LAMP-TR-118 February 2005
CS-TR-4693
UMIACS-TR-2005-03

Evaluation of Resources for Question Answering Evaluation

Jimmy Lin

College of Information Studies
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742
jimmylin@umd.edu

Abstract

Controlled and reproducible laboratory experiments, enabled by reusable test collections, represent a well-established methodology in modern information retrieval research. In order to confidently draw conclusions about the performance of different retrieval methods using test collections, their reliability and trustworthiness must first be established. Although such studies have been performed for *ad hoc* test collections, currently available resources for evaluating question answering systems have not been similarly analyzed. This study evaluates the quality of answer patterns and lists of relevant documents currently employed in automatic question answering evaluation, and concludes that they are not suitable for post-hoc experimentation. These resources, created by pooling runs of TREC QA track participants, do not produce fair and reliable assessments of systems that did not participate in the original evaluation. Potential solutions for addressing this evaluation gap are discussed.

Keywords: question answering, evaluation, reusable test collections

1 Introduction

The use of test collections to assess the performance of information retrieval systems is a well-established methodology, dating back to the Cranfield experiments in the 60's [6]. Test collections enable the effectiveness of different retrieval methods to be compared without human involvement to assess the relevance of the returned documents. Thus, they allow information retrieval experiments to be conducted with rapid turnaround in a controlled laboratory setting.

In the past decade or so, large test collections for the so-called *ad hoc* task have been created by pooling the ranked lists returned by participants in large-scale evaluations such as TREC. With pooling, the top n documents (the *pool depth*) from each run are gathered, and after removing duplicates, are presented to human assessors for evaluation. These judgments are then used to evaluate the entire ranked list of all runs.

Previous work [16] has shown that the pooling methodology creates trustworthy and reliable resources for post-hoc experimentation, i.e., pooled judgments can accurately assess the retrieval performance of systems that did not participate in the original evaluation. Researchers have probed other aspects of the TREC test collections, including the effect of topic size [14], the effect of incomplete judgments [4], the effect of different evaluation metrics [3], and the effect of different notions of relevance [11, 12, 10]. In general, these studies have confirmed the reliability of using TREC test collections as a laboratory tool for measuring the effectiveness of different retrieval methods.

In the past few years, question answering has emerged as an active area of research. Combining traditional information retrieval and natural language processing technologies, question answering systems aim to directly return answers to questions posed in everyday language such as "How old was Nolan Ryan when he retired?", instead of returning a ranked list of documents that a user must then manually examine. Although the community is moving towards more difficult questions such as ones that require reasoning, these so-called "factoid" questions, which can be answered by a named-entity or a short noun phrase, remain a staple of question answering research.

Since 1999, the NIST-organized question answering tracks at TREC (see, for example, [13]) have served as a locus of research in the field, providing an annual forum for the evaluation of diverse systems fielded by teams from all over the world (the model has been duplicated and elaborated on by CLEF in Europe and NTCIR in Asia, which have also introduced cross-language elements in question answering). Drawing lessons from other TREC efforts, the need for a question answering test collection has been identified since the earliest days of the TREC QA tracks [15]. However, to date, a truly reusable test collection still does not exist. Each year, NIST releases evaluation resources intended to guide the development of future question answering systems. However, these resources have been used in ways they were never intended for, e.g., to directly assess the accuracy of question answering systems and to make comparative judgments about the performance of different question answering techniques. Such results are potentially misleading because the reliability and trustworthiness of measures derived from these resources in post-hoc experiments have not been established. To date, no researcher has analyzed existing question answering evaluation resources in the same way that *ad hoc* test collections have been dissected.

This study presents an evaluation of presently available resources for evaluating factoid question answering. Specifically, the following questions will be addressed: Why is creating a reusable test collection for question answering so difficult? Can existing resources be employed to measure the performance of new question answering techniques, i.e., can they be used for post-hoc experimentation? Results will demonstrate that presently available resources do not produce reliable scores and should not be used to evaluate systems that did not participate in the original TREC evaluations. In addition to a detailed analysis of present practices, this paper will discuss potential solutions for building reusable question answering test collections in the future.

2 Evaluating QA Systems

In the TREC instantiation of the question answering task, a system response to a natural language question is a pair consisting of an answer string and a supporting document (from which the answer string was extracted). To this response unit, a human assessor assigns one of four labels: *correct*, *inexact*, *unsupported*, and *wrong*. In order for a response unit to be judged as *correct*, the answer string must contain exactly the information requested by the question and the supporting document must present the answer in a manner such that a human reading the document could verify its correctness. If the answer string contains extraneous words, the entire response is judged *inexact*. If the answer string is correct (i.e., exact), but the document does not clearly answer the question, an *unsupported* label is assigned. Otherwise, the response unit is *wrong*. In a typical QA track at TREC, participating systems are given four to five hundred factoid questions and are allowed to return one response per question. Performance is measured by answer accuracy (precision), i.e., the fraction of the questions that were answered correctly by a system.

It is worthwhile to mention that this study primarily addresses the TREC definition of question answering, where an answer is considered correct only if it has support from a particular document drawn from a fixed collection. Support, however, is not intrinsic to question answering and the notion may be less applicable to other types of question answering, e.g., using the Web or FAQ lists.

Question answering test collections are difficult to construct for two reasons. Because judgments are made with respect to both an answer string and a particular document, identical answer strings may be correct or unsupported, depending on the specific document from which that string came. This issue is compounded by the fact that there is no such thing as a "canonical answer form", akin to the docid in *ad hoc* retrieval. Small changes in the answer string might affect the judgment value, and there is (currently) no algorithmic way to determine whether a change between a judged string and a string to be evaluated is significant or not. Thus, human judgments are required deal with intricacies such as exactness and acceptability of variants.

Despite these challenges, NIST has been compelled to provide resources to guide future system development. Each year, answer patterns in the form of regular expressions and a list of relevant documents containing those answers (the reldocs list) are compiled by pooling runs submitted by participants. In addition, NIST provides a scoring script that matches these answer patterns against system responses.

Although TREC organizers never intended for the answer patterns and reldocs lists to be used as a test collection, they nevertheless have—results based thereon, for example, comparing the relative effectiveness of different techniques, have been widely reported in the literature. Typically, two measures of answer accuracy, *strict* and *lenient*, are reported. In the strict measure, a response is considered correct only if the answer string matches the answer patterns and its supporting document is among those marked as relevant (from the reldocs list). In the lenient measure, the supporting document is ignored, and response units are scored only on the answer strings.

It is well-known that the strict measure underestimates answer accuracy because document-level relevance judgments are incomplete: an acceptable answer may be judged incorrect simply because its supporting document does not appear in the reldocs list. On the other hand, the lenient measure overestimates performance because documents often coincidentally contain the correct answer string without actually answering the question. However, it is implicitly assumed that the combination of the two different evaluation criteria would closely approximate true question answering effectiveness. However, no one to date has rigorously verified this assumption, and indeed, this study will reveal that TREC answer patterns and the reldocs list are not suitable for post-hoc experimentation. Before delving into details, however, relation of this work to previous meta-evaluations of *ad hoc* retrieval will be surveyed.

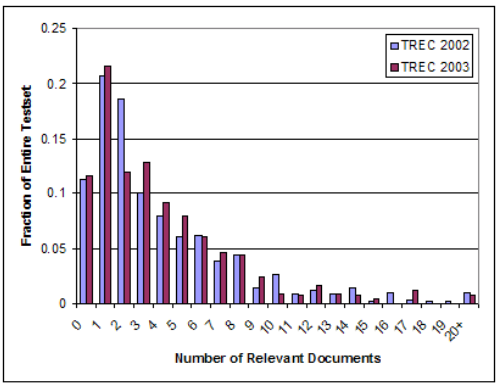

Figure 1: Histogram of questions (in terms of fraction of the entire testset) binned by number of relevant documents for TREC 2002 and TREC 2003.

3 Reasons for Caution

Existing resources for automatic question answering evaluation were essentially created by employing the pooling methodology. Fortunately, aspects of this process as applied to *ad hoc* retrieval have been well studied. Although Zobel [16] demonstrated that a pool depth of one hundred documents produces a fair ranking of systems, he started noticing adverse affects on system rankings when the pool size was reduced below fifty documents. In the current setup of the question answering task, the pool depth is one, because each system is only allowed to return one response per question. This result alone should raise suspicions regarding the reliability of these resources, considering that Buckley and Voorhees [4] has shown test collections not to be robust with respect to massively incomplete relevance judgments. For the TREC 2002 testset, the list of known relevant documents is quite small, averaging 3.95 relevant documents per question (σ=4.07, max=23); reldocs for the TREC 2003 testset average 3.90 document per question (σ=3.84, max=25). A histogram of questions from both testsets binned by their count of relevant documents is shown in Figure 1.[1] A casual examination of the corpus reveals many relevant documents that are not in the official reldocs list; the danger here is that systems may not be properly rewarded for extracting answers from these documents (if support is considered a necessary condition for successfully answering a question).

Two other reasons for the soundness of the pooling strategy in *ad hoc* retrieval are that participants contributing to the pool employ a relatively diverse set of strategies and that they achieve reasonable performance. Unfortunately, the same cannot be said of question answering systems. In the most recent QA track at TREC 2004, Ellen Voorhees reported in her overview talk that the median score for 92.2% of factoid questions was zero. On the diversity front, question answering systems fare no better: because many teams focused only on the answer extraction aspect of the task, they simply used the ranked list of documents supplied by the PRISE system, made available by NIST. By Monz's [9] count, 10 out of 36 participants (28%) used only these documents in TREC 2001. For TREC 2002, the number was 7 out of 34 participants, or 21%. Naturally, this reduces the diversity of documents that contribute to the pool.

To quantify the effect of missing document relevance judgments, a "perfect" run was created by a human for the first one hundred questions from the TREC 2002 testset. Since a human manually extracted answers for each question from the document collection, the run should achieve an accuracy of 1.0 when automatically evaluated with existing resources.[2] This, unfortunately, is not the case: the

[1] Since the TREC 2002 testset has 500 factoid questions, while the TREC 2003 testset has only 413 factoid questions, the number of questions has been normalized as a fraction.

[2] modulo variation caused by differences in opinion

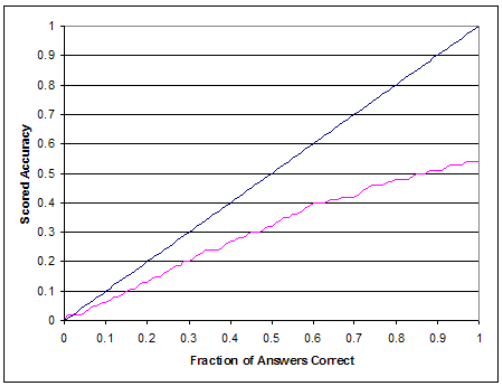

Figure 2: Artificially created runs with a known fraction of correct answers, plotted against automatically-derived strict answer accuracy (reference line has a slope of one).

human-generated run achieved a strict accuracy of only 0.53 using current evaluation resources.

Given this human-generated run, one can vary the percentage of correct answers and observe changes in the automatically-derived strict score. The one hundred questions in this testset were ordered from "easiest" to "hardest", based on the number of participating systems that answered the questions correctly. From the human-generated run, 101 different "fake" runs were produced, each with a known number of correct answers—sampled in the order of increasing difficulty, such that easier questions were always chosen first. Each of these runs were automatically evaluated; the resulting strict score plotted against the fraction of known correct answers is shown in Figure 2. With an "ideal" set of judgments, this graph should be a straight line with a slope of one through the origin, as shown by the reference line. However, this is clearly not the case—we can see that existing evaluation resources underestimate answer accuracy.

In the following sections, quantitative evidence cautioning against using existing evaluation resources for post-hoc experimentation will be presented. Note that within the context of a single evaluation, the reliability of human-generated scores have been verified to within a known margin of error (see, for example, [13]). That is, comparisons between different submitted runs are trustworthy and valid.

4 Strict and Lenient Measures

To assess the reliability of available answer patterns and reldocs lists, they were employed to assess runs that were actually submitted to TREC. For the 500 factoid questions in the TREC 2002 testset, the Kendall's τ correlation between the official system rankings and the system rankings produced using the automatically-generated strict measure (matching both answer patterns and reldocs) is 0.817; for the lenient measure (matching answer patterns only), the correlation is 0.820. Kendall's τ computes the "distance" between two rankings as the minimum number of pairwise adjacent swaps necessary to convert one ranking into the other. The correlation is above the 0.8 threshold generally considered "good", but less than the 0.9 threshold NIST aims for. For the TREC 2003 testset (considering only the 413 factoid questions), the Kendall's τ correlation between the official score and the strict measure is 0.883, and 0.807 between the official score and the lenient measure. Figure 3 shows the official score of each run (answer accuracy), along with the strict and lenient automatically-generated scores for both TREC 2002 and TREC 2003, arranged by descending official score.[3]

[3]The official score is sometimes higher than both the strict and lenient automatically-derived scores because the NIST-supplied scoring script does not correctly handle questions in which there are no known answers in the corpus; this, however, does not significantly affect these results and the conclusions drawn therefrom.

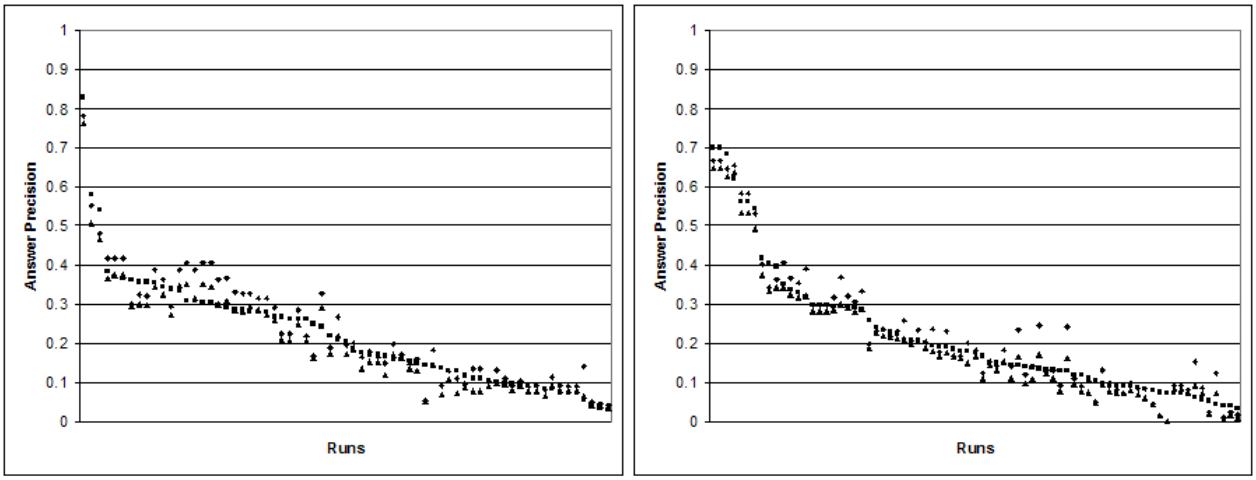

Figure 3: Official scores (squares) plotted with lenient (diamonds) and strict (triangles) automatically-generated scores for all TREC 2002 runs (left) and all TREC 2003 runs (right).

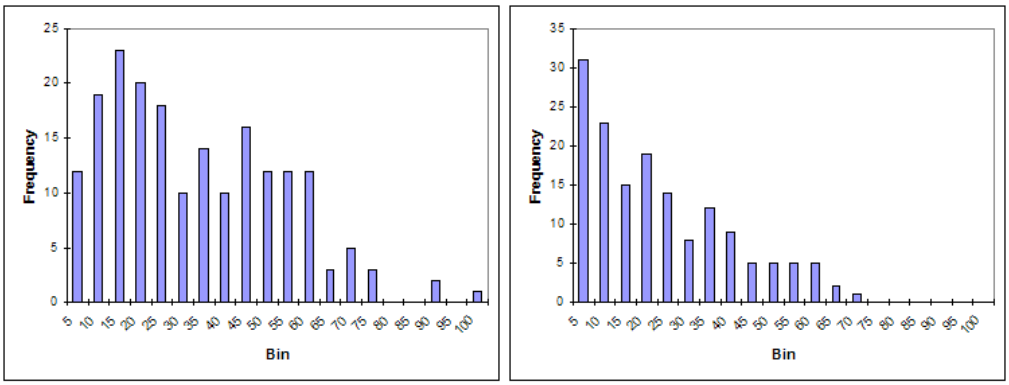

Figure 4: Rank swaps between official and automatically-generated strict score binned by difference in official scores for TREC 2002 (left) and all TREC 2003 runs (right). Bin units are thousandths, e.g., the binned labeled 25 has $0.02 \leq \delta < 0.025$.

What exactly do the above correlations mean? The Kendall's τ gives us an idea about the prevalence of rank swaps, instances where different evaluation conditions give different conclusions about which run out of a pair is better. However, there is measurement error associated with any evaluation, so rank swaps between runs whose official scores differ by less than this error margin are inconsequential. For TREC 2002, Voorhees [13] determined the score difference to be 0.05, i.e., in order to confidently conclude that one run is better than another, there must be an absolute score difference δ of at least 0.05.[4] Figure 4 shows histogram of all observed rank swaps binned by their difference in official score. For TREC 2002, 192 rank swaps were observed, out of 2211 pairwise comparisons (67 runs); of those, 38 rank swaps occurred when δ was greater than 0.05. For TREC 2003, 154 rank swaps were observed, out of 2275 pairwise comparisons (75 runs); of those, 13 rank swaps occurred when δ was greater than 0.05. It appears that resources from TREC 2003 replicate the official judgments better; nevertheless, it is noted that existing resources are not completely reliable, even when applied to submitted runs.

These results highlight the difficulty in crafting a good set of answer patterns and the problems associated with treating answers and supporting documents independently (recall that in the manual

[4]The same analysis has not be performed for TREC 2003, but it is assumed that the figure is comparable.

assessment process, they are always considered as a single unit). Answer patterns are often too restrictive, i.e., they do not accept correct answers, and at the same time, not sufficiently restrictive, i.e., they accept answers that are not supported. An often-observed mode of failure is the inability of answer patterns to weed out answer strings contained in documents that do not directly answer the question. Consider the following example:

> **Question:** Who was the first black heavyweight champion?
> **Answer:** Jack Johnson
> **Supporting document**: ... Louis was the first African-American heavyweight since Jack Johnson who was allowed to get close to that symbol of ultimate manhood, the heavyweight crown...

Although Jack Johnson was the first black heavyweight champion, the document does not support the answer, i.e., it does not state that he was the *first* one. Thus, the above response would be marked *unsupported*. In many cases, the lenient measure greatly overestimates answer accuracy when considering support, particularly for a class of systems that extracts answers from the World Wide Web, and then "projects" these answers onto the corpus used by the TREC evaluations [2, 8]. Lin et al. [8], for example, reported that their system could not find relevant supporting documents for approximately a quarter of otherwise correct answer strings. Thus, lenient scores could be inflated by as much as a third.

5 Post-hoc Experimentation

The previous experiment employed available resources to assess the performance of systems that participated in the TREC QA tracks. It attempted to answer the question: Can the answer patterns and reldocs lists duplicate human judgments on submitted TREC runs? The Kendall's τ values and analysis of rank swaps appear to suggest *yes*, although caution is warranted because a number of significant rank drops were observed. What about use of these resources for post-hoc evaluation, i.e., to score systems that were not TREC participants? This section presents two experiments that explores this question, whose answers appears to be *no*.

5.1 The Take-One-Out Experiment

The "take-one-out" experiment was designed to simulate the effect of using the answer patterns and the reldocs to score a run that was not evaluated by TREC assessors; the methodology is similar to the study conducted by Zobel [16]. For each run in a particular year's evaluation, a new reduced set of relevant documents is created by removing the contributions of that run.[5] This resulted in 67 variant reldocs lists for TREC 2002 and 75 variant reldocs lists for TREC 2003, one with contributions from each run removed. Each of these judgment sets were then used to evaluate the run whose contributions were removed, simulating the effect of post-hoc evaluation. Ideally, the score of a run evaluated using the reduced judgments should be very close to the score generated using the complete set of judgments. This would indicate that the score of a run is invariant with respect to its participation in TREC, and hence these resources can be used for post-hoc evaluation. Zobel discovered exactly this result for *ad hoc* test collections.

The results of this experiment, however, present a different story. Scores generated with the reduced judgments were invariably lower than scores generated with the complete set of judgments, indicating that existing resources underestimate the performance of systems that did not participate in the original

[5] In actuality, multiple runs from the same organization were removed together because they tended to return very similar documents.

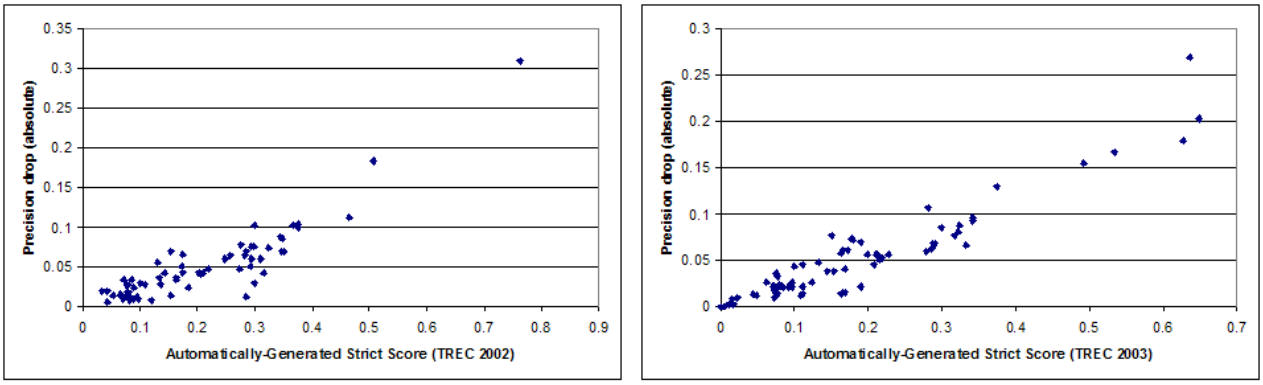

Figure 5: Scatter plot of automatically-generated strict score of each run with the reduced judgments against the precision drop for TREC 2002 (left) and TREC 2003 (right).

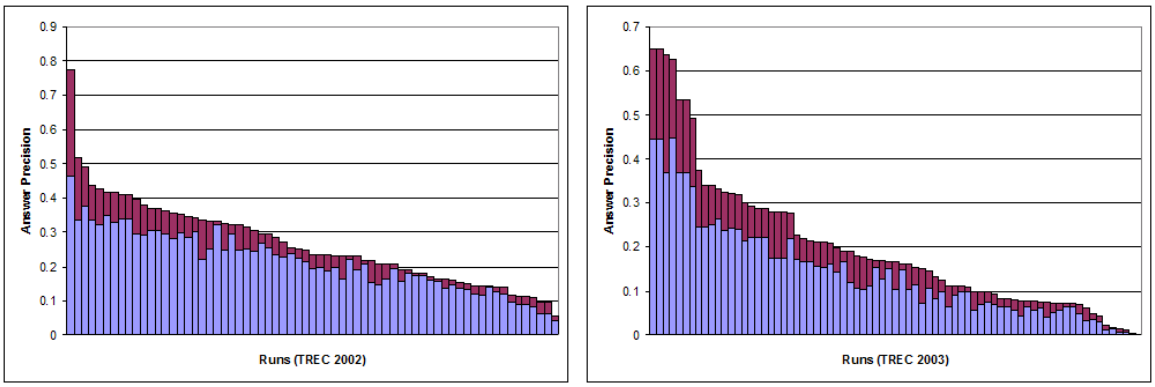

Figure 6: Comparison of automatically-generated strict score with original relevance judgments (longer bar) and strict score with reduced judgments (shorter bar) for TREC 2002 (left) and TREC 2003 (right).

TREC evaluations. Figure 5 shows scatter plots of the difference in precision between the two evaluation conditions against the original score. The higher the original score, the more the answer accuracy is under-estimated by available resources, which is consistent with Figure 2.

In Figure 6, the strict scores with the original and reduced judgments are shown as a bar graph, in order of descending original strict scores (TREC 2002 on left, TREC 2003 on right). As an example, the top scoring run from TREC 2003 achieved a precision of 0.649; removing its contributions from the pool drops the score to 0.446, nearly thirty percent lower (an absolute difference of twenty percentage points)! Without its contributions to the pool, the top scoring run would now be ranked seventh. Such flips in relative rankings suggest that a run evaluated with existing resources should not be compared with a run that was submitted to TREC.

5.2 The Take-Two-Out Experiment

The previous experiment demonstrated that presently available resources for question answering underestimate answer accuracy when used in post-hoc experimentation. Specifically, a new run cannot be reliably compared to an existing run. What about two fresh runs? If the resources underestimate performance in a consistent manner, might they be reliable when used to compare *two* runs that were not originally part of TREC?

The "take-two-out" experiment was devised to answer this question. For every pairwise comparison between runs from TREC 2002 and TREC 2003, a reduced judgment set with contributions removed

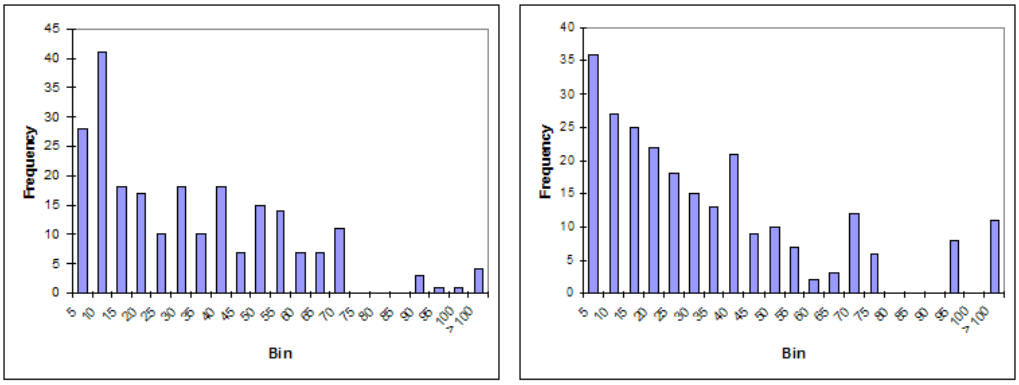

Figure 7: Histogram of rank swaps for the "take-two-out" experiment for TREC 2002 (left) and TREC 2003 (right). Bin units are thousandths.

from *both* runs was created. This simulates the effect of using the resources to score two new runs that did not participate in the original evaluation. For TREC 2002, 230 rank swaps were observed (out of 2211 pairwise comparisons with 67 runs); in 48 of those, the difference in official score was greater than 0.05. For TREC 2003, 245 rank swaps were observed (out of 2275 pairwise comparisons with 75 runs); in 49 of those, the difference in official score was greater than 0.05. The histogram of these rank swaps binned by the difference in official score is shown in Figure 7. The maximum observed score difference that produced a rank swap was 0.254 for TREC 2002 and 0.262 for TREC 2003.

These results indicate that existing evaluation resources are not suitable for use in post hoc experimentation when one wishes to evaluate not only the correctness of the answer string, but also the supporting document from which it was extracted.

6 Potential Solutions

Given the important role that reproducible, accurate, and reliable laboratory experiments play in information retrieval, how can we address the current lack of a truly reusable test collection for question answering? One solution that has been suggested is to simply increase the pool depth to a point where a good set of judgments can be gathered (say, one hundred documents). Instead of returning a response unit consisting of an answer string and supporting document, systems could return a traditional ranked list of documents (which presumably contain the answer somewhere within). Currently, there are plans to implement this solution for the TREC 2005 QA track.

This pooling approach, however, has the downside that it essentially adds a completely separate *ad hoc* retrieval component to the already difficult-to-evaluate question answering task. Although we can be fairly confident that this strategy will produce a set of judgments for post-hoc experimentation, one wonders if there is a more cost-effective method for achieving the same effect.

Given that answer strings must be "exact" in the current setup of the TREC question answering track, it becomes possible to "work backwards" from an answer to fetch all relevant supporting documents. This approach has been previously employed to create a small reusable test collection for question answering [1]. Working from known answers, it is possible to craft very specific boolean queries using selected terms from both the question and the answer. Naturally, not all retrieved documents will answer the question—they must still be manually examined. However, it is hoped that this list of documents will be much smaller than the one gathered by pooling. Given the poor performance of most current systems, the vast majority of documents in any pool will be irrelevant.

To illustrate this process, consider the question "What is the name of the volcano that destroyed the ancient city of Pompeii?", whose answer is "Vesuvius" (or some variant thereof such as "Mt.

Vesuvius"). One might assume that all relevant documents must, at the very least, contain the terms "Pompeii" and "Vesuvius". Thus, documents satisfying the boolean query "Pompeii AND Vesuvius" should encompass the set of relevant documents. As it turns out, there are only twenty-eight such documents in the entire AQUAINT corpus (used in the TREC QA tracks) that satisfy this constraint. These documents still must be manually examined to see if they actually contain the correct answer and provide the necessary support. An example of a clearly relevant document is APW19990823.0165 , which states that "In A.D. 79, long-dormant Mount Vesuvius erupted, burying the Roman cities of Pompeii and Herculaneum in volcanic ash." An example of an irrelevant document containing both "Pompeii" and "Vesuvius" is NYT20000704.0049; the article discusses winemaking in Campania, the region of Italy where both Pompeii and Vesuvius are located—it describes vineyards near the ruins of Pompeii and grape varieties that grow in the volcanic soil at the foot of Mt. Vesuvius.

In this particular example, the strategy described above would involve less manual labor than the standard pooling setup with a depth of one hundred documents. In general, one could argue that this technique of searching for the known answer is more efficient at gathering judgments because users do not have to examine as many documents on average. However, there a few drawbacks to this approach, described below.

First, there are many types of questions where a good boolean query cannot be constructed to gather an initial set of candidate documents for manual examination. Consider a question such as "What country is Berlin located in?", which potentially has many answer instances in the document collection. A reasonable boolean query such as "Berlin AND Germany" would return too many hits to be practically examined manually. It is unclear how one might further restrict this query or at what point one should simply stop judging documents. Questions involving numeric answers represent another difficult category of information needs: take, for example, "What is the population of Nigeria?" Population figures are highly variable because journalists report these numbers to different levels of granularity, e.g., some round to the nearest million, others give official census figures, and still others provide projected numbers. Furthermore, since population figures change over time, it is difficult to formulate an appropriate boolean query to retrieve a sufficiently narrow set of documents for manual assessment. These types of questions are relatively prevalent, and potentially problematic even when the numeric answers remain (relatively) constant. For example, there are at least two different heights of Mt. Everest reported in the AQUAINT corpus, corresponding to different geographic surveys. Since current guidelines do not take into account issues such as factual error, all of these answers would be accepted as "correct" for the question "How tall is Mount Everest?", even though some heights are more accurate than others. Since it is not possible to *a priori* predict *all* possible answer variations in these cases, working backwards from known answers could potentially result in incomplete judgments.

Another limitation of this proposed approach stems from the assumption that relevant documents must contain certain terms from either the question or the answer, or both. However, this may simply not be the case due to such problems as vocabulary mismatch. For example, George Washington may never be mentioned by name, but might rather be simply referred to as "the father of our nation". In principle, these variations are impossible to predict *a priori* without examining the collection. Consider the question "When did Lenin die?", whose answer is January 21, 1924. Working backwards from the answer, the query "Lenin AND 1924" might be a reasonable start to generating the initial set of documents to assess. However, this method would not retrieve the following document:

> **Question:** When did Lenin die?
> **Answer:** January 21, 1924
> **Supporting document**: ...Friday is the 70th anniversary of Lenin's death...(article dated Thursday, January 20, 1994)

Some relevant documents simply don't contain obvious terms. Unfortunately, it is unknown how prevalent this phenomenon is, or what effects such documents might have on the evaluation of question

answering systems. However, documents in which answers are not obviously stated represent opportunities for sophisticated linguistic techniques that, for example, involve temporal reasoning. Since the question answering task was in part designed to exercise advanced natural language processing techniques, relevance judgments should encompass these more difficult-to-extract answers.

Search-guided relevance assessment [5] is an alternative strategy to gathering relevance judgments that addresses many of the abovementioned limitations. Instead of a single-shot process, assessors interactively search the collection (typically with a ranked retrieval system), iterating between topic research and relevance assessment. This technique overcomes many vocabulary mismatch problems because users engage in iterative query refinement, drawing terms from the very documents they are assessing. Since answer terms can still be part of the query, this procedure has the potential advantage of producing documents that are more likely to be relevant. At the same time, since assessors are working with a ranked retrieval systems, documents that do not have all query terms present may be retrieved, allowing the possibility that documents with difficult-to-extract answers are judged. Typically, search-guided relevance assessment either stops after a fixed interval or after a period where fewer and fewer relevant documents are found. Cormack et al. [7] compared a less-structured variant of search-guided relevance assessment called Interactive Searching and Judging with traditional pooling and concluded that the technique is an attractive alternative. In addition, search-guided relevance assessment can be further enhanced with pooling of unjudged documents to serve as a quality control [5].

7 Conclusion

The notion of quantitative evaluations is central to information retrieval research. Although this focus is considered the mark of a mature discipline and has contributed to steady advance in the state of the art, we must confirm that our evaluations are meaningful and that their results are reliable. The work presented in this paper suggests that one must be cautious when employing existing resources to evaluate question answering systems—they should not be used to make quantitative comparisons regarding the effectiveness of different techniques. The errors associated with scores derived from answer patterns and reldocs lists preclude reliable conclusions from being drawn.

This meta-evaluation of question answering points out shortcomings in existing evaluation methodology as it relates to building reusable test collections for post-hoc experimentation, but potential solutions are also discussed. Somewhat ironically, a good evaluation infrastructure may spell the end of the evaluation that created it; in the same way that the TREC *ad hoc* collections have obviated the need to run the *ad hoc* tracks annually at TREC, the construction of a large-scale, reusable factoid question answering test collection may obviate the need to evaluate factoid questions on an annual basis. The resulting resources could then be employed to explore new directions in question answering.

8 Acknowledgments

I'd like to thank Doug Oard for engaging discussions and helpful suggestions, Ellen Voorhees for insightful comments, and Kiri for her kind support.

References

[1] Matthew W. Bilotti. Query expansion techniques for question answering. Master's thesis, Massachusetts Institute of Technology, 2004.

[2] Eric Brill, Jimmy Lin, Michele Banko, Susan Dumais, and Andrew Ng. Data-intensive question answering. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, 2001.

[3] Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*, 2000.

[4] 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)*, 2004.

[5] Christopher Cieri, Stephanie Strassel, David Graff, Nii Martey, Kara Rennert, and Mark Liberman. Corpora for topic detection and tracking. In James Allan, editor, *Topic Detection and Tracking: Event-Based Information Organization*, pages 33–66. Kluwer Academic Publishers, Norwell, Massachusetts, 2002.

[6] Cyril W. Cleverdon, Jack Mills, and E. Michael Keen. Factors determining the performance of indexing systems. Two volumes, ASLIB Cranfield Research Project, Cranfield, England, 1968.

[7] 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 1998)*, 1998.

[8] Jimmy Lin, Aaron Fernandes, Boris Katz, Gregory Marton, and Stefanie Tellex. Extracting answers from the Web using knowledge annotation and knowledge mining techniques. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*, 2002.

[9] Christof Monz. *From Document Retrieval to Question Answering*. PhD thesis, Institute for Logic, Language, and Computation, University of Amsterdam, 2003.

[10] Eero Sormunen. Liberal relevance criteria of TREC—counting on negligible documents? In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, 2002.

[11] 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 1998)*, 1998.

[12] Ellen M. Voorhees. Overview of the TREC-9 question answering track. In *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*, 2000.

[13] Ellen M. Voorhees. Evaluating the evaluation: A case study using the TREC 2002 question answering track. In *Proceedings of the 2003 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL 2003)*, 2003.

[14] 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)*, 2002.

[15] Ellen M. Voorhees and Dawn M. Tice. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*, 2000.

[16] Justin Zobel. 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 (SIGIR 1998)*, 1998.