UQV100: A Test Collection with Query Variability

Peter Bailey
Microsoft,
Australia
pbailey@microsoft.com

Alistair Moffat
The University of Melbourne,
Australia
ammoffat@unimelb.edu.au

Falk Scholer
RMIT University,
Australia
falk.scholer@rmit.edu.au

Paul Thomas
Microsoft,
Australia
pathom@microsoft.com

ABSTRACT

We describe the UQV100 test collection, designed to incorporate variability from users. Information need "backstories" were written for 100 topics (or sub-topics) from the TREC 2013 and 2014 Web Tracks. Crowd workers were asked to read the backstories, and provide the queries they would use; plus effort estimates of how many useful documents they would have to read to satisfy the need. A total of 10,835 queries were collected from 263 workers. After normalization and spell-correction, 5,764 unique variations remained; these were then used to construct a document pool via Indri-BM25 over the ClueWeb12-B corpus. Qualified crowd workers made relevance judgments relative to the backstories, using a relevance scale similar to the original TREC approach; first to a pool depth of ten per query, then deeper on a set of targeted documents.

The backstories, query variations, normalized and spell-corrected queries, effort estimates, run outputs, and relevance judgments are made available collectively as the UQV100 test collection. We also make available the judging guidelines and the gold hits we used for crowd-worker qualification and spam detection.

We believe this test collection will unlock new opportunities for novel investigations and analysis, including for problems such as task-intent retrieval performance and consistency (independent of query variation), query clustering, query difficulty prediction, and relevance feedback, among others.

1. INTRODUCTION

Test collection-based evaluation is the most widely used methodology for measuring the effectiveness of information retrieval systems. A typical test collection consists of a set of queries, a collection of documents to search over, and a set of relevance judgments that indicate, for query-document pairs, whether the document was a topically related resource for that query. To evaluate a search system, a ranked answer list is generated for each query, and the relevance of each item is determined with reference to the available judgments [3]. Finally, the relevance information is condensed into a single number, based on a chosen effectiveness metric, which may take into account features such as the number of relevant answers that were retrieved, at what positions in the ranked list the relevant answers were located, and so on.

Collection-based evaluation has several advantages: it supports reproducible experimentation; and, while constructing a test collection is typically resource-intensive in terms of time and labor – particularly the creation of relevance judgments – it is then inexpensive to run any number of further experiments using the same framework. However, there are also limitations in terms of the realism of the evaluation. For example, the user is almost entirely removed from the evaluation, and is represented via a single search query that instantiates the underlying information need.

The TREC Query Track studied the impact on effectiveness evaluation of multiple user-generated search queries, all aiming to resolve the same underlying information need. Analysis concluded that "topics are extremely variable; queries dealing with the same topic are extremely variable" while "systems were only somewhat variable" [4]. A specific concern is the coverage that existing collections offer when widely varying queries are admitted, even for a single topic. Moffat et al. [10] examine the adequacy of relevance judgments in a standard single-query test collection in the face of such query variations; their results demonstrate that a large proportion of documents retrieved for the variable queries are unjudged, including many near the top of their rankings. Our purpose in this work is to address the challenge posed by this earlier work, and develop a test collection that explicitly includes query variability as a factor.

2. THE COLLECTION

Corpus, Topics, and Backstories ClueWeb12-B [11] was used as an underlying corpus due to its wide availability, scale, coverage of modern Web documents, and additional annotations. One hundred topics from the 2013 and 2014 TREC Web tracks [5, 6] were taken as the basis for information-need statements (background stories, or "backstories", written by us), in a manner similar to that described by Bailey et al. [2]. Topic numbers 201–300 were used; where a topic contained subtopics, one of them was selected as the focus of the backstory, and the others were ignored. An example backstory is shown below, for topic 215 (*maryland department of natural resources*), subtopic 2 (*How do you get a Maryland fishing license?*):

> *Having heard of the pristine environment in Maryland, you have long dreamed of taking a fishing holiday there. However, you think that you may need a fishing license in Maryland. How do you get one?*

SIGIR '16, July 17–21, 2016, Pisa, Italy

ACM 978-1-4503-4069-4/16/07 … $15.00.
http://dx.doi.org/10.1145/2911451.2914671.

	Mean	Min.	Max.
Raw queries per worker	41.2	1	100
Raw queries per backstory	72.5	35	106
Normalized queries per backstory	61.0	22	101
Spell corrected queries per backstory	57.7	19	101

Table 1: Query counts through the data simplification process.

Each backstory provides a brief motivating context, hopefully with some degree of realism, that helps individuals imagine themselves in a similar information-seeking situation and informs their query and effort responses [2]. The wording makes use of anaphora (co-referencing entities via pronouns) to avoid offering obvious queries.

Query Variations and Effort Estimates We developed two crowd worker interfaces. Both presented the backstory, and then asked the worker to enter the first query they would use to access information via a search engine in response to the backstory, and for estimates of the effort (in terms of number of useful documents, and number of queries) that they anticipated needing to satisfy the information need. We varied from the radio-button interface described by Bailey et al. [2]. In one interface we asked for effort estimates using graphical slider widgets ranging from 0 to 101 for the estimate of the number of useful documents required, and 1 to 11 for the number of queries that would need to be issued. In the other interface, we provided text entry fields requiring integer non-negative numbers. These two interfaces were released to two different English-speaking crowds.

As is often the case with crowds, a number of low quality workers participated. A mixture of methods was used to identify suspicious data, including noting workers who entered the same query text for multiple responses, or who provided undeviating effort estimates regardless of the backstory. Data from these workers was removed. At the conclusion of the cleaning process we had data from 263 individuals, spread across the 100 backstories. The count of workers per backstory averaged 108 (min: 105, max: 113) and there were a total of 10,835 individual queries and effort estimates provided. Each query was normalized to lowercase, with extraneous whitespace and trailing punctuation removed, and passed through the Bing search engine's spelling service to generate a final canonical form of each worker's query. This was done to avoid differences arising in how systems might handle such basic query normalization and spell correction of the query variations, and to reflect how queries would be pre-processed in a live system. It also reduced the total number of unique query variations (Table 1).

For example, for the Maryland fishing license backstory listed earlier, there were 53 unique spell-corrected query variations obtained. The average effort estimate in terms of useful documents required was 2.7 (which lies in the lowest decile of the 100 backstories). There were 13 variations which occurred more than once, of which 7 occurred only twice. The most popular query was "*maryland fishing license*" which occurred 14 times. Many of the single occurrence variations are expressed in more natural language forms, such as "*how do i get a fishing license in maryland*", "*who can get a fishing license in maryland*", and "*is a fishing license needed in maryland*". Only 8% of occurrences were identical to the corresponding TREC title query. In these regards, our collection process provided similar query diversity as is reported by Bailey et al. [2].

Relevance Judgments The 2013 and 2014 TREC Web collections include NIST-generated relevance judgments covering the 100 topic/subtopic pairs, based on the sets of 61 and 30 participating system runs, many of which will have run the nominal "title" query associated with the topic. The TREC judgments use a six-category scale with four ordinal relevance levels for informational tasks; a category for navigational tasks; and a final category for spam [6]. Analysis of the NIST relevance judgments indicates that these definitions were not strictly followed. For example, topic 298 had approximately 50% of the documents given the *Home Page* label for the entire collection, all of which came from pages on the official website of the Jehovah's Witnesses; yet many of these pages are only of marginal relevance. We adapted the TREC category names and descriptions to provide slightly clearer expression of what each category label should capture, and added more information about what a searcher might do after reading a document in this category. Our assessors would receive little training, and so the judging guidelines and qualification test had to suffice. The new rating categories were: *Essential*; *Very Useful*; *Mostly Useful*; *Slightly Useful*; *Not Useful*; and *Junk*. The lowest, rating *Slightly Useful*, includes good-quality pages containing links to documents that would probably have useful information, even if they did not include the requisite information themselves. We believe this reflects real user behavior in web search activities. The full description of each rating is available in the judging guidelines provided as part of the collection.

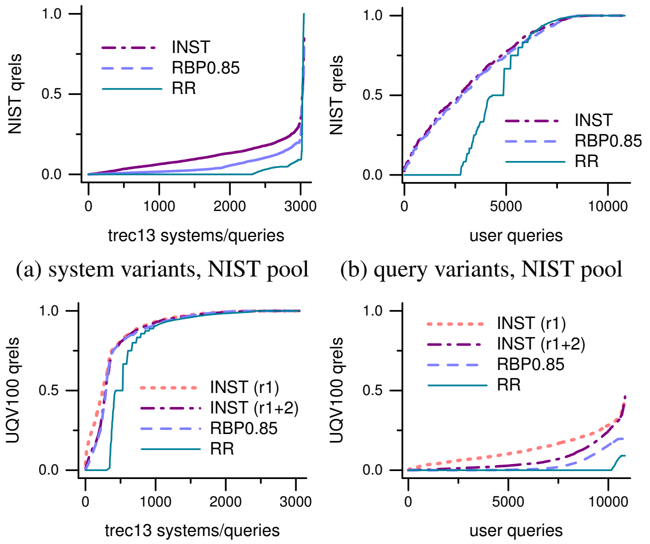
For effectiveness metrics that use ordinal (graded) relevance, the *Junk* category should be merged with the *Not Useful* category, after which the scale can be applied directly as a five-level ordinal relevance scale. For metrics that require binary relevance judgments, our recommendation is to fold the *Junk* and *Not Useful* categories into a single *Not Relevant* category, and to fold the remaining levels into a single *Relevant* category.

Quality Control A total of 120 documents were randomly selected from those topic/subtopic-document pairs available in the original TREC qrels files that did not appear in our top-10 pools. These were stratified-sampled to obtain an even spread of TREC relevance labels. Each of the topic/subtopics was replaced by the corresponding backstory, and then each document was judged by two of the authors with respect to the judging guidelines. Any disagreements were discussed, and a final label agreed. Five of the pairs were subsequently discarded, due to genuine disagreement and potential for confusion, leaving a set of 115 "gold" judgments to draw from.

Judgments were then sourced via a crowd worker platform. From the gold set of backstory-document labels, a set of 27 were selected, again attempting to stratify as evenly as possible across the label categories. Before being asked to provide ratings, workers had to correctly rate documents from at least four of seven randomly chosen test questions (from the set of 27), and had up to three attempts. Additional random injections from the remaining gold judgments were used by the crowd-sourcing system to calibrate ongoing worker quality levels, and eliminate workers failing to meet standards without incurring unnecessary costs. There was no intentional overlap between the workers who provided the original query variations and effort estimates and those carrying out judging.

Aptness of Relevance Judgments A critical question that arises is that of how to assess what we denote as *aptness*, the extent to which a set of relevance judgments is fit for scoring a set of runs generated by systems and/or queries. Moffat and Zobel [7] introduce the notion of a *residual*, a numeric quantification of the extent of the (upward) uncertainty of a score that arises from the presence of unjudged documents. Residuals can be calculated for any weighted-precision effectiveness metric, including Reciprocal Rank (RR), by taking the difference between the "all unjudged documents are non-relevant" and the "all unjudged documents are maximally relevant" run scores. The greater the residual, the less apt the judgments.

Figure 1a illustrates this approach. To construct the graph, the



(a) system variants, NIST pool

(b) query variants, NIST pool

(c) system variants, UQV pool

(d) query variants, UQV pool

Figure 1: Residuals for (left column) $50 \times 61 = 3{,}050$ TREC runs from 2013, and for (right column) 10,835 user-generated queries including repeats. Residual scores for three metrics are shown in each pane, computed using (top row) the NIST qrels selecting only the query subtopic used to generate the corresponding backstory, and using (bottom row) the new UQV100 qrels. All runs are relative to the ClueWeb12-B collection, and all metrics were evaluated over the top 200 documents retrieved; with Indri/BM25 used to construct runs for the user query variants in the right column. In the bottom row, the additional INST line shows the first round of judgments; both rounds were used for RBP and RR.

$50 \times 61 = 3{,}050$ combinations between TREC 2013 contributing systems and topics were scored using the NIST-provided qrels file and four different metrics, including the INST mechanism of Bailey et al. [2], and residuals computed. Those residual were then independently sorted for each metric, and plotted by ascending value. The relatively low residuals shown in Figure 1a for the RR and RBP0.85 provides evidence of the aptness of the judgments for the evaluation of the contributing runs. But when using the user-supplied T values (which average at 4.7), INST has an expected search depth of 8.6, and the higher residuals show that the available judgments are not such a good fit. Moffat et al. [9] discuss INST in detail, including the relationship between T, expected search depth, and residual.

Use of NIST Judgments for Query Variants Figure 1b was generated using the same methodology as Figure 1a, but using the 10,385 user queries (5,764 distinct). The NIST-supplied qrels are again used; what is clear is that the judgments are no longer apt, and that none of these three metrics, not even RR, should be used to generate effectiveness scores for these runs. Indeed, residuals of over 0.5 arise for more than half of the queries. Even if by chance the lower-end scores for some metric displayed some particular attribute (for example, gave rise to a statistically significant system comparison), it would be risky to trust such a conclusion.

Working With INST We are interested in exploring the implications of using Bailey et al.'s INST metric when forming a test collection. INST is designed to be sensitive to the user's search goal, and is parameterized by a value T, the expected number of useful documents that will be required. For example, when $T = 1$, the user is anticipating needing to retrieve one relevant document, and the information need may well be navigational or factoid in nature. Higher values of T correspond to richer information needs.

Depth	Total pool	Per run	Per document
1	2,741	0.25	0.25
2	5,157	0.48	0.24
5	11,755	1.08	0.22
10	21,895	2.02	0.20
20	40,478	3.74	0.19
50	91,556	8.45	0.17
100	170,166	15.71	0.16
200	316,171	29.18	0.15

Table 2: Pool sizes over 10,835 topic-query combinations. The final two columns give per run and per retrieved document averages.

The effort-influenced variability embedded in INST means that it is desirable to judge different topics to different depths, rather than use a uniform pool. Moreover, INST is an adaptive metric and becomes increasingly top-weighted as relevant documents are encountered. In combination, these two factors suggest a two-stage judgment process: first, uniform pooling to a relatively shallow depth, followed by a *gap analysis* to determine the topics, and document within topics, where further judgment effort should be applied in order to ensure that all residuals were broadly comparable to within the limits of the judgment budget. In doing so, we are in part implementing the mechanism described by Moffat et al. [8].

Collecting New Judgments: Round 1 Table 2 lists the uniform pool sizes that were generated from the user queries, with the final two columns normalized first on a per topic-query basis, and then second on a per-document retrieved basis. Despite the fact that there are nominally only 100 different information needs covered by these queries, and despite the fact that on average each query occurs twice, even at depth 10 fully 20% of the documents retrieved must be judged in order to cover the pool.

A first round of crowd-worker judgments was collected using a uniform pool depth of 10 and three-way overlap judging, based on the use of Indri/BM25 to construct runs against the ClueWeb12-B corpus for each of the 10,385 spell-corrected query variants (5,764 distinct). A median label from the (in almost all cases) three individual judge labels (after removing ratings that indicated a judge was unable to provide a label due to page load failure or other reasons) was assigned. Results are reported using this median label; in the collection data resources we also provide a re-estimated label based on the Community BCC algorithm described by Venanzi et al. [12].

Collecting New Judgments: Round 2 We then scored each run using INST, and for each document-topic combination, computed the sum over the runs of the residuals associated with that document [8]. That list of document-topic pairs was then sorted into decreasing order, and a further 5,501 documents taken from the front of it and judged, with aggregate per-document residuals of between 1.653 and 0.051. This targeted process had the effect of applying deeper pooling on topics with higher T values, and on topics where there were comparatively few relevant documents identified. Note that the adaptive and goal-sensitive nature of INST means that we are unable to determine which additional judgments were required without the initial round of judgments being completed.

Figure 1c and 1d show the application of the new qrels, with two lines plotted for INST showing the further improvement in average residual delivered by the Round 2 judgments. Using the full set of

UQV100 judgments, more precise measurement of effectiveness for the query variants can be achieved (Figure 1d), because that facet is the basis of the pooling. But the ability to accurately evaluate different retrieval systems has been significantly compromised (Figure 1c) compared to the NIST judgments arising from pooling across systems. Determining how best to cater to the cross-product of these competing requirements is a clear direction for future work.

As a future extension we plan to incorporate further variability by using one or more systems that vary considerably from the Indri/BM25 system we have used here. We will update the collection progressively as further labels become available.

Observations on Judgments The final set of 27,396 judgments represent all 100 topics, and reflect contributions from 179 judges, after qualification tests and anti-spam measures were applied. Each topic was judged by 42 judges on average (range 17–68). With each topic/document pair having been judged (for the most part) three times, we computed an inter-assessor agreement as Krippendorff's $\alpha = 0.26$, or 0.24 with binary labels.

We emphasize that the TREC and UQV100 judgments are different, should not be aggregated, and are not interchangeable. TREC judgments were created by well-qualified, specialist analysts against TREC judging guidelines and topic descriptions; each judge would read hundreds of documents per topic. UQV100 judgments were made by judges from a more diverse population, working under very different constraints, against UQV100 judging guidelines and backstories – which may incur some topic intent drift – and judges would read many fewer documents per topic on average. UQV100 judgments were also subject to automated quality control tests as well as post-hoc data cleaning.

3. POTENTIAL USES

We present three possible uses of UQV100, beyond the obvious example of using it to investigate system performance on a per query or per backstory basis. When assessing the latter, we recommend averaging the query-level performance across all query variations belonging to a specific backstory first, and then averaging these across the 100 backstories (double-averaging).

Query Clustering While commercial web search engines have large query logs and can use these to examine query–query relationships via within-session re-formulations or co-clicks on the web graph, there is no equivalent in current test collections available to the broader IR community. The UQV100 collection supports some new investigations in query clustering by having so many query variations per backstory. As observed in an extensive investigation of query clustering [1], there are various practical applications of improved query clustering abilities. Even in commercial web search engines, there are difficulties in finding many examples of queries to cluster for rare queries; however this dataset contains many rare queries in each backstory cluster. Either the raw or spell-corrected query variations could be used in this application.

Query Transformations and Difficulty Prediction Performance has been shown to vary widely by the query variation [2], yet the information seeking task remains the same. With many different query variations, each with performance scores, it is now possible to examine which query variations within a cluster have the best performance, and hence investigate what query transformations on low-performing queries lead to better results. An alternative to this investigation would be to use the collection for learning to predict query difficulty [14], given each cluster has a range of scores.

Relevance Feedback Blind relevance feedback approaches have been well studied [13]. The UQV100 collection can support new analysis of such algorithms, given the large number of query variations per cluster. The analysis can examine either individual queries, or the entire set of variations, and may provide additional opportunities for exploring how structural or syntactic elements of query expression (for example, presence of natural language structures, stop-words, term count, and so on) can lead to altered performance of relevance feedback.

4. CONCLUSIONS

We have constructed a substantial new information retrieval test collection (UQV100), using the ClueWeb12-B corpus as the underlying set of documents. The additional resources for the test collection consist of: 100 backstories and their paired TREC topic/subtopic id; corresponding normalized and spell-corrected query variations (an average of 58 per backstory) and effort estimates; judging guidelines; gold hits for qualification and quality control for future judgments of unjudged documents; top-*k* document ID rankings for all queries using Indri/BM25 against ClueWeb12-B; and relevance judgments (as per judging guidelines) for as a minimum the top-10 pooled documents for each query variation relative to the corresponding backstory. These resources are all freely available from `http://dx.doi.org/10.4225/49/5726E597B8376`.

Acknowledgment This work was supported by the Australian Research Council's *Discovery Projects* Scheme (project DP140102655). Xiaolu Lu (RMIT University) provided helpful assistance.

References

[1] R. Baeza-Yates, C. Hurtado, and M. Mendoza. Improving search engines by query clustering. *JASIST*, 58(12):1793–1804, 2007.

[2] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. User variability and IR system evaluation. In *Proc. SIGIR*, pages 625–634, 2015.

[3] C. Buckley and E. M. Voorhees. Retrieval system evaluation. In E. M. Voorhees and D. K. Harman, editors, *TREC: Experiment and Evaluation in Information Retrieval*, chapter 3, pages 53–75. MIT Press, 2005.

[4] C. Buckley and J. Walz. The TREC-8 query track. In *Proc. TREC*, 1999. NIST Special Publication 500-246.

[5] K. Collins-Thompson, P. N. Bennett, F. Diaz, C. L. A. Clarke, and E. M. Voorhees. TREC 2013 web track overview. In *Proc. TREC*, 2013. NIST Special Publication 500-302.

[6] K. Collins-Thompson, C. Macdonald, P. N. Bennett, F. Diaz, and E. M. Voorhees. TREC 2014 web track overview. In *Proc. TREC*, 2014. NIST Special Publication 500-308.

[7] A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Trans. Inf. Sys.*, 27(1):2.1–2.27, 2008.

[8] A. Moffat, W. Webber, and J. Zobel. Strategic system comparisons via targeted relevance judgments. In *Proc. SIGIR*, pages 375–382, 2007.

[9] A. Moffat, P. Bailey, F. Scholer, and P. Thomas. INST: An adaptive metric for information retrieval evaluation. In *Proc. Aust. Doc. Comp. Symp.*, pages 5:1–5:4, 2015.

[10] A. Moffat, F. Scholer, P. Thomas, and P. Bailey. Pooled evaluation over query variations: Users are as diverse as systems. In *Proc. CIKM*, pages 1759–1762, 2015.

[11] The Lemur Project. The ClueWeb12 Dataset, 2012. URL `www.lemurproject.org/clueweb12.php/`.

[12] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi. Community-based Bayesian aggregation models for crowdsourcing. In *Proc. WWW*, pages 155–164, 2014.

[13] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proc. SIGIR*, pages 4–11, 1996.

[14] E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow. Learning to estimate query difficulty: Including applications to missing content detection and distributed information retrieval. In *Proc. SIGIR*, pages 512–519, 2005.