*, H.3.5 *Online Information Services*

General Terms: Experimentation, Measurement

Keywords: retrieval test collections

1. INTRODUCTION

Search is the most popular Internet application after email. The proliferation of information available on the web makes search a critical application. The emergence of the web as the world's dominant information environment has created a surge of interest in search, and consequently important advances in search technology. However, it is difficult to measure the effectiveness of web search algorithms because our current methodologies assume that the document collection does not change.

The dominant evaluation procedure is known as the Cranfield or test collection methodology. A test collection consists of a set of documents, a set of information need descriptions (possibly including actual queries), and a mapping of needs to the documents that are relevant to them. In response to each information need, a query is formulated and documents are retrieved from the collection using two (or more) search algorithms. The results of each search are examined to see which documents are relevant and which are not. If significant and noticeable differences in effectiveness are observed, and the differences are consistent across multiple test collections, we can conclude that one search algorithm is better than another.

The Cranfield methodology is so named after the first formalized measurements of search systems conducted by Cleverdon at the College of Aeronautics at Cranfield [12]. It was subsequently refined by many, most notably by Sparck Jones and van Rijsbergen in 1975, and in recent years in the scope of the Text REtrieval Conferences (TREC) [21]. In TREC parlance, the information needs are called "topics", and the mapping of topics to relevant documents is called the "relevance judgments" or "qrels". A "run" is the set of documents retrieved by some search algorithm for each of the topics in the test collection.

The Cranfield paradigm makes several assumptions in order to simplify and operationalize the measurement of search effectiveness. The assumption that we are chiefly concerned with in this paper is that the document collection is static with respect to the runs being measured. A second assumption that we address is that the relevance judgments are complete – all documents are judged with respect to all topics. The TREC pooling process has shown that judgments need not be complete in order to accurately measure the relative performance of two or more systems [22, 19]. Since we are interested in the problem of collection decay, where our document collection and relevance judgments evolve out from under us, we will focus on measures which do not rely on the completeness assumption at all, such as bpref [7].

The Cranfield paradigm further assumes that the information needs (and thus the relevance judgments) are also static, so that for example if one wishes to measure the quality of a "find more like this" facility, it is assumed that the initial set of search results do not change the user's definition of what is relevant. Other assumptions include the notion that the search process can be abstracted away from such vital system details as how queries are created and how results are presented to or indeed used by the end user. In this present work we retain these assumptions as part of the experimental design.

The phenomenon of change and decay on the web has been well studied. Cho and Garcia-Molina tracked 720,000 web pages daily over the course of four months in order to specify design choices for an incremental crawler [9]. Fetterly et al. expanded on that study, tracking more than 150

SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.
ACM 1-59593-369-7/06/0008.

million web pages weekly over 11 weeks and also looking at content changes within pages [14]. Ntoulas et al. crawled 154 different complete sites weekly, and examined change in linkage, changes in page content, and new pages being created [15]. With respect to information disappearing on the web, Bar-Yossef et al. looked closely at soft- and hard-404 errors, and proposed models of web decay based on a Markov chain model of dead link propagation inspired by PageRank [3]. These studies all examine general web crawls in order to understand change on the web as a whole. Brewington and Cybenko looked at the change rates of pages specifically requested by users of a web clipping service [5]. This last work is the closest to what we have done here, except that we are concerned with pages that are relevant to particular search topics. We are also particularly concerned with change specifically as it affects measurements of search quality.

2. PROBLEM DEFINITION

There are many challenges to measuring search effectiveness on the web using the test collection methodology. We believe the key challenge is the requirement of a static document collection. Holding the document collection fixed allows for straightforward reproducibility of results, but is a limiting requirement when we wish to measure search on the web. We propose allowing the document collection to change, while keeping the topics and the relevance judgments initially fixed. Specifically, consider that we have a set of topics and relevance judgments that were constructed for a collection of web pages, and we wish to measure a set of live web search algorithms using them. We would rather avoid making any new relevance judgments if at all possible.

If the documents are allowed to change as they do on the live web, we must account for several possible cases: judged documents will change over time, new pages will appear that are not in the collection, and the runs being measured may be collected at different points in time. First, documents in the collection which have already been judged are likely to have changed, or may no longer exist at all. A relevant document which no longer exists on the web is certainly no longer relevant. If it still exists but its content has changed, we might compare the new document's similarity to the judged document using standard IR similarity measures or a near-duplicate-document measure [6, 10, 4]. In this paper, we choose a simpler strategy which highlights the limits of our approach, and assume that a changed document is no longer relevant until we devote such resources to judge it anew. For documents judged not relevant, we assume that the page remains irrelevant to the topic even if it changes. We call a document *valid* if it has not changed since it was initially judged, or if we have re-examined the document and applied a new relevance judgment. A valid topic is one that has valid documents.

Needless to say, new web pages have come into existence since the initial relevance judgments were compiled, and some of these may be relevant. This can be more or less of a problem for evaluation depending on the timeliness of information desired by the searcher. Rather than make any guess about the relevance of new unjudged documents, we monitor how they are retrieved and how they might affect our determination of the relative effectiveness of the search engines being measured.

Lastly, it may be the case that the runs which we want to compare may have been been executed at different times or on different web crawls. This is likely to be the case when we want to compare live web search engines, but consider also that we may wish to examine several parameter settings of an engine or group of engines on a single large web snapshot, using existing relevance judgments which predate the snapshot. In this paper, we measure a single search engine and are careful to collect our runs within a short period of time, but in general one can use our methodology to compare runs done at different times or compare multiple engines. In these cases, to maximize fairness the set of relevance judgments should be constrained to the intersection of valid documents, so that runs are compared over documents which they all have equal opportunity to rank.

In the next section, we examine the decay of relevance information in an existing test collection. We then illustrate the affect of collection decay over time on our retrieval effectiveness measures using a set of TREC runs. We also present a small experiment measuring retrieval runs from a live web search engine using the decayed relevance judgments. The experiment motivates a maintenance regimen for test collections in order to measure search in dynamic collections.

3. DATA

In this paper we use the GOV2 collection from the TREC terabyte track [11]. The goal of the terabyte track is to scale information retrieval experiments beyond the gigabyte range it typically works in today, and to study how that scaling affects the experimental methodology. The GOV2 collection is a fairly exhaustive crawl of US federal and state government web pages collected in the winter of 2003-4, and contains about 25 million web pages or about 468GB of text. (There is an associated 800GB of image and binary data which are not typically searched in retrieval experiments, but which are used when making relevance judgments.) According to Cho and Garcia-Molina [9], `.gov` pages tend to be much more static than those in other domains. Fetterly et al. [14] confirmed this and also found that whereas generally longer pages change more often, this is not the case in `.gov`. Thus, rates of change that we observe here are likely to be slower than on the web in general.

In TREC 2004 and 2005, two sets of fifty topics were created for the GOV2 collection. These topics are general informational searches, such as might be done by someone compiling a research report. They consist of a short title (often used as a query), a sentence-length description, and a narrative paragraph which defines what the user expects the search system to return. There are 99 topics numbered 701-800; topic 703 was dropped from the 2004 evaluation because no relevant documents were found for it. In those TREC cycles, research teams submitted dozens of runs consisting of the top 10,000 ranked results for each topic according to their search engines. The top hits from two runs from each group were collected into a pool for each topic and judged by the NIST assessor that created the topic. This process yielded 103,368 relevance judgments for these 99 topics. We use this combined topic set in order to maximize the number of usable topics after time is taken into account.

To gather the history of each judged page since the crawl was done, we consulted the Internet Archive.[1] Using their

[1] `http://www.archive.org/`

Wayback Machine service, we downloaded page revisions since February 15th, 2004, the end date of the GOV2 crawl. Only 56,693 of the judged pages were present in the Wayback archives; we will presume for lack of better information that the others disappeared immediately after the GOV2 crawl. We obtained a total of 199,137 page versions, an average of 3.5 revisions per page. Of these, 15,676 page versions were reported as present in the archives, but were not available due to system downtime. Since in this study we work with the timestamps alone, we did not worry about the content of these missing versions.

Figure 1 illustrates the "lifetimes" of topics 701-750 (the TREC 2004 topics), when we assume that a document becomes irrelevant the first time it changes. Each topic's line shows the number of unchanged relevant documents remaining each day. The longest line (topic 739) extends for 369 days. That is in fact the longest lifetime of all 99 topics and represents the extent of historical information available from the Internet Archive at the time of writing. Coverage is more complete for the first 280-300 days. Some topics are more volatile, with their relevant documents disappearing quickly, while others exhibit a more gradual drop-off. At the end of the change history, there are on average 38 relevant documents remaining per topic.

The distribution of times between changes for both relevant and irrelevant pages is shown in Figure 2. The longest gap between changes that we observed was 387 days. 4.5% of gaps represent same-day changes; another 9.4% are 1-day gaps. The average gap between changes is 62.23 days and the median is 49. This supports previous findings that `.gov` pages change slowly. There are also peaks around 60 and 120 days which are due to default page revisit policies in the Internet Archive crawls. According to the first-change heuristic, the number of relevant documents decreases below 50% of the original after 156 days for the average topic.

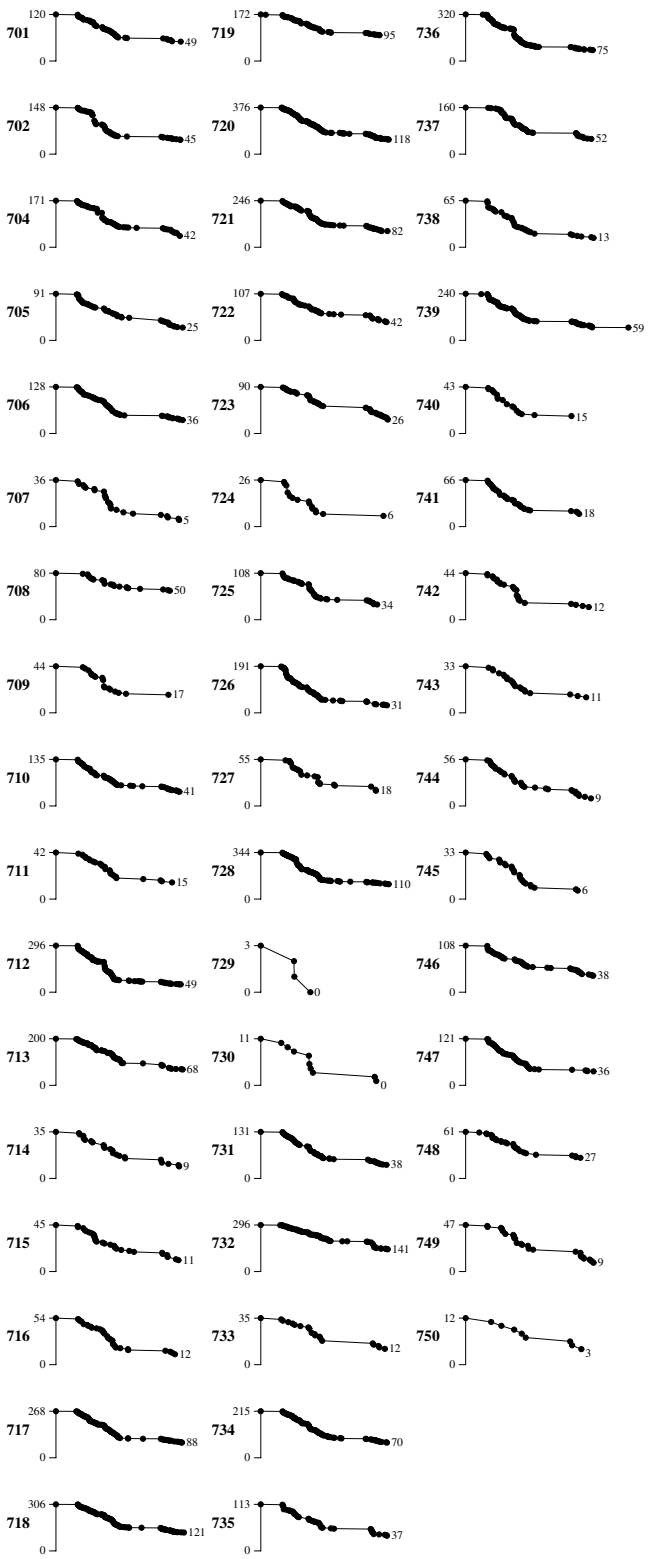

Figure 1: Timelines of the number of valid relevant documents for topics 701-750 of the TREC 2004 Terabyte test collection.

4. MEASURES FOR DECAYED COLLECTIONS

The question of measures is critical when working with dynamic collections. In a sense, the relevance judgments are always incomplete, even less so than in a static test collection. Traditional retrieval metrics such as mean average precision (MAP), precision at the top 10 documents retrieved (P@10), and mean reciprocal rank (MRR) of the first relevant document depend completely on the ranks of the relevant documents which have been retrieved by the system; unjudged retrieved documents are considered to be irrelevant. In our situation here, where so many of the documents are unjudged due to either being outside the collection or having changed since they were judged, such a measure would mostly indicate the sparsity of our relevance data rather than any comparative measure of the runs. Instead, we use a relatively new measure, *bpref*, to compare the runs, and consider carefully which documents we should try to judge in order to improve the picture.

The bpref measure, proposed by Buckley and Voorhees in 2004 [7], computes a preference relation of whether judged relevant documents are retrieved ahead of judged irrelevant documents. Thus, it is based on the relative ranks of judged

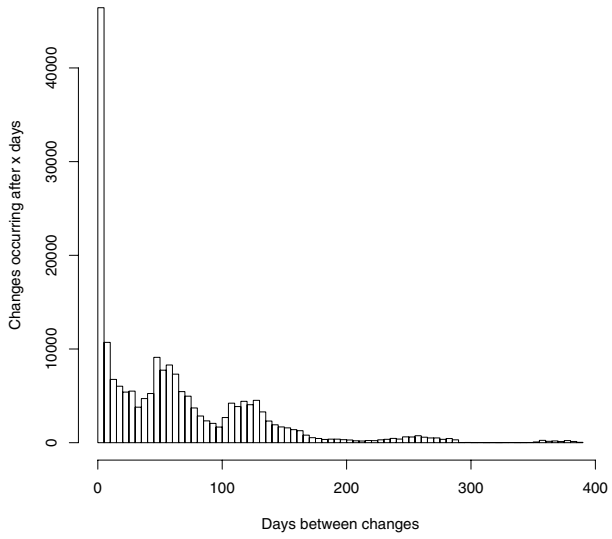

Figure 2: Histogram of time gaps between page changes.

documents only. The bpref measure is defined as[2]

$$\text{bpref} = \frac{1}{R}\sum_{r}(1 - \frac{|n \text{ ranked higher than } r|}{\min(R, N)})$$

where R is the number of judged relevant documents, N is the number of judged irrelevant documents, r is a relevant retrieved document, and n is a member of the first R irrelevant retrieved documents. Bpref can be thought of as the inverse of the fraction of judged irrelevant documents that are retrieved before relevant ones. Bpref and mean average precision are very highly correlated when used with complete judgments. But as judgments are degraded (in Buckley and Voorhees' study, by taking random samples of the judgments and the collection), rankings of systems by bpref still correlate highly to the original ranking, whereas rankings of systems by MAP do not [7].

To better understand how bpref and MAP behave as the collection decays, we examined TREC runs from the 2004 terabyte track at one-week intervals through the collection change history. This is a different approach than Buckley and Voorhees took, in that we are observing the real-world "downsampling" of the collection over time. Furthermore, we are able to compare bpref and MAP in the TREC terabyte collections which were not available to them.

Figure 3 shows the MAP and bpref scores for each of the 70 runs from TREC 2004 when measured against the qrels that remain valid each week. The absolute value of the score is not important, but that shape of each curve is. As the collection decays, MAP decreases. Bpref fluctuates somewhat, and actually increases as we lose more and more relevance information. This much is consistent with the findings of Buckley and Voorhees. At the end of the graphs, bpref drops

[2]This definition of bpref corrects a bug in [7] and follows the actual implementation in `trec_eval` version 8.0; see the file `bpref_bug` in the `trec_eval` distribution for details.

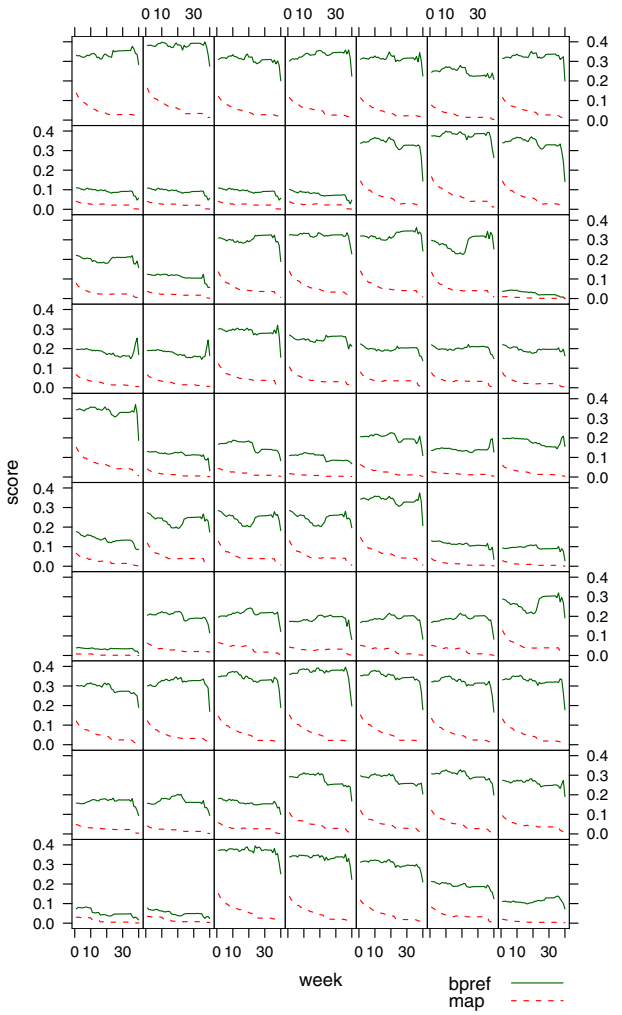

Figure 3: MAP and bpref scores for the TREC 2004 runs, scored according to the qrels remaining after each week. Each subgraph shows a single run. Past week 41, insufficient judged documents were retrieved to use even bpref.

sharply because we have only 103 judged documents total remaining to be retrieved, out of 28,102 in week 1. This is a smaller percentage than Buckley and Voorhees examined.

Even though bpref does show some fluctuations as relevance information decays, the relative ordering of systems according to bpref remains fairly close to the order in week 1. Figure 4 shows the correlation of weekly system rankings to the original ranking using Kendall's tau. Whereas the correlation using MAP falls below 0.9 at 18 weeks, the lowest correlation for bpref during the entire period is 0.91 at week 41. Thus, when the systems are compared using the bpref measure, we arrive at a consistent ordering despite severe decay in relevance data.

Note that using TREC runs to illustrate the effect of collection decay is anachronistic because the runs were performed on the GOV2 collection as it was initially compiled, and the decay we observe happens after this point. In an operational setting, the runs always come from a document

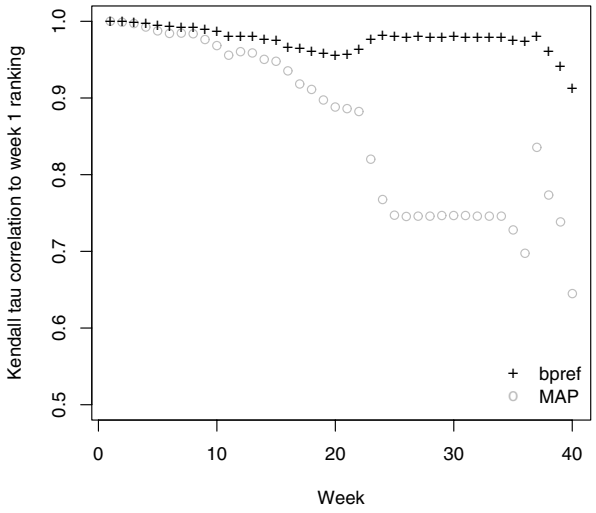

Figure 4: Kendall's tau correlation of weekly system rankings to the week 1 ranking. bpref agrees more closely than MAP to the original ranking as the collection degrades.

collection that is more recent than the relevance judgments. In the next section we conduct just such an experiment.

5. MEASURING SEARCH ENGINES

We ran a small experiment to test the methodology for measuring "live web" search using a popular web search engine. This experiment tests if shorter or longer queries give better performance for the engine in question. We hypothesize that shorter queries will be more effective since most search engines combine terms in a noisy-AND formula.

As stated above, the 99 TREC terabyte topics include a short title field and a sentence-length description field. For the short queries, we used the title field. For the long queries, we added description field words which were not present in the title. Stop words were removed from both long and short queries. In some cases, the description text includes phrases in quotation marks; we retained these quoted phrases since the search engine allows this as an operator but removed most other punctuation. Long queries were limited to nine terms (counting quoted phrases as a single term). All queries included a search restriction requiring that hits come from `.gov` sites.

For each query, we attempted to retrieve the top 100 documents. This was a compromise between the limitations of the search engine API and the need for our rankings to go deep enough to find judged documents. For some queries the search engine returned less than 100 documents. For each search result, we checked to see if the URL corresponded to a document in the GOV2 collection, since these are the only documents for which we have judgments. If we were trying to measure multiple search engines or a single search engine over time, we would need to restrict ourselves to the intersection of retrieved documents between the engines in order to ensure a fair collection.

Run	Retr.	In GOV2	Judged		Rel	
short-q	9375	2616 (28%)	1800	(19%)	888	(9%)
	in decayed qrels:		1019	(11%)	107	(1%)
long-q	9080	2318 (26%)	951	(10%)	425	(5%)
	in decayed qrels:		584	(6%)	58	(0.6%)

Table 1: Number of retrieved documents present in the collection, judged, and relevant for the two runs.

We also observed a somewhat amusing phenomenon, which is that if you search for TREC topics after the TREC conference cycle is completed, you tend to find the TREC topic file high up in the ranking. Fortunately, these pages are not in the collection and are thus ignored.

5.1 Determining valid topics

We first look at how many documents returned by the search engine occur in the collection, how many of those were judged, and how many were judged relevant. Table 1 shows that less than a third of retrieved documents are in the collection, one-half to two-thirds of these were judged, and one-third to one-half of these last are relevant. The long-query run finds somewhat fewer GOV2 documents and only half as many relevant documents as the short-query run, a clue that our hypothesis may turn out to be correct.

These searches were conducted after the epoch of our historical data on the relevance judgments, but for simplicity we will assume that the revision data we have is current. Recall that we assume pessimistically that the first change to a relevant page reverts the page to an unjudged status. We generate the set of relevance judgments that corresponds to pages that have remained unchanged. The lines labeled "decayed qrels" in Table 1 indicate how many judged and relevant documents were retrieved. The short-query run retrieved no judged documents for topics 717, 770, 779, 793, and 796. The long-query run retrieved no judged documents for 702[3], 705, 739, 750, 758, 763, 770, 779, 780, 787, and 800. We are left with 85 valid topics to compare the two runs, a good-sized topic set.

5.2 Per-topic results and analysis

Over the 85 topics, the short-query run has an average bpref of 0.0304, and the long-query run scores 0.0161, supporting our hypothesis that short queries perform better than long queries for this search engine.

However, despite having 85 valid topics, we still have very little data with which to measure these two runs. The short-query run has on average only 11 judged and 1.2 relevant documents per topic; the long-query run, 6.7 judged and 0.7 relevant. Furthermore, the short-query run finds no known relevant documents for 40 of these topics, and the long-query run finds no relevant for 53.

Even though bpref is designed for our degraded collection scenario, by using topics with only one or two judged documents we are forcing bpref to its corner case, and thus we should read it with care. In Table 2, we focus on 27 topics for which both runs found at least one relevant document.

The average within this subset still supports the hypothesis, but if we look closer at the per-topic results we should

[3]In fact, the search engine returned only one document for the long query for topic 702, and this hit was the TREC topic file.

		short-q		long-q	
	#rel	rel.ret	bpref	rel.ret	bpref
701	49	1	0.0171	1	0.0196
708	50	5	0.0936	6	0.1164
710	41	1	0.0190	1	0.0184
712	49	1	0.0196	2	0.0400
713	68	1	0.0147	2	0.0277
719	95	5	0.0465	3	0.0316
720	118	2	0.0155	1	0.0084
721	82	1	0.0115	1	0.0115
722	42	1	0.0232	1	0.0238
725	34	8	0.1696	1	0.0216
731	38	2	0.0492	2	0.0492
732	141	1	0.0067	1	0.0070
736	75	3	0.0389	3	0.0395
741	18	2	0.0988	2	0.0988
746	38	1	0.0263	3	0.0789
752	75	5	0.0645	2	0.0219
761	41	6	0.1374	1	0.0238
766	22	3	0.1364	1	0.0413
767	102	3	0.0291	2	0.0193
771	46	2	0.0359	2	0.0388
776	23	1	0.0302	1	0.0397
777	39	2	0.0388	2	0.0440
782	33	4	0.1010	3	0.0735
791	10	3	0.2300	1	0.1000
797	37	1	0.0219	1	0.0263
798	8	1	0.0000	2	0.0312
799	33	3	0.0854	2	0.0588
Avg		2.6	0.0578	1.9	0.0411

Table 2: Per-topic measures for each run. "#rel" is the number of relevant documents in the decayed qrels. "rel.ret" is the number of relevant retrieved for that topic.

be cautious in accepting that conclusion. The number of retrieved documents judged to be relevant is between two and three on average, and so the bpref value is based on very few pairs of relevant and irrelevant documents. While the short-query run does find more relevant documents on average, and has most of the highest bpref scores per topic, the long-query run actually beats the short query run on 13 of the topics. The short-query run wins for 11 topics, but with a higher bpref difference. Although the sample is small, we observe that a one-sided Wilcoxon test is not significant ($p = 0.22$), but a one-sided paired t-test is ($p = 0.04$).

We conclude from this experiment, somewhat cautiously, that short queries do work better than long queries for this search engine; short queries tended to return more relevant documents, but it's hard to measure the quality of the ranking from so few documents. The situation would improve if there were 3 to 5 more judged relevant documents for each run in each topic, as we show in the next section.

6. MAINTAINING TEST COLLECTIONS

In their recent paper on building test collections incrementally, Carterette and Allan propose a method for choosing which documents to judge by giving priority to documents whose relevance will expose the greatest difference among the systems. Using the MAP measure, they compute the potential difference in MAP if an unjudged document were to become relevant. To avoid bias, the process continues until adding more relevant documents stops affecting the relative performance of the systems [8]. Fundamentally, this approach builds on the results of Zobel [22], who suspected that the TREC relevance judgments were incomplete, but found that discovering more relevant documents did not affect the relative performance of systems very much. Carterette and Allan's approach takes a set of judgments which are too coarse to compare systems with, and moves them to the point observed by Zobel in an optimal way.

We do the same thing here with two changes. First, we would rather work with bpref than MAP, since bpref requires many fewer relevance judgments to achieve stability. Choosing bpref also means that we don't need to optimize the document selection process as strongly as we would for MAP, since bpref only considers relative ranks of judged documents.

Second, we identify three types of unjudged documents. Because we are maintaining an existing collection rather than building a new one, we have documents which are in the original collection and were unjudged. Since our starting point here is a TREC collection built by pooling a large number of system outputs, we should probably opt not to examine these documents.

We also have unjudged documents that lie outside the original collection and should be candidates for examination. When selecting out-of-collection documents to judge, it is important to avoid bias in favor of one run or another. The TREC pooling process, while not efficient, places a high priority on avoiding bias towards particular systems. We can maximize impact on the measure and avoid bias by selecting documents retrieved highly by both runs which have a high coefficient of variance in the rank retrieved.

Lastly, we have previously judged documents which have changed. We have chosen to invalidate their relevance judgments, but in particular previously-relevant documents would be a good place to start recovering relevance information. This would also follow if we had chosen to invalidate relevance judgments according to document similarity measures.

In the case of previously-judged documents, we can choose documents to judge in an unbiased way by selecting them in order of most-recent change. In Figure 5, we simulate the re-judging process assuming that documents regain their old relevance value, and show the effect on bpref scores. Each point on the x-axis represents recovering one document for each run with the next latest change timestamp, assigning them their original relevance value, and recomputing the bpref score. We can see that for most topics, our conclusion is unchanged: either the runs are indistinguishable in effectiveness, or their initial effectiveness ranking is preserved. For some inconclusive topics we gain enough information to distinguish them after re-judging very few "expired" documents. It is also clear that we should first focus on those topics with the fewest retrieved relevant documents.

Once these topics have been stabilized, maintenance effort should be directed at reviving the 14 topics we were forced to discard because they retrieved no judged documents. Priority should be given to topics with more than 20 retrieved documents which are unjudged due to page revisions, since in the 27 topic subset we see 5-20 judged irrelevant documents retrieved for every judged relevant one. Further, we should choose topics where a large number of those revisions

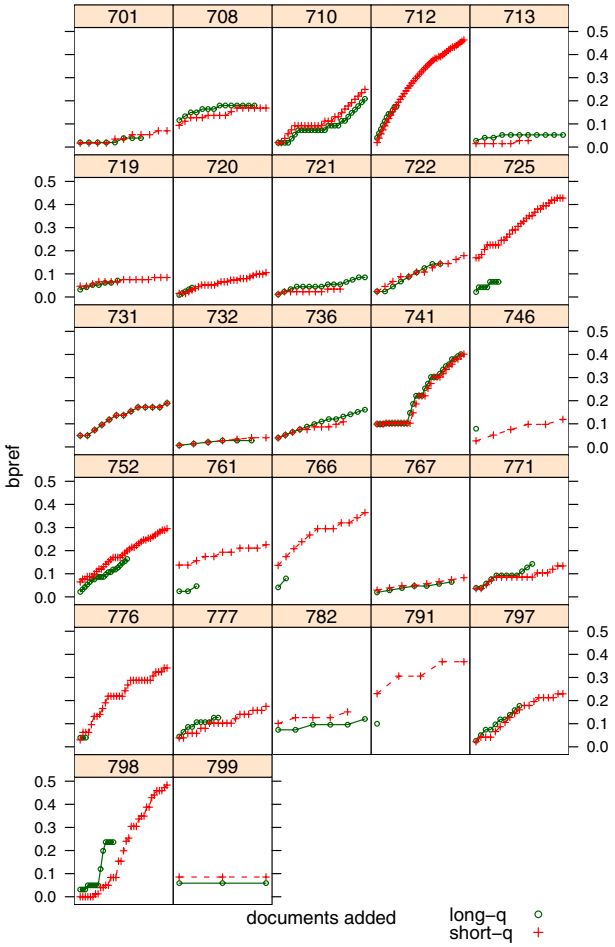

Figure 5: Change in bpref in the 27-topic subset as changed documents are re-judged according to their original relevance value.

affected relevant documents, to ensure that we will recover relevant documents without needing to examine too many irrelevant ones.

7. CONCLUSION

We have shown that static test collections can be used to measure search in a changing document collection such as the live web by tracking changes in judged documents, applying heuristics to determine the decay of relevance information, and carefully re-examining the "old" relevant documents as well as the unjudged documents retrieved in each experiment. Test collections with large sets of relevance judgments remain usable for a long time; the documents we use here were nearly two years old when these experiments were run. As judged documents change, measures such as bpref which work with incomplete information can be used with little or no additional relevance assessment.

We propose the following approach for maintaining a test collection of topics and relevance judgments atop a changing web. First, one must consider how the initial test collection was created. The collection we address in this paper was built as part of a collaborative process which typically involves ten or twenty research teams using different systems with various tuned parameter settings and which may also include manually collected search results in addition to automatic system rankings. Equivalent approaches include pooling the outputs of a smaller but highly diverse range of retrieval methods [18], or iterative search-and-judge procedures [13]. Test collections built using these methods avoid bias towards any particular search strategy by looking broadly and deeply into the collection for relevant documents.

Alternatively, the test collection might come out of an industry search environment, for example a search engine company, or an organization attempting to tune an intranet search engine. In this case there may be only one or two search algorithms contributing documents to judge, and one should be concerned about bias. If the relevance judgments are derived from a single retrieval strategy, then a new approach will retrieve many unjudged documents. The procedure of Carterette and Allan [8] may be useful here.

In either case, as the document collection changes over time, there are several indicators to watch:

The rate at which judged documents change as newer web crawls are done, combined with a heuristic for deciding when a page's relevance judgment no longer applies. Our heuristic is based on the rate of change; alternatively it might be based on a similarity or fingerprinting comparison. The rate of change should be observed per topic, rather than per-document.

The number of topics with only one or two retrieved documents that have valid relevance judgments. These topics will need some maintenance in the form of additional judgments in order to be useful. Alternatively, we may decide that topics which fail to retrieve any judged documents can be retired or redone from scratch.

The number of retrieved documents whose relevance values expired due to changes in the page. If we are still retrieving these pages they are good candidates for re-judging, particularly if they used to be relevant.

The number of retrieved documents which lie outside the collection used to create the initial relevance judgments. These documents are unjudged and would not have been judged initially, and this number will only grow over time. If the researcher has several different retrieval algorithms at hand, these can be pooled and judged using the processes described in any of the above mentioned papers.

The total number of valid topics. If fewer than 30 topics are usable due to relevance decay, then unusable topics should be patched. One can look to guidelines for topic-set size such as [20, 17], but keep in mind that the experimental conditions may necessitate more topics. More topics are always better. Sanderson and Zobel suggest that having many topics judged less is better than having fewer topics judged more completely [16].

By following these indicators through frequent, repeated experiments, a test collection may be maintained over the live web or other dynamic collection and its usable lifetime extended considerably. As topics decay, one can re-examine past documents, spend resources to judge new documents, or retire topics in favor of developing new ones. Maintenance is cheaper in terms of topic development and relevance assessment time, and permits the comparison of runs from different versions of the collection.

8. FUTURE WORK

The results presented here are somewhat preliminary, and there are a number of improvements and future directions we would like to explore.

We plan to conduct a fuller examination of the bpref measure, and system rankings in general in dynamic collections. We would also like to study explicit measures for detecting bias in selecting documents for relevance assessment. The notion of incremental and maintained test collections makes such measures critically important.

Lastly, we have not fully considered the ramifications of comparing runs done at different points in time on different versions of the collection. When a difference is discovered, is it due to algorithmic advantage or differences in the statistical distribution of features in the collection?

9. REFERENCES

[1] *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*, Melbourne, Australia, August 1998. ACM Press.

[2] *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)*, Sheffield, UK, July 2004. ACM Press.

[3] Ziv Bar-Yossef, Andrei Z. Broder, Ravi Kumar, and Andrew Tomkins. Sic transit gloria telae: Towards an understading of the web's decay. In *Proceedings of the Thirteenth International World Wide Web Conference (WWW 2004)*, pages 328–337, New York, NY, May 2004.

[4] Yaniv Bernstein and Justin Zobel. A scalable system for identifying co-derivative documents. In *Proceedings of the Eleventh Symposium on String Processing and Information Retrieval (SPIRE 2004)*, Padova, Italy, October 2004.

[5] Brian E. Brewington and George Cybenko. How dynamic is the web? In *Proceedings of the 9th International WWW Conference*, pages 257–276, Amsterdam, The Netherlands, May 2000.

[6] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29(8/13):1157–1166, 1997.

[7] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)* [2], pages 25–32.

[8] Ben Carterette and James Allan. Incremental test collections. In *Proceedings of the Fourteenth ACM Conference on Information and Knowledge Management (CIKM 2005)*, Bremen, Germany, November 2005.

[9] Junghoo Cho and Hector Garcia-Molina. The evolution of the web and implications for an incremental crawler. In *Proceedings of the 26th International Conference on Very Large Databases (VLDB 2000)*, pages 200–209, September 2000.

[10] Abdur Chowdhury, Ophir Frieder, David Grossman, and Mary Catherine McCabe. Collection statistics for fast duplicate document detection. *ACM Transactions on Information Systems*, 20(2):171–191, April 2002.

[11] Charles L. A. Clarke, Ian Soboroff, and Nick Craswell. Overview of the TREC 2004 terabyte track. In E. M. Voorhees and L. P. Buckland, editors, *Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004)*, Gaithersburg, MD, November 2004.

[12] C. W. Cleverdon. The cranfield tests on index langauge devices. In *Aslib Proceedings*, volume 19, pages 173–192, 1967.

[13] Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. Efficient construction of large test collections. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)* [1], pages 282–289.

[14] Dennis Fetterly, Mark Manasse, Marc Najork, and Janet L. Wiener. A large-scale study of the evolution of web pages. *Software Practice and Experience*, 34:213–237, 2004.

[15] Alexandros Ntoulas, Junghoo Cho, and Christopher Olston. What's new on the web? the evolution of the web from a search engine perspective. In *Proceedings of the Thirteenth International World Wide Web Conference (WWW 2004)*, pages 1–12, New York, May 2004.

[16] Mark Sanderson and Justin Zobel. Information retrieval system evaluation: Effort, sensitivity, and reliability. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005)*, pages 162–169, Salvador, Brazil, August 2005. ACM Press.

[17] Ian Soboroff. On evaluating web search with very few relevant documents. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)* [2], pages 530–531.

[18] Ian Soboroff and Stephen Robertson. Building a filtering test collection for TREC 2002. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003)*, pages 243–250, Toronto, Canada, July 2003. ACM Press.

[19] Ellen M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)* [1].

[20] Ellen M. Voorhees and Chris Buckley. The effect of topic set size on retrieval experiment error. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, pages 316–323, Tampere, Finland, August 2002. ACM Press.

[21] Ellen M. Voorhees and Donna K. Harman, editors. *TREC: Experiments in Information Retrieval Evaluation*. MIT Press, 2005.

[22] Justin Zobel. How reliable are the results of large-scale retrieval experiments? In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)* [1], pages 307–314.