

Deconstructing Nuggets: The Stability and Reliability of Complex Question Answering Evaluation

Jimmy Lin and Pengyi Zhang
College of Information Studies
University of Maryland
College Park, MD 20742, USA
{jimmylin,pengyi}@umd.edu

ABSTRACT

A methodology based on “information nuggets” has recently emerged as the *de facto* standard by which answers to complex questions are evaluated. After several implementations in the TREC question answering tracks, the community has gained a better understanding of its many characteristics. This paper focuses on one particular aspect of the evaluation: the human assignment of nuggets to answer strings, which serves as the basis of the F-score computation. As a byproduct of the TREC 2006 ciQA task, identical answer strings were independently evaluated twice, which allowed us to assess the consistency of human judgments. Based on these results, we explored simulations of assessor behavior that provide a method to quantify scoring variations. Understanding these variations in turn lets researchers be more confident in their comparisons of systems.

Categories and Subject Descriptors

H.3.4 [Information Storage and Retrieval]: Systems and Software—*Performance evaluation*

General Terms

Measurement, Experimentation

Keywords

TREC, complex information needs, human judgments

1. INTRODUCTION

Quantitative evaluation has always played an important role in information retrieval research, a tradition that dates back to the Cranfield experiments in the 60’s [4]. The reliance on controlled, reproducible experiments to guide progress in the field naturally places evaluation methodologies at the center of much attention. For the *ad hoc* retrieval task, significant work has focused on validating and refining

the laboratory tools that researchers use (e.g., [13, 2, 10, 3], just to name a few). These studies provide a degree of confidence in the meaningfulness of results from IR experiments.

In the past decade or so, question answering has emerged as an active area of research. By combining term-based information retrieval techniques with deeper linguistic processing technologies, QA systems return *answers* instead of *hits*. A departure from the ranked list has necessitated the development of new evaluation methodologies to assess the quality of system output. So-called “factoid” questions such as “When was Sputnik launched?” presented interesting evaluation challenges and have been explored in many studies [15, 6, 9]. In contrast, the nugget-based methodology for evaluating answers to complex questions—the focus of this paper—has received comparatively little attention.

After having made substantial headway in factoid QA, researchers have turned their attention to more complex information needs that cannot be answered by simply extracting named entities (persons, organization, locations, dates, etc.) from documents. These questions might, for example, involve inferencing or synthesizing information from multiple documents. The definition and “other” questions in the NIST-sponsored TREC QA evaluations exemplify this shift to more complex information needs [12]. The complex, interactive question answering task (ciQA)—introduced by NIST in TREC 2006 and the focus of our paper—is another instance of this trend. For these tasks, NIST has developed an evaluation methodology based on “information nuggets”, in which humans are called upon to establish the presence of important facts in system responses.

We tackle the following question: are human-based nugget assignments stable? In other words, can assessors reliably determine the nugget content of answer strings? The answer holds important implications for the reliability of system scores and the degree of confidence researchers can have in system comparisons. As a byproduct of the ciQA task in TREC 2006, submissions with identical answer strings were independently judged twice by NIST assessors (without their explicit knowledge). This fortunate circumstance provided a unique opportunity to answer this research question. Analysis shows that, on the whole, NIST assessors are sufficiently consistent such that human variability does not have a large effect on system scores. Through simulation studies, we are able to quantify these scoring variations. We believe that insights gained through analysis of ciQA results can be broadly applied to other complex QA evaluations that employ the same methodology.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGIR’07, July 23–27, 2007, Amsterdam, The Netherlands.

Copyright 2007 ACM 978-1-59593-597-7/07/0007 ...\$5.00.

This paper is organized as follows: The nugget-based evaluation methodology and the ciQA task are described in Sections 2 and 3, respectively. A rough analysis of assessor inconsistencies is presented in Section 4 and refined in Section 5. We describe a simulation-based method for quantifying the effects of inconsistent judgments in Section 6 and discuss our model in Section 7. Future extensions are outlined in Section 8 before the conclusion.

2. NUGGET-BASED EVALUATIONS

To date, NIST has conducted several formal evaluations of complex question answering: definition questions at TREC 2003, “other” questions at TREC 2004–2006, and ciQA at TREC 2006 (see next section). All of them have employed the nugget-based evaluation methodology, which has evolved into the *de facto* standard for complex QA evaluation over the past few years. Given the central role of evaluation in guiding research, aspects of the process warrant closer examination. This section provides a brief overview of the evaluation process; the reader is advised to consult related work for more details [11, 12, 7, 8].

Answers to complex questions are comprised of an unordered set of [document-id, answer string] pairs (we use *answer string* and *response* interchangeably throughout this paper). Although no explicit limit is placed on the length of each answer string and how many answer strings there are, verbosity is penalized. To evaluate system output, NIST pools answer strings from all systems, removes their association with the systems that produced them, and presents them to a human assessor. Using these responses and research performed during the original development of the question, the assessor creates an “answer key”—a list of “information nuggets” about the target. An information nugget is defined as a fact for which the assessor can make a binary decision as to whether a response contains that nugget [11]. The assessor also manually classifies each nugget as either *vital* or *okay*. Vital nuggets represent concepts that must be present in a “good” answer, in the opinion of the assessor; on the other hand, okay nuggets contribute worthwhile information but are not essential (they are neither explicitly penalized nor rewarded). Under the pyramid extension [8], multiple assessors are called upon to supply vital/okay judgments (after the creation of the answer key), which are then aggregated to produce a nugget importance weight.

After the nugget answer key is created, the assessor then manually scores each run. A single assessor is responsible for each question (over all submitted runs), so she will always use her own answer key. The nugget descriptions themselves vary in quality, ranging from sentence extracts to shorthand notes primarily meant as memory aids. For each answer string, the assessor decides whether or not each nugget is present. Assessors do not simply perform string matches in this decision process—rather, the matching occurs at the conceptual level, abstracting away from issues such as vocabulary differences, syntactic divergences, paraphrases, etc. In this paper, we study the nugget assignment process using detailed records provided by NIST. These files indicate the location of nuggets in system output, as determined by the assessor. Note that the textual descriptions of the nuggets are themselves byproducts of the evaluation and have no “official status”. However, researchers have leveraged these text fragments for automatic evaluation of answers to complex questions [7].

Template: What evidence is there for transport of [military equipment and weaponry] from [South Africa] to [Pakistan]?

Narrative: The analyst is interested in South African arms support to Pakistan and the effect such support or sales has on relations of both countries with India. Additionally, the analyst would like to know what nuclear arms involvement, if any, exists between South Africa and Pakistan.

Figure 1: A sample ciQA topic.

Once nugget assignments have been made, the final F-score is straightforwardly derived. Nugget recall is computed with respect to vital nuggets only, or alternatively using nugget weights in the pyramid method [8]. Nugget precision is approximated by a length allowance. Both components are combined into an F-score with $\beta = 3$, which gives recall three times more weight than precision. The final score of a run is an average over the F-scores of individual questions. As the exact formula is not material to this work, the reader is invited to consult [11, 8] for details.

The most salient aspect of the nugget-based evaluation methodology from the viewpoint of stability and reliability is the complex mapping between useful fundamental units of information (i.e., nuggets) as determined by assessors and the presence of those units in system output. In *ad hoc* retrieval, this correspondence is straightforward, as docids from assessors’ relevance judgments can be directly matched to hits in the ranked lists returned by systems. The same cannot be said for information nuggets, which represent concepts that may manifest in a variety of natural language expressions. As previously discussed, it is the role of NIST assessors to establish semantic-level correspondences between system output and answer nuggets. Whether this can be done consistently remains an open question—one that we explore in this paper. The issue of assignment consistency has been raised in the past: for example, by Hildebrandt et al. [5]. They illustrated the general issue with anecdotal evidence, but fell short of a detailed analysis. To our knowledge, we are the first to systematically study this question.

3. COMPLEX, INTERACTIVE QA

The complex, interactive question answering (ciQA) task was started in TREC 2006 as a secondary QA task to push the community simultaneously in two directions: towards more complex information needs (beyond factoids) and towards richer models of interaction (beyond today’s batch-style evaluation framework).

The ciQA task extended and refined the so-called “relationship” questions piloted in TREC 2005 [14]. A relationship is defined as the ability of one entity to influence another, including both the means to influence and the motivation for doing so. Eight “spheres of influence” were noted in a previous study funded by the AQUAINT program: financial, movement of goods, family ties, communication pathways, organizational ties, co-location, common interests, and temporal. Evidence for both the existence or absence of ties is relevant. The particular relationships of interest naturally depend on the context.

A relationship question in the ciQA task—a topic, in standard TREC parlance—is composed of two parts (see

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | average |
|--------------------------------|------|------|------|------|------|------|------|------|------|------|------|---------|
| type | A | A | A | A | A | M | M | A | A | A | A | |
| # responses per topic: initial | 40 | 40 | 37 | 37 | 41 | 42 | 4 | 28 | 28 | 28 | 28 | 32 |
| # responses per topic: final | 37 | 37 | 37 | 38 | 42 | 36 | 4 | 24 | 24 | 20 | 28 | 30 |
| avg length per topic: initial | 6041 | 6062 | 5822 | 5822 | 6911 | 6828 | 1299 | 3864 | 3850 | 4969 | 4775 | 5042 |
| avg length per topic: final | 5564 | 5575 | 5800 | 5795 | 6996 | 6103 | 1299 | 3300 | 3295 | 3405 | 4932 | 4733 |
| # total responses | 1105 | 1105 | 1102 | 1127 | 1254 | 1080 | 133 | 730 | 721 | 587 | 827 | 888 |
| # identical responses | 814 | 623 | 631 | 682 | 482 | 1001 | 133 | 457 | 643 | 374 | 647 | 590 |
| % overall | 74% | 56% | 57% | 61% | 38% | 93% | 100% | 63% | 89% | 64% | 78% | 70% |
| avg length per response | 148 | 150 | 153 | 151 | 172 | 162 | 288 | 133 | 133 | 174 | 173 | 167 |

Table 1: Basic descriptive statistics for the submitted runs. (for type, A=automatic, M=manual)

complete example in Figure 1). The question template is a stylized information need that has a fixed structure and free slots whose instantiation varies across different topics (represented by items in square brackets). The narrative is free-form natural language text that elaborates on the information need, providing, for example, user context, a more articulated statement of interest, focus on particular topical aspects, etc. For TREC 2006, five templates were developed and the test set included six instantiations of each, for a total of 30 topics.

In order for large-scale evaluation of interactive question answering to be practical, user-system interactions in ciQA were encapsulate in HTML pages called interaction forms—similar to clarification forms in the TREC 2005 HARD track, which focused on single-iteration clarification dialogues [1]. Instead of arbitrarily complex interface controls, interactions were limited to elements that could appear on an HTML form—checkboxes, radio buttons, text input boxes, and the like. Each topic is associated with its own individual interaction form, which can have anything on it so long it satisfies technical restrictions imposed by NIST.

The ciQA evaluation followed a multi-step procedure. In the first round, participants submitted initial answers along with the interaction forms. NIST assessors then interacted with these pages and the results of user input (i.e., CGI bindings) were sent back to the participants. In the second round, participants submitted final answers based on user feedback. NIST then evaluated both the initial and final submissions together. By comparing the two, it was possible to quantify the effects of the interactions.

An interesting byproduct of the ciQA task setup was that, in many cases, the same answer string was independently evaluated twice, since quite often the same response appeared in both the initial and final submissions. Since all runs were evaluated together, the NIST assessors were not explicitly aware of any repetition. Thus, we had a naturally-occurring experiment that provided a unique look into assessor behavior and the nugget assignment process.

4. INITIAL ANALYSIS

The ciQA task in TREC 2006 drew participation from six groups. NIST received ten initial and eleven final runs, which yielded 11 pairs of corresponding initial-final submissions. In the spirit of TREC, we have anonymized runtags and simply refer to them by number. However, we do note that 6 and 7 are manual runs, a difference that will have significant implications later on. To be clear, when we mention a run by number, we are actually referring to the pair of cor-

responding initial and final submissions. In instances where the distinction is important, the initial or final submission will be explicitly referenced.

As previously described, submissions to NIST consisted of a set of responses (answer strings). For ciQA, guidelines explicitly stated that they should be ordered in terms of the likelihood that the answer string contains an information nugget—in essence, a ranked list.¹ Since most participants employed sentence-level extraction techniques (or variants thereof), we can roughly equate an answer string with a sentence extracted from the collection. However, since manual runs were allowed, answer strings could contain arbitrary text not found in any document (e.g., higher-level abstractive summaries written by humans).

All run pairs were processed to retain answer strings that were exactly the same in the initial and final submissions. We employed a strict string comparison, thus yielding a set of responses that were identical in both submissions.² Due to the ciQA setup, these answer strings were independently assessed twice. Therefore, by comparing human judgments on both occasions, we can gain insights into the nugget assignment process. We did not examine identical answer strings in runs submitted by different participants.

Descriptive statistics for the submitted runs can be found in Table 1. Lengths are measured in non-whitespace characters, the standard for TREC QA evaluations. In total, the test data included thirty topics, six each for five distinct templates. Overall, the final submissions decreased slightly in length, both in terms of number of responses and total length. Run 7, a manual run, was unlike any of the others—both the initial and final submissions (which appear to be identical) contained far fewer responses and were much shorter overall. Otherwise, the remaining runs were not particularly notable based on these statistics alone. Table 1 also shows the number of responses in each run and the amount of overlap between initial and final submissions. These values are important because they tell us how representative the overlapping responses are of the complete system output. Note in particular that initial and final submissions differ quite a lot in runs 2, 3, and 5.

Our initial analysis focused on inconsistencies in nugget assignments between the same response in the initial and final submissions—we call these “nugget flips”. Note that the notion of precedence is irrelevant for the purposes of

¹This is not a requirement for complex QA evaluation in general.

²Note that identical answer strings did not necessarily appear in the same position (both in absolute terms and relative to other answer strings).

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | average |
|------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|--------------|--------------|--------------|--------------|--------------|
| case 1 Y→Y | 65 8.0% | 68 10.9% | 74 11.7% | 78 11.4% | 61 12.7% | 132 13.2% | 65 48.9% | 41 9.0% | 54 8.4% | 82 21.9% | 114 17.6% | 76 12.9% |
| case 2 N→N | 731 89.8% | 529 84.9% | 517 81.9% | 572 83.9% | 374 77.6% | 771 77.0% | 55 41.4% | 403 88.2% | 566 88.0% | 263 70.3% | 502 77.6% | 480 81.4% |
| case 3 Y→N | 5 0.6% | 9 1.4% | 15 2.4% | 15 2.2% | 19 3.9% | 43 4.3% | 6 4.5% | 7 1.5% | 11 1.7% | 17 4.5% | 17 2.6% | 15 2.5% |
| case 4 N→Y | 13 1.6% | 17 2.7% | 25 4.0% | 17 2.5% | 28 5.8% | 55 5.5% | 7 5.3% | 6 1.3% | 12 1.9% | 12 3.2% | 14 2.2% | 19 3.2% |

Table 2: Distribution of the four base cases for identical responses evaluated twice.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | average |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|---------|
| 1/(1 + 3) | 92.9% | 88.3% | 83.1% | 83.9% | 76.3% | 75.4% | 91.5% | 85.4% | 83.1% | 82.8% | 87.0% | 83.6% |
| 2/(2 + 4) | 98.3% | 96.9% | 95.4% | 97.1% | 93.0% | 93.3% | 88.7% | 98.5% | 97.9% | 95.6% | 97.3% | 96.2% |

Table 3: Consistency of assessors’ binary decision on nugget content.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | average |
|------------------|----|----|----|---|----|----|----|---|---|----|----|---------|
| nuggets “lost” | 4 | 4 | 5 | 8 | 9 | 15 | 5 | 3 | 7 | 10 | 9 | 7.2 |
| nuggets “gained” | 11 | 9 | 14 | 8 | 11 | 24 | 6 | 3 | 7 | 6 | 7 | 9.6 |
| net | +7 | +5 | +9 | 0 | +2 | +9 | +1 | 0 | 0 | -4 | -2 | +2.5 |

Table 4: Net balance of nugget assignments when comparing initial and final runs.

this study, since all runs were evaluated at roughly the same time. We only refer to “initial answer string” and “final answer string” for convenience. There are four possible cases that could arise when an identical answer string is independently evaluated twice, enumerated below:

- **Y→Y (case 1):** A nugget assignment was made in both the initial and final answer strings.
- **N→N (case 2):** Neither the initial answer string nor the final answer string received nugget assignments.
- **Y→N (case 3):** The initial answer string was assigned a nugget, but not the final answer string—this represents the situation where the assessor overlooked a nugget.
- **N→Y (case 4):** The initial answer string was not assigned a nugget, but the final answer string received a nugget assignment—this represents the situation where the assessor caught an oversight.

Although in principle it is possible for an assessor to assign more than one nugget to an answer string (i.e., a response containing multiple information nuggets), this seldom happens. For all practical purposes and without affecting subsequent analyses, we can safely treat assessors’ decision as binary: an answer string either contains a nugget or it doesn’t. For now, let us assume that nugget assignment on each answer string happens independently—this is not true, as we discuss in Section 5.

Instance counts and the frequency of the four cases described above, across all 11 run pairs, are listed in Table 2. How consistent are these binary nugget assignments? An analysis of the raw data is presented in Table 3. From this, we can see that in instances where a nugget was assigned to

an answer string in the initial submission, a nugget was also assigned to the same answer string in the final submission on average 83.6% of the time (the first row). In cases where no nugget was assigned to the answer string in the initial submission, the same decision was made in the final submission 96.2% of the time (the second row). The second figure is not surprising, since it is easy to recognize irrelevant responses, but the first figure suggests that binary decisions about the presence or absence of nuggets are relatively consistent.

We note that the distribution of the four cases appear similar across all submitted runs, with the exception of run 7, a manual run. This particular submission was much shorter in length and much more precise, thus skewing the relative proportion of cases 1 and 2. Also, in both manual runs, cases 3 and 4 occur more frequently—this holds implications for our simulation studies, as we shall see later.

What is the net effect of these “flips”? When independently evaluating the same answer string twice, an assessor is likely to miss some nuggets (compared to before) and pick up some other nuggets (not previously found). Table 4 quantifies this by showing the total number of nuggets “gained” (i.e., nuggets assigned in the final submission but not in the initial submission) and the total number of nuggets “lost” (i.e., nuggets assigned in the initial submission but not in the final submission). Overall, the net differences are small, but note the large numbers of nuggets gained and lost for run 6, a manual run. It is not possible to directly translate these numbers into F-score differences since some nuggets are more important than others (either based on the vital/okay distinction or the pyramid nugget weight). The relationship between figures in Table 2 and Table 4 will become apparent in the next section.

For reference, the official NIST answer key contains a total of 447 nuggets for 30 topics (an average of 16.2 nuggets

| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | average |
|---------|-----|------------------|-----|-----|-----|-----|-----|-----|----|-----|-----|-----|-----|---------|
| case 1a | Y→Y | same nugget | 54 | 58 | 60 | 64 | 58 | 115 | 53 | 38 | 50 | 72 | 102 | 65.8 |
| case 1b | Y→Y | diff nugget | 11 | 10 | 14 | 14 | 3 | 17 | 12 | 3 | 4 | 10 | 12 | 10.0 |
| case 2 | N→N | | 731 | 529 | 517 | 572 | 374 | 771 | 55 | 403 | 566 | 263 | 502 | 480 |
| case 3a | Y→N | already assigned | 1 | 5 | 10 | 7 | 10 | 28 | 1 | 4 | 4 | 7 | 8 | 7.7 |
| case 3b | Y→N | other | 4 | 4 | 5 | 8 | 9 | 15 | 5 | 3 | 7 | 10 | 9 | 7.2 |
| case 4a | N→Y | already assigned | 2 | 8 | 11 | 9 | 17 | 31 | 1 | 3 | 5 | 6 | 7 | 9.1 |
| case 4b | N→Y | other | 11 | 9 | 14 | 8 | 11 | 24 | 6 | 3 | 7 | 6 | 7 | 9.6 |

Table 5: Detailed breakdown of the four base cases described in Table 2.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | average |
|-------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|---------|
| count | 26 | 23 | 33 | 30 | 23 | 56 | 23 | 9 | 18 | 26 | 28 | 26.8 |
| % | 3.2% | 3.7% | 5.2% | 4.4% | 4.8% | 5.6% | 17.3% | 2.0% | 2.8% | 7.0% | 4.3% | 4.5% |
| % (−case 2) | 31.3% | 24.5% | 28.9% | 27.3% | 21.3% | 24.3% | 29.5% | 16.7% | 23.4% | 23.4% | 19.3% | 24.5% |

Table 6: Prevalence of inconsistent judgments.

per topic). From Table 2, we see that the overlapping answer strings contain an average of 76 nuggets per run across all topics (the Y→Y case). Although the figures in Table 4 exhibit some variance, the values are relatively small when compared to the total number of nugget assignments made. This appears to support the claim that assessors are sufficiently consistent as to not impact final F-scores by much. The question of *how much* will be taken up in Section 6.

5. DETAILED ANALYSIS

In this section, we revisit the assumption that nugget assignments are made independently for each answer string, which was adopted in the previous analysis for convenience. This is a simplification, because NIST assessors take the complete system output into consideration during the evaluation process. For the ciQA task, they were instructed to assign a nugget to the first answer string that contains the information. Thus, nugget assignments were affected by the ordering of responses in a particular system’s output, which may have changed between the initial and final submissions.

How does this consideration affect our previous analysis? It did not capture the fact that, in many cases, the assessor assigned the same nugget to different answer strings—due to different orderings of the responses or other idiosyncrasies. This does not have an impact on the overall score, but makes assessors seem more inconsistent than they really are. If we take into account order-related dependencies and other details in the nugget assignment process, we arrive at several subcases of the four basic types of nugget assignments shown in Table 2, discussed below.

Just because a nugget was assigned to the same answer string in the initial and final submissions doesn’t necessarily mean that the assessor found the *same* nugget. Case 1 (Y→Y) actually breaks down into two scenarios:

- *same nugget*: the same nugget was assigned to both answer strings.
- *different nugget*: the nugget assigned to the initial answer string is not the same as the one assigned to the final answer string.

There are no subcases for case 2. For case 3 (Y→N) and 4 (N→Y), there are two subcases each:

- *already assigned*: the assignment was inconsistent because the same nugget had already been assigned to another answer string.
- *other*: all other cases.

In fact, the *already assigned* subcases cannot be considered inconsistencies, since they were the correct decisions given the evaluation guidelines (a nugget cannot be assigned twice). The *other* subcases represent situations where assessors’ behavior can not be readily explained.

Table 5 shows counts for all these subcases, which breaks down the figures from Table 2 in greater detail. Case 2 is repeated for convenience. The nuggets “lost” and “gained” in Table 4 correspond exactly to the cases 3b and 4b, respectively.

Considering this more detailed analysis, we see that “inconsistent” judgments are represented by cases 1b, 3b, and 4b. We cannot rule out ordering effects as the cause for cases 3a and 4a, since the assessor appears to be adhering to the evaluation guidelines in those situations. So how often are assessors’ judgments actually inconsistent? This is shown in Table 6. Considering all judgments, NIST assessors are only inconsistent on average 4.5% of the time for each answer string. This figure appears low since it includes judgments about irrelevant answer strings (and there are many of those). The final row of Table 6 shows the same figures, except with instances of case 2 (no nuggets assigned) removed.

6. EFFECTS OF INCONSISTENCIES

What are the effects of inconsistencies in the nugget assignment process? If the same run were assessed multiple times by the same human, we would expect variations in the F-score, but how big would these variations be? This answer is critical because it affects how researchers make comparisons between different techniques: in order to conclude that one system is “better” than another, the element of assessor variability must be accounted for. Confidence in making such comparisons is the key to rapid advances in the state of the art.

We have developed a simulation framework that attempts to answer this question. The analyses performed in the

previous sections yield a characterization of assessor behavior, which we incorporate into simulations. As an approximation, we propose an incremental linear model of assessment, in which each individual answer string is considered in turn. Based on the known outcome (assignments made by NIST assessors), we can randomly perturb the “official” judgments, thereby simulating the effect of independently evaluating the answer string multiple times. In this simple stochastic model, the two relevant probabilities are:

- $P(\text{nug}|y)$: What’s the probability that the assessor would recognize the same information nugget if the exact answer string were judged a second time?
- $P(\text{nug}|n)$: What’s the probability that the assessor would recognize the presence of a nugget that was previously overlooked?

From this, we can simulate the effects of evaluating system output, and by repeating the simulation many times, we can quantify the range of scoring variations. Informally, this allows us to “draw error bars” around system performance, thus facilitating more meaningful comparison between different QA techniques. We describe this in more detail below.

6.1 Simulations

Based on our simple model of assessor behavior, we can simulate outcomes of the same run evaluated multiple times. We accomplish this by iterating over all responses in a submission. At each point, we make the following decision:

- If there is already a nugget assigned to this answer string, then the assignment will remain with $P(\text{nug}|y)$. We randomly generate an event with this probability—if the complement occurs, then the nugget assignment is removed. This corresponds to an assessor overlooking a nugget.
- If there is no nugget assigned to this answer string, then there is a probability $P(\text{nug}|n)$ that one will “appear”. In this case, we randomly assign a nugget that hasn’t already been assigned. This corresponds to the situation where the assessor identifies a nugget that was previously overlooked in the response.

For simplicity, we ignore the actual content of the answer strings (a decision we discuss in Section 8). Also, we do not consider the case where a different nugget is assigned to the same answer string (corresponding to case 1b in Table 5).

Iterating through the answer strings in a particular submission, we arrive at a simulated set of nugget assignments based on the model outlined above. From these assignments, we compute a new F-score for the submission. Repeating this experiment multiple times, we get a range of simulated scores representing how the submission would have fared had it been independently assessed many times.

We then compute the standard deviation of all these simulated F-scores. Assuming the values are normally distributed, we know that approximately 95% of the scores will fall within $\pm 2\sigma$ of the mean. This range, in essence, provides the confidence interval that quantifies variations that can be attributed to assessor inconsistencies in nugget assignment (at least according to our simulation model). Thus, if a submission were independently evaluated multiple times, we can claim with 95% confidence that the F-score will fall within

this range. The upshot is that we have now quantified the effects of judgment variation. Obviously, these computations are dependent on the fidelity of the simulation and the realism of our assessor model. We first present experimental results and defer a detailed discussion to Section 7.

6.2 Results

The assessor model described above was applied to all final submissions from the TREC 2006 ciQA task. For the “flip” probabilities, we employed figures from Table 3. Values in the first row were selected for $P(\text{nug}|y)$ and one minus the values in the second row were used for $P(\text{nug}|n)$. We ran two separate sets of experiments: in the first, individual probabilities were employed for each run (the values in the eleven different columns); in the second, the average probabilities were used for simulating all runs. For convenience, we refer to these as the individual and overall conditions, respectively. Since this method for determining evaluation reliability is untested, we opted to start with the simpler breakdown of assessor inconsistencies, as opposed to the more detailed analysis presented in Table 5. Possible extensions are discussed in Section 8.

For each run, we simulated the scoring process 100 times, and then computed statistics (mean and standard deviation) over these values. Following the official TREC 2006 ciQA task guidelines, we computed pyramid F-scores, which incorporate nugget weights into the recall calculation [8].

Results of the simulations with individual probabilities are shown on the left in Figure 2. Equivalent results from the simulation with overall probabilities are shown on the right in the same figure. In both bar graphs, the bars are sorted in decreasing order of the mean of the simulation results. The error bars indicate the range $\pm 2\sigma$ (around the mean), which defines the 95% confidence interval for judgment variability (as previously described). The solid diamonds show the official NIST F-scores of each run.

The two different sets of experiments produce different results. For the individual probabilities case, official NIST scores fall squarely with our $\pm 2\sigma$ interval for all the final submissions except for those from runs 2, 3, and 5. From Table 1, we see that these are exactly the runs where the initial and final submissions had little overlap—which meant that less data were available for parameter estimation. Otherwise, for many of the submissions, the simulated means were very close to the official F-scores.

The right bar graph in Figure 2 shows results from the overall probabilities condition, in which model parameters were derived from averaging across all runs. This seems to yield a simulation with greater fidelity, as there is a better correlation between the ranking generated by official NIST scores and the ranking generated by the simulation (sorted by the means). However, we note that for the final submissions in runs 6 and 7, simulated scores are very different (much lower) than the official scores. It is no coincidence that both of these are manual runs—a point we will take up in the next section.

7. DISCUSSION

The importance of meta-evaluation in IR research is well established, since progress in the field hinges on accurate and reliable measurements of system performance. This work represents a contribution to question answering evaluation because it is the first in-depth exploration of nugget assign-

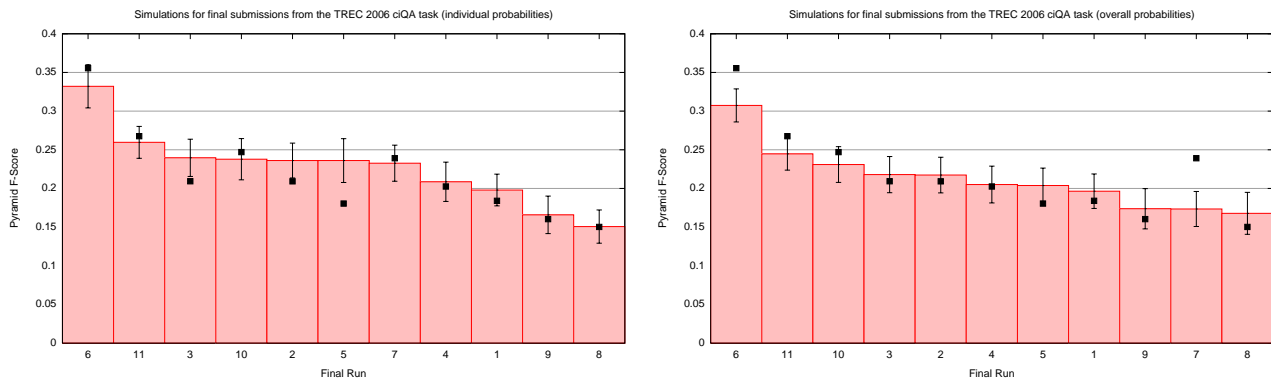


Figure 2: Simulations for final submissions from the TREC 2006 ciQA task with individual probabilities (left) and overall probabilities (right).

ment inconsistencies that we are aware of. Voorhees [11] mentioned scoring variations attributed to assessors, but did not undertake a detailed analysis. In the TREC 2003 QA track, duplicate runs were also independently assessed twice (but for different reasons). For identical runs, she noted a maximum F-score difference of 0.043, with an average of 0.013. Our detailed analysis of assessor behavior and the results of the simulation experiments help researchers better understand the characteristics of nugget-based evaluations. We believe that many insights gained in this study can be broadly applied to other tasks that employ the same evaluation methodology, even though the actual information needs may be different.

The differences between the bar graphs in Figure 2 suggest that it is more desirable to construct a general model of assessor behavior using aggregate data from all submissions. The system ranking generated with overall probabilities correlates better with the official NIST ranking. This makes sense, as we expect a uniform distribution of assessor errors in general. This assumption, however, appears to be violated in some situations, as we explain.

Most complex QA systems today employ some variant of passage or sentence retrieval. Since they are fundamentally extractive, the quality of the answer strings is relatively uniform. Put another way, if one were to build language models of responses, they would be expected to have relatively low cross-entropy. Thus, we should observe the same types of assessor inconsistencies across all submissions. This intuition is borne out by our analyses, as the distribution of the “nugget flip” cases is fairly similar across automatic runs. Our observations are also confirmed by the simulation experiments, where official scores fall within the confidence interval derived from our model of assessor inconsistencies. However, human involvement in manual runs might make them qualitatively different from primarily extractive runs—thus, we might expect different types of inconsistencies.

In particular, estimation of $P(\text{nug}|n)$ is sensitive to two factors: the underlying performance of the run and the length of the response. One would expect runs with human involvement to be better. And the better the system output is to begin with, the greater the chance that a response will actually contain a nugget. With respect to the second point, $P(\text{nug}|n)$ is derived from overlapping responses between initial and final submissions, and the number of overlapping

responses is at least indirectly related to the length of each submission. Implicit in the parameter is the assumption of equal length across all submissions—otherwise, longer runs would be unfairly rewarded because they generate more random events by the simulation (i.e., $N \rightarrow Y$ flips) and shorter runs would be unfairly penalized for the opposite reason.

We believe that these observations explain why the simulation model does not agree with official NIST scores for the manual runs 6 and 7. The first contained manually-selected sentences “padded” with results from a sentence retrieval algorithm. Therefore, some of the responses were inherently better than others. In addition, we see from Table 2 that run 6 had a proportionally larger number of $Y \rightarrow N$ and $N \rightarrow Y$ flips, probably caused by human involvement in preparing the submission. Due to these factors, our simulation model does not appear to accurately capture assessor behavior. The explanation for why our mean simulation score diverges from the actual NIST score for run 7 appears simple: the submission was much shorter in length compared to the others, thus reducing the likelihood of generating $N \rightarrow Y$ flips.

In general, differences in run length also explain the relationship between the mean simulation F-scores and the official NIST F-scores (Figure 2, right). Considering results from the overall probabilities condition, our experiments appear to have underestimated the performance of runs 11 and 10, which are shorter than average. Similarly, our experiments may have overestimated the performance of run 5, the longest.

8. EXTENSIONS

Simulations have been previously employed as a general method to quantify evaluation stability and reliability—for example, Zobel’s “take one out” experiments [16] demonstrated that *ad hoc* test collections built from pooling are reusable; Lin [6] applied similar methods to examine resources for factoid QA evaluation. Since it is impractical to manually assess system output repeatedly, simulations offer an attractive alternative. Our approach, however, is different in trying to better understand assessor behavior and incorporating this knowledge into actual models of the assessment process. Although the current implementation is relatively crude, our general approach can certainly be extended with more faithful representations. In this section,

we discuss a number of possible refinements.

First, our model of the evaluation process does not actually take content into account—that is, the probabilities of nugget assignment are not conditioned on any features of the answer string. Intuitively, one would expect this to be an important factor, since the semantic distance between an answer string and an answer nugget might affect its “confusability”. One might conceivably incorporate some content-based metric, for example, nugget match according to POURPRE [7]. However, it would be very difficult to realistically model this effect, and the extra degree of freedom in estimating parameters might cause data sparseness to become an issue.

Second, we assume that each answer string is independently assessed in sequential order. In reality, assessors assign nuggets in a more “holistic” fashion—they consider the responses and the answer nuggets simultaneously. The assessor is unlikely to read system output in a strict linear fashion, and most likely scans the answer strings repeatedly, focusing on regions of interest. In addition, effects such as assessor fatigue may be significant for long submissions—for excessively lengthy outputs, an assessor might at some point simply stop reading. None of these considerations are incorporated into the static probabilities in our assessor model. Specifically, the issue of answer length is of concern, since our experiments have revealed it to be an important factor in simulation fidelity.

Third, it might be insightful to model inconsistencies for individual NIST assessors, since they might be prone to different types of errors. Although such information is not presently available to researchers, to our knowledge NIST does keep records of the mapping between assessors and topics. The setup of the evaluation ensures that one single assessor examines all runs. Thus, inter-assessor differences might be more pertinent than inter-run differences.

Obviously, the validity of any simulation result hinges on the fidelity of the underlying model. A more faithful representation will yield a better characterization of scoring variations. A better understanding of this “noise” will help researchers more accurately diagnose system performance. Thus, explorations of evaluation methodology must keep pace with system development, suggesting a constant need for careful meta-evaluation.

Although we have no doubt that our simulation-based method can be applied to analyze other instances of complex QA evaluation (for example, “other” questions in TREC), it remains to be seen if model parameters generalize across different tasks. In other words, do the values $P(\text{nug}|y)$ and $P(\text{nug}|n)$ quantify intrinsic human variability, or are they task specific? This remains an open question, and one we are keen on exploring.

9. CONCLUSION

Due to fortunate circumstances, the TREC 2006 ciQA task provided a unique look into the nugget-based evaluation methodology that is commonly employed in complex QA. Analysis of run data yielded statistics about assessor inconsistencies that were then used to develop a model of the evaluation process. Simulation experiments using these models allowed us, for the first time in QA evaluation, to quantify the effects of assessor inconsistencies in nugget assignment. These results help us make more meaningful comparisons between systems, isolating actual differences from

human-attributable scoring variations. This ability provides the community with a more accurate compass for charting research progress.

10. ACKNOWLEDGMENTS

This work has been supported in part by DARPA contract HR0011-06-2-0001 (GALE). We are grateful to Ellen Voorhees, Hoa Dang, and all the NIST assessors for making TREC possible; also, to Diane Kelly for her role in making the ciQA task possible. The first author would like to thank Kiri and Esther for their kind support.

11. REFERENCES

- [1] J. Allan. HARD track overview in TREC 2005: High accuracy retrieval from documents. In *Proceedings of TREC 2005*.
- [2] C. Buckley and E. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the SIGIR 2004*.
- [3] B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In *Proceedings SIGIR 2006*.
- [4] C. Cleverdon, J. Mills, and E. Keen. Factors determining the performance of indexing systems. Two volumes, ASLIB Cranfield Research Project, Cranfield, England, 1968.
- [5] W. Hildebrandt, B. Katz, and J. Lin. Answering definition questions with multiple knowledge sources. In *Proceedings of HLT/NAACL 2004*.
- [6] J. Lin. Evaluation of resources for question answering evaluation. In *Proceedings SIGIR 2005*.
- [7] J. Lin and D. Demner-Fushman. Automatically evaluating answers to definition questions. In *Proceedings HLT/EMNLP 2005*.
- [8] J. Lin and D. Demner-Fushman. Will pyramids built of nuggets topple over? In *Proceedings of HLT/NAACL 2006*.
- [9] J. Lin and B. Katz. Building a reusable test collection for question answering. *JASIST*, 57(7):851–861, 2006.
- [10] M. Sanderson and J. Zobel. Information retrieval system evaluation: Effort, sensitivity, and reliability. In *Proceedings SIGIR 2005*.
- [11] E. Voorhees. Overview of the TREC 2003 question answering track. In *Proceedings of TREC 2003*.
- [12] E. Voorhees. Overview of the TREC 2004 question answering track. In *Proceedings of TREC 2004*.
- [13] E. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *IP&M*, 36(5):697–716, 2000.
- [14] E. Voorhees and H. Dang. Overview of the TREC 2005 question answering track. In *Proceedings of TREC 2005*.
- [15] E. Voorhees and D. Tice. Building a question answering test collection. In *Proceedings SIGIR 2000*.
- [16] J. Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proceedings SIGIR 1998*.