A Nugget-based Test Collection Construction Paradigm

Shahzad K. Rajput Virgil Pavlu Peter B. Golbus Javed A. Aslam

College of Computer and Information Science
Northeastern University
360 Huntington Ave, #202 WVH
Boston, MA 02115, USA
{rajput, vip, jaa}@ccs.neu.edu

ABSTRACT

The problem of building test collections is central to the development of information retrieval systems such as search engines. The primary use of test collections is the evaluation of IR systems. The widely employed "Cranfield paradigm" dictates that the information relevant to a topic be encoded at the level of documents, therefore requiring effectively complete document relevance assessments. As this is no longer practical for modern corpora, numerous problems arise, including *scalability*, *reusability*, and *applicability.*

We propose a new method for relevance assessment based on relevant *information*, not relevant *documents.* Once the relevant information is collected, any document can be assessed for relevance, and any retrieved list of documents can be assessed for performance. Starting with a few relevant "nuggets" of information manually extracted from existing TREC corpora, we implement and test a method that finds and correctly assesses the vast majority of relevant documents found by TREC assessors, as well as up to four times more additional relevant documents. We then show how these inferred relevance assessments can be used to perform IR system evaluation. Our main contribution is a methodology for producing test collections that are highly accurate, more complete, scalable, reusable, and can be generated with similar amounts of effort as existing methods, with great potential for future applications.

1. INTRODUCTION

Collections of retrieval systems are traditionally evaluated by (1) constructing a test collection of documents, (2) constructing a test collection of queries, (3) judging the relevance of the documents to each query, and (4) assessing the quality of the ranked lists of documents returned by each retrieval system for each topic using standard measures of performance such as precision-at-cutoff, nDCG, average precision, and so forth. Much thought and research has been devoted to each of these steps in, for example, the annual text retrieval conference TREC [12].

For large collections of documents and/or topics, it is impractical to assess the relevance of each document to each topic. Instead, a small subset of the documents is chosen, and the relevance of these documents to the topics is assessed. When evaluating the performance of a collection of retrieval systems, as in the annual TREC conference, this judged "pool" of documents is typically constructed by taking the union of the top c documents returned by each system in response to a given query. In TREC, $c = 100$ has been shown to be an effective cutoff in evaluating the relative performance of retrieval systems. Both shallower and deeper pools have been studied [22, 12], both for TREC and within the greater context of the generation of large test collections. Pooling is an effective technique since many of the documents relevant to a topic will appear near the top of the lists returned by (quality) retrieval systems; thus, these relevant documents will be judged and used to effectively assess the performance of the collected systems; unjudged documents are assumed to be non-relevant.

This process, often referred to as the "Cranfield paradigm" for information retrieval evaluation, essentially operates in two phases: Phase 1, "Collection Construction", constitutes Steps (1) through (3) above—documents, topics, and relevance assessments are all collected. Following Phase 1, we have a *test collection* that can be used to evaluate the performance of systems in Phase 2, "Evaluation" (Step (4) above). Note that evaluation can be performed on the systems that contributed to the pool, and perhaps even more importantly, it can be performed on new systems that did not originally contribute to the pool. A test collection is *accurate* if it correctly assesses the performance of systems that contributed to the pool, and it is *reusable* if it correctly assesses the performance of new systems that did not originally contribute to the pool. That a test collection must be accurate is a given, but for a test collection to be truly useful, it must also be reusable: New information retrieval technologies will be tested against existing test collections, as happens continually with the various TREC collections, and for those assessments to be meaningful, these test collections must be reusable. In order for a Cranfield paradigm test collection to be both *accurate* and *reusable*, the relevance assessments must be *effectively complete.* In other words, the vast majority of relevant documents must be found and judged; otherwise, a novel retrieval system could return unseen relevant documents, and the assessment of this system with respect to the test collection will be highly inaccurate. Unfortunately, the burden of effectively complete assessments is quite large; in TREC 8, for example, 86,830 relevance judgments were collected in order to build a test collection over a relatively small document collection for just 50 topics.

1.1 Limitations of the Cranfield Paradigm

The key limitation of the Cranfield paradigm is that (1) during collection construction *the information relevant to a topic is encoded by documents* and (2) during evaluation *the information retrieved by a system is encoded by documents.* Thus, in order to assess the performance of a system, one must determine which relevant documents are retrieved (and how), and this necessitates effectively complete relevance judgments.

Other retrieval tasks engender variants on the Cranfield paradigm, but they all tend to retain the central feature above, that the *information* relevant to a topic is *encoded* by documents. The difference is that other metadata is often collected which is specific to the retrieval task. For example, *Graded Relevance* was introduced in web search; instead of documents being "relevant" or "non-relevant", they can be "highly relevant", "relevant", "marginally relevant", or "non-relevant". However, the information relevant to a topic is still encoded by documents (together with their relevance grades), and the information retrieved by a system is also encoded by documents. For *Novelty and Diversity* measurements, the information relevant to a query is encoded by documents, and the information retrieved by a system is encoded by documents and their "marginal utility" with respect to previously retrieved documents, or associated "subtopics" and some measure of the coverage of those documents over the subtopics.

This central feature of the Cranfield paradigm and its variants, that the information relevant to a topic is encoded by documents, resulted in hundred of thousands of documents analyzed, both by governmental organizations (TREC, NTCIR) and large corporations (Google, Microsoft). Even so, and critical to evaluation, many relevant documents are missed; ultimately, it gives rise to several related problems:

1. **Scalability:** Given that the information relevant to a topic is encoded by documents, and given the necessity of effectively complete relevance assessments that this entails for accurate and reusable evaluation, the Cranfield paradigm and its variants cannot scale to large collections and/or topic sets. For example, the query "Barack Obama" yields 65,800,000 results on Google as of December 2010, and it would be impossible to judge all at any reasonable cost or in any reasonable time.

2. **Reusability:** The problem of scale directly gives rise to problems of reusability: (1) For a static collection, novel systems will retrieve unjudged but relevant documents, and the assessments of these systems will be inaccurate. (2) For dynamic collections (such as the World Wide Web), new documents will be added and old documents removed, rendering even statically constructed "effectively complete" relevance assessments incomplete over time, with an attendant loss in reusability.

3. **Applicability:** It can be difficult to apply a test collection designed for one retrieval task and evaluation to another retrieval task or evaluation, especially for test collections that are designed to "fix" the scale and reusability issues described above using current methodologies. This issue is discussed below.

As described below, our new methodology successfully addresses all three of these issues.

1.2 Related work

Various attempts to address the issues described above have been proposed. (1) Sampling techniques such as infAP [21], statAP [8], and their variants have been used extensively in various TREC tracks, including the Million Query Track, the Web Track, the Relevance Feedback Track, the Enterprise Track, the Terabyte Track, the Video Track, and the Legal Track. These techniques are designed to directly address the *scale* issue described above. A carefully chosen sample of documents is drawn from the pool, these documents are judged, and a *statistical estimate* of the true value of a performance measure over that pool is derived. Given that accurate estimates can be derived using samples as small as roughly 5% of the entire pool, these methods permit the use of pools roughly 20 times the size of standard fully-judged pools. This increases reusability, for example; however, it is only a stop-gap measure. These methods cannot scale to collections the size of the web (where potentially 65 million documents are relevant to a query such as "Barack Obama"), they only partially address the issue of dynamic collections such as the web, and they reduce applicability in that the samples drawn and estimates obtained are typically tailored to specific evaluation measures such as average precision. (2) The Minimal Test Collection methodology [7] also employed in the TREC Million Query Track has generally similar benefits and drawbacks, as described above. (3) Crowd-sourcing relevance judgments, via Mechanical Turk, for example, has also been proposed to alleviate the scale issue [1]. However, this too is only a stop-gap measure, in roughly direct proportion to the relative ease (in time or cost) of crowd-sourced judgments vs. assessor judgments: If 10 to 100 crowd-sourced judgments can be obtained in the same time or at the same cost as 1 assessor judgment, then pools one to two orders of magnitude larger than standard pools can be contemplated, but this still does not scale to the web or address the issue of dynamic collections, as described above. The use of click-through data has also been proposed [16], but this is only applicable to the web and only for those documents with sufficient "clicks".

In order to address the inherent limitations of the Cranfield paradigm and variants thereof described above, we propose a test collection construction methodology based on *information nuggets*. We refer to minimal, atomic units of relevant information as "nuggets". Nuggets can range from simple answers such as people's names to full sentences or paragraphs. In this model, assessors indicate as relevant only the relevant portions of documents. This relevant *information* is used to automatically assign relevance judgments to documents and/or evaulate retrieval systems.

We note that nuggets of a somewhat different kind are widely used in other contexts. For example, the evaluation of question answering systems [13, 14, 10, 18, 15] uses nuggets, which in this context tend to be very short and specific answers to "who", "when", and "where" type questions. Nuggets have also been used as a form of user feedback in multi-session information distillation tasks [20, 19]. Conceptual nuggets are currently used in novelty and diversity evaluation: subtopics can be thought of as nuggets [9], or systems can be evaluated on coverage of both subtopics and nuggets [2]. However, in none of these contexts are nuggets used to infer relevance automatically. All of the above are still susceptible to the limitations of the Cranfield paradigm.

2. METHODOLOGY

The central issue with the Cranfield paradigm and with its variants described above is that the information relevant

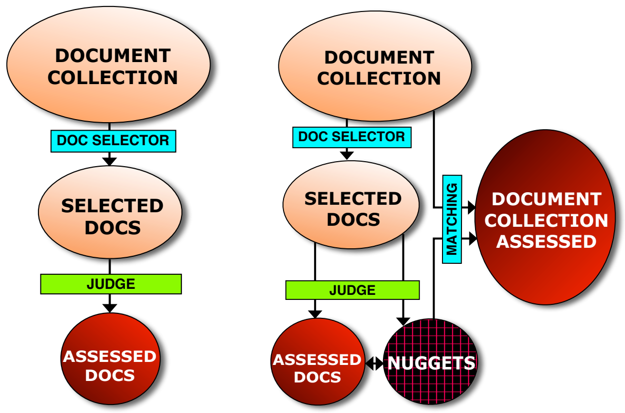


Figure 1: For a given query, selected documents are evaluated as "relevant/nonrelevant". Left: traditional TREC strategy for relevance. Right: proposed nuggets method.

to a topic is encoded by *documents*, and in the presence of large topic sets or large and/or dynamic collections, it is difficult or impossible to find and judge all relevant documents. Our thesis is that while the number of *documents* potentially relevant to a topic can be enormous, the amount of *information* relevant to a topic, the nuggets, is far, far smaller. For example, consider the web query "Barack Obama". The vast majority of the potentially 65 million documents relevant to this topic probably do not contain any information that could not be found in his biography, or even just his Wikipedia page. Furthermore, relevant documents are constantly being created and destroyed, whereas major changes to the set of relevant information are much less frequent. Thus, collecting and encoding the relatively small set of relevant nuggets, as opposed to the dynamically changing and effectively infinite set of *documents* relevant to a topic, will enable us to address the issues of scalability, reusability, and applicability described above.

Figure 1 graphically illustrates the differences between the traditional Cranfield-style evaluation methodology (left) and the nugget-based methodology proposed (right). The nuggets themselves are the relevant and useful pieces of information for a given topic—the information that the user seeks. As a set, they yield a natural encoding of the information relevant to a topic. In principle, if this set is complete, we can use the nuggets to infer the relevance of any document.

To build our test collection, we ask assessors to view documents as before. However, rather than providing binary or graded "relevance judgments", we instead ask the assessor to extract nuggets. Our thesis is that while collecting effectively complete document relevance judgment sets is impractical (on a large scale) or impossible (in a dynamic environment), collecting effectively complete nugget sets is much more tractable. Certainly the problem of collecting effectively complete nugget sets is no harder than the problem of collecting effectively complete relevance judgment sets: any effectively complete set of relevant documents must collectively contain an effectively complete set of nuggets, by definition, and the judges would find these nuggets at the time of assessment. However, an effectively complete set of relevant documents will contain vast quantities of highly redundant information (nuggets or their variants), and thus the effectively complete set of nuggets will likely be vastly smaller, more tractable, and more easily found with far fewer total judgments.

In Phase 2 of our nugget-based evaluation paradigm ("Evaluation"), we dynamically assess the relevance of documents retrieved by a system under evaluation, using the nuggets collected. This is accomplished, in principle, by matching the information relevant to a topic (as encoded by the nuggets) with the documents in question. For small collections and/or recall-oriented measures of performance, one could in principle automatically assess the entire collection via nuggets. In practice, even if the set of nuggets collected is not complete, the inferred set of relevant documents is significantly larger than the set obtained by assessing documents. The proposed method also permits the use of all standard measures of retrieval performance, and probably of all future measures of interest.

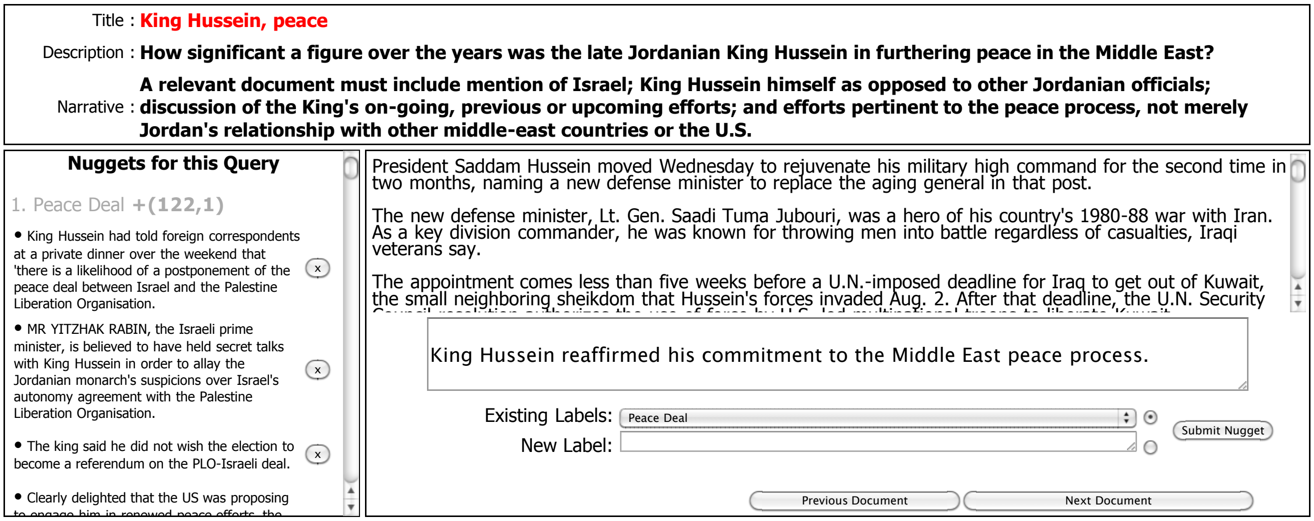
We further note that this nugget-based evaluation paradigm can be easily extended to accommodate other retrieval tasks:

- Graded Relevance: Nuggets can be graded at the time of extraction, in much the same manner that documents are graded, and the matching function can take these nugget grades into account when assigning grades to documents: the "stronger" the match with more "relevant" nuggets, the "higher" the overall grade.
- Novelty: Nuggets could be clustered or categorized, either automatically or by the assessor. A document could then be judged relevant if it contains relevant information (as above), but its *marginal utility* will only be high if it contains relevant information not already seen in the output list, i.e., information from heretofore unseen nugget clusters or categories.
- Diversity: Nuggets can be assigned to aspects or subtopics by the assessor. Then the *coverage* of a document or list can be determined by matching information across nugget aspect or subtopic classes, thus permitting diversity-based evaluation.

3. PILOT IMPLEMENTATION

Building a test collection in our framework consists of three distinct tasks: (1) selecting documents from which to extract nuggets, (2) extracting nuggets from those documents, and (3) using the extracted nuggets to algorithmically create relevance judgments for any desired subset of the corpus. Each of these tasks could be performed in various ways; in this section, we document the decisions we made in implementing our methodology.

Document Selection. Any human assessment of documents must use a selection procedure, e.g. sampling or pooling. Typically for this selection, documents retrieved by many systems and/or at higher ranks are preferred to documents retrieved by fewer systems and/or at lower ranks. While virtually all known selection mechanisms can be used, we preferred in our implementation one that balances preference for top/often documents with coverage (or depth) of the sample; We used the statAP selection mechanism because it is designed specifically on this principle applied to Average Precision measure, and it has been shown to be an effective document selection procedure in previous TREC ad hoc tracks for system evaluation [8].



Title : King Hussein, peace

Description : How significant a figure over the years was the late Jordanian King Hussein in furthering peace in the Middle East?

Narrative : A relevant document must include mention of Israel; King Hussein himself as opposed to other Jordanian officials; discussion of the King's on-going, previous or upcoming efforts; and efforts pertinent to the peace process, not merely Jordan's relationship with other middle-east countries or the U.S.

Nuggets for this Query

1. Peace Deal +(122,1)

- King Hussein had told foreign correspondents at a private dinner over the weekend that 'there is a likelihood of a postponement of the peace deal between Israel and the Palestine Liberation Organisation.
- MR YITZHAK RABIN, the Israeli prime minister, is believed to have held secret talks with King Hussein in order to allay the Jordanian monarch's suspicions over Israel's autonomy agreement with the Palestine Liberation Organisation.
- The king said he did not wish the election to become a referendum on the PLO-Israeli deal.
- Clearly delighted that the US was proposing to engage him in renewed peace efforts, the

President Saddam Hussein moved Wednesday to rejuvenate his military high command for the second time in two months, naming a new defense minister to replace the aging general in that post.

The new defense minister, Lt. Gen. Saadi Tuma Jubouri, was a hero of his country's 1980-88 war with Iran. As a key division commander, he was known for throwing men into battle regardless of casualties, Iraqi veterans say.

The appointment comes less than five weeks before a U.N.-imposed deadline for Iraq to get out of Kuwait, the small neighboring sheikdom that Hussein's forces invaded Aug. 2. After that deadline, the U.N. Security Council resolution authorizes the use of force by U.S.-led multinational troops to liberate Kuwait.

King Hussein reaffirmed his commitment to the Middle East peace process.

Existing Labels: Peace Deal ●

New Label: ○

Figure 2: Nugget extraction interface.

Nugget Extraction. Nugget extraction was performed by our internal assessors (primarily graduate students working on IR research). For each *relevant* document in the sample, the assessor was asked to extract the relevant nuggets (see Figure 2 for the nugget extraction interface). They were instructed to find the smallest part of text that constitutes relevant information in itself; however nuggets are not restricted to text as it appears in the document: slight modifications of the text, e.g. co-reference disambiguation, deleting contextual stopwords, etc. were encouraged. In the end, the vast majority of nuggets are relevant information encoded in the form of actual text contained in relevant documents.

Assessors were also given the option of adding query keywords, which would be used later as a retrieval filter. If a query has keywords associated with it, a document must contain at least one keyword to be considered relevant for that query. For example, consider the topic "JFK assassination". An assessor might add the keyword "`Kennedy`". If a document does not contain this term, it will not be considered to match any nugget.

Inferred Relevance Judgements. According to the typical TREC definition of relevance for ad hoc retrieval, a document is considered relevant if it contains a single relevant piece of information. Thus if a document contains a known relevant information nugget, then it is necessarily relevant. However, a document may "match" the *information* or *meaning* of a nugget, without matching verbatim the nugget text; the matching strategy has to account for possible mismatches of *text* that are in fact matches of *information*. There are some simple approaches one could use for such a matching strategy, e.g. matching based only on text, like our own implementation below; there are also some complicated approaches to this problem: NLP-based, thesaurus-synonyms-ontology, statistical clustering including mutual information techniques, language dependence learning like CRF, machine learning, etc.

To test our methodolgy, we implemented a text-based matching algorithm that automatically infers the relevance of documents given the nuggets extracted. Each document received a relevance score after matching with all nuggets. The matching algorithm is based on a variant of *shingle matching*, which is often used in near-duplicate detection [5, 6]. A shingle is a sequence of k consecutive words in a piece of text. For example, after stopwording, the nugget `"John Kennedy was elected president in 1960"` has the following shingles for $k = 3$: (`John Kennedy elected`), (`Kennedy elected president`), and (`elected president 1960`).

Given the set of nuggets, we computed a relevance score for each document by (1) computing a score for each shingle, (2) combining these shingle scores to obtain a score for each nugget, and (3) combining these nugget scores to obtain a score for the document:

- **Shingle score:** For any nugget and each shingle of size k, let S be the minimum *span* of words in the document that contains all shingle words in any order. A shingle matches well if it is contained in a small span. We define the shingle score as follows

  $$shingleScore = \lambda^{(S-k)/k}.$$

  where λ is a fixed decay parameter. A shingle that matches "perfectly" will have a score of 1. Note that, in contrast to standard *shingle matching* used for duplicate detection, we do not require all shingle words to be present in the matching document in the same order or contiguously.

  Our method is inspired by near-duplicate detection, but is in fact quite different. High scores indicate a match of known relevant information, not necessarily of redundant or duplicate text. Furthermore, while a known nugget is required to be present in a document for a good match, the document often contains new/unknown relevant information as well.

- **Nugget score:** To obtain a score for each nugget, we average the scores for each of its shingles.

  $$nuggetScore = \frac{1}{\#shingles} \sum_{s \in shingles} shingleScore(s)$$

We note briefly that we have explored learning algorithms for the combination of nugget scores into a document score, using the sample as a training set of documents. So far our conclusion is that the improvement (if any) in performance of such techniques like Regression or Boosting does not justify the increase in complexity compared to simple functions like "`max`".

- **Document score:** Each document gets a relevance score equal to the maximum matching score with any nugget:

$$docScore = \max_{n \in nuggets} nuggetScore(n)$$

- **Inferred relevance judgment:** We convert a document relevance score to an inferred relevance score by performing a simple thresholding, i.e. if a document score is above the threshold θ, then the document is considered to be inferred relevant. This threshold is found by trial and error, but it is constrained to be a constant across all experiments; a better performance can be obtained by setting the threshold differently for each query or experiment – such variable thresholds could be learned from data.

4. ANALYSIS

In this section, we validate the performance of our method, show that our method requires far less human effort while producing many more assessments than the traditional procedure, and analyze the causes if disagreement between inferred judgements and TREC assessments. To this end, we constructed two separate test collections based on well-studied collections produced by previous TREC tracks.

The first experiment uses ad hoc retrieval data from the TREC 8 ad hoc task: a collection of about half million newswire articles (in clean text) considered to have effectively complete assessments (depth-100 pool), with an average of about 1,736 assessed documents for each of 50 queries. There were 129 IR systems submitted to the TREC 8 ad hoc task; we refer to this data collectively (documents, judgments, queries, systems) as "ad hoc".

The second experiment is based on data from the TREC09 web track, which uses the ClueWeb09 html collection of about one billion documents; it contains an average of only about 528 documents assessed per query; it is considered to have incomplete assessments. Queries are shorter, but have specific subtopics. About 120 IR systems were submitted to TREC for this task. This data is referred to as "web".

Using statAP sampling, we selected 200 documents for each query from each collection. Of these documents, we extracted nuggets from only those that had been judged *relevant* by TREC assessors (we did not assess relevance at this stage; we did assess relevance on new documents for web data, later, as validation). The TREC 8 ad hoc collection sample, denoted "SampleAdHoc", was approximately 11% of the full pool assessed by TREC. The TREC09 web sample, denoted "SampleWeb", contains approximately 38% of the full pool assessed by TREC.

On average, about 87 nuggets were extracted per query for the ad hoc sample and about 62 nuggets were extracted per query for the web sample (Table 1).

Sample	Documents	Relevant Documents	Nuggets
SampleAdHoc	200	34.02	86.98
SampleWeb	200	25.18	61.82

Table 1: Sample statistics (query average)

Truncated Result List	MAP	Precision	Recall	F1
SampleAdHoc	0.48	0.18	0.47	0.26
SampleAdHoc+InfRel	0.76	0.82	0.65	0.73
SampleWeb	0.24	0.23	0.25	0.24
SampleWeb+InfRel	0.75	0.88	0.60	0.71

Table 2: Matching performance: Inferred relevance judgments compared directly to published TREC assessments.

4.1 Validation

There are two main features of our method that require validation. Not only must we demonstrate that our inferred relevance judgments are correct, we must also show that information is redundant enough that even nuggets represented as *strings* can return far more relevant documents that traditional methods. In short we must validate our method with respect to both precision and recall.

The notion of correctness of relevance judgments is somewhat problematic. Inter-assessor disagreement [4, 17] is a well known phenomenon–the question of relevance is ambiguous for many documents. Bearing this in mind, we demonstrate the correctness of our inferred judgments in two ways: by comparing our judgments to the judgments provided by TREC, and by verifying with independent human assessors the inferred relevance of documents outside TREC qrel.

Given the nuggets extracted, we employ the matching algorithm with a threshold of 0.8 to infer binary relevance for all documents retrieved by any system. For a fair comparison, we produce our own nuggets-based qrel as the inferred relevance assessments on TREC judged documents, and refer to it based on the sample of documents from which nuggets were extracted, e.g. "SampleAdHoc+InfRel(Nuggets)" refers to the judgments by TREC assessors of documents in the ad hoc sample, plus the judgments inferred for the other documents.

We can assess the quality of our inferred relevance judgments by either sorting documents by their document relevance scores and creating a ranked list, or thresholding to create a qrel file. Restricting our list to only those documents judged by TREC, we compute an average precision of AP=0.75 or better for our ranked list, which implies the vast majority of relevant documents are ranked higher than non-relevant ones.

After thresholding, we can compare our obtained qrel against the published TREC qrel in terms of precision, recall, F1, etc (see Table 2). This process is sometimes referred to as qrel inference. For comparison, a previously published result on qrel inference [3] cites an F1 of 0.68 (compared with our F1=0.73). The previously published result required significant extra ranking structural information, which is highly contextual and may be not available.

To further test the correctness of our inferred relevance judgments, as well as test the hypothesis that our method finds many additional relevant documents, we also used the nuggets extracted from SampleWeb to infer the relevance of

	Within qrel	Outside qrel*	Total
Judged Rel	2969	14624	17593
Judged NonRel	411	5329	5740
Total	3380	19953	23333
Agreement	87.84%	73.29%	75.40%

Table 3: Agreement of Inferred Relevance With Judged Relevance (* denotes an estimate with 99% confidence, based on a validated sample of about 4000 documents).

documents retrieved within top 300 ranks by any web system (depth-300 pool) – about 5891 documents per query. Our matching algorithm marked about 430 documents relevant per query; on average 360 of these were unjudged by TREC(see Table 3).

Validation of the inferred relevance assessments outside the TREC qrel was performed by taking a uniform random sample of about 80 TREC-unjudged documents per query, and having them assess for relevance by humans using Amazon's Mechanical Turk service [1]. If a document had multiple assessments for a given query, the majority vote was used. In case of a tie, the document was discarded from measurement.

The results of this experiment showed an agreement of 73.29% between the Mechanical Turk judges and our inferred assessments on these unjudged documents. Given the sample size of about 4000 documents, there is a 99.9% statistical confidence that the number of relevant documents outside the TREC qrel is at least 14,223 (maximum likelihood estimate is 14,624). Overall, even using only nuggets extracted from a small sample of assessed documents, our method created relevance judgments that were both highly correlated with existing qrels and validated by human assessors with very high accuracy (precision), and also found many relevant documents not found by TREC (recall).

4.2 Effort Required

Since our samples are small compared to those judged by TREC, our methodology requires significantly less human effort. We found that extracting nuggets from a relevant document took roughly four times longer than providing a binary relevance judgment for that document. No additional time is required to extract nuggets from non-relevant documents.

TREC assessors judge between 50 and 100 documents an hour.[2] For the sake of computation, assume that it requires one minute to assess each document. At that rate, the entire TREC 8 ad hoc qrel took about 36 man-weeks to produce. SampleAdHoc required about 4 man-weeks for binary relevance assessments. For the relevant documents found in the sample, we spent an additional 2.1 man-weeks on extracting nuggets; thus the total human effort required for our method on SampleAdhoc is about 6.2 man-weeks.

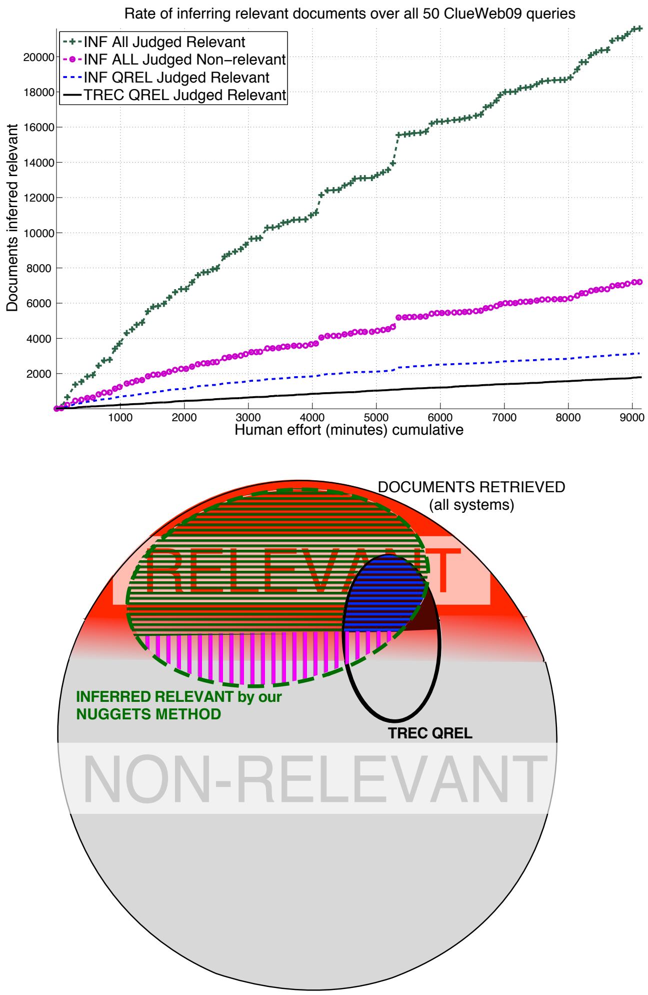


Figure 3: The top plot shows the rate of finding relevant documents per unit of time: TREC qrel (black), nuggets inferred from TREC qrel (blue), nuggets inferred–assessor disagreement (pink) and nugget inferred–assessor agreement (green). The result of this process is shown in the bottom diagram, which illustrates the vast number of inferred relevant documents our method finds.

Under the same assumption, TREC spent about 11 man-weeks creating the full web qrel of about 26,000 documents. SampleWeb, which is about 38% the size of entire full qrel, required about 4 man-weeks. Nugget extraction from relevant documents in the sample took another 1.6 man-weeks, for a total human effort on SampleWeb of about 5.6 man-weeks.

Our method required *substantially* less time to produce both collections.

4.3 Failure analysis

While it is sometimes difficult to decide which of two conflicting relevance judgments is correct, it is often easy to determine that one of them is wrong. In Table 4, we categorize about 400 instances of conflicting relevance judgments between TREC assessors and our inferred judgments. The analyzed documents represent the most egregious errors in terms of document score and TREC relevance assessment.

[1] mturk.com. Each Mechanical Turk job was verified for quality: each job consisted of 30 documents out of which 10 were verification documents with known TREC assessments, and they were required to correctly assess 70% of these 10 documents; if below 70% threshold on verification documents, the job was not accepted. Furthermore, some jobs were performed by multiple judges.

[2] Private communication with TREC organizer.

	Sources of Errors	Ad Hoc(%)	Web(%)
1	Assessor disagreement	46 (–)	71 (–)
2	Relevant document marked spam (independent filter [11])	N/A	45 (–)
3	Missing nugget/aspect	64 (50.3%)	47 (55.9%)
4	Matching problems	12 (9.5%)	2 (2.4%)
5	Non-relevant document matches "bad" nugget	39 (30.7%)	4 (4.7%)
6	Relevant document doesn't match "bad" nugget	7 (5.5%)	14 (16.7%)
7	Relevance cannot be captured by a text string	5 (3.9%)	11 (13.1%)
8	Nugget doesn't match due to HTML parsing	N/A	6 (7.1%)

Table 4: Failure Analysis. Percentages are computed out of total count excluding disagreement and spam.

We describe 7 reasons that this may occur (see Table 4).

1. Assessor disagreement – In this case, upon visual inspection, either we agreed with our inferred judgment and disagreed with the TREC judgment, or else we felt that either could be considered correct.

2. Relevant document marked as spam - We did not apply our matching algorithm to documents in the ClueWeb09 corpus that were marked as spam by the Waterloo spam filter [11]. Our method does not address the spam problem in any way; the filter used is totally independent of our matching method and can be used, or not, or replaced with any other filter.

3. Missing nugget – We may have missed a relevant document because we did not have an appropriate nugget. It may be that we did not have any nuggets covering a certain aspect of the query. This can be fixed by extracting more nuggets.

4. Matching problems – Either the match is not exact due to limitations of our particular matching function (for example not recognizing synonyms), or the existing semantic match cannot be expressed in text. This can be fixed by employing more complex matching functions.

5. Non-relevant document matches "bad" nugget – An incorrectly collected nugget can be "bad" in a variety of ways. A nugget could be rendered meaningless by stopping and stemming, it could be too long and therefore support partial matches, or it could have been not specific enough. The latter can often be fixed by applying keyword filters. The first two can be eliminated with proper training.

6. Relevant document doesn't match "bad" nugget – This mainly occurred when the nugget as collected actually contained several nuggets. A document might match well only part of a nugget, but this was not enough for our algorithm to consider it a good match. Again, proper training of assessors can address this issue.

7. Relevance cannot be captured by a text string – It is not always possible to capture relevance with just a string. For example, a query might ask for specific images or home pages, in which case image data or URLs might be more appropriate choices for nuggets.

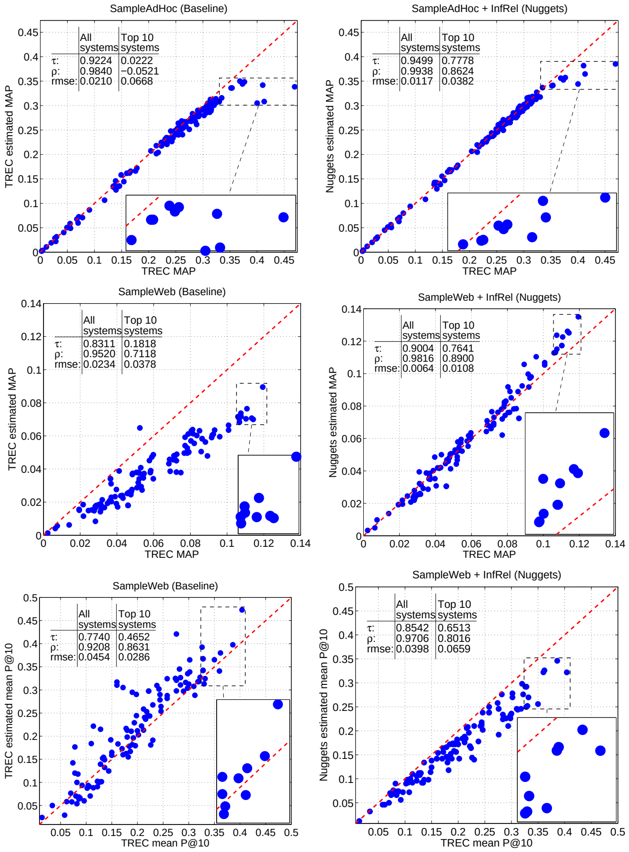


Figure 4: Evaluation comparison. Left: evaluations obtained using only the assessed sample; Right: evaluations obtained using the assessed sample and the inferred documents. Top: evaluations obtained using SampleAdHoc; Middle: evaluations (MAP) obtained using SampleWeb; Bottom: evaluations (P@10) obtained using SampleWeb. Top 10 systems are zoomed in the same plot.

8. Nugget doesn't match due to HTML parsing – The HTML structure of some web pages was complicated enough to foil our nugget matching algorithm.

Our analysis shows that while many of our errors are either unavoidable, e.g. spam, or not really errors, e.g. disagreement, most can be corrected with simple technological fixes such as keyword filters and better HTML parsing, or the proper training of judges.

5. APPLICATION TO EVALUATION

Not withstanding the incompleteness of the published TREC qrels (see Section 4.1), we will treat them as the "ground truth" for the purpose of demonstrating the utility of our test collection methodology to IR system evaluation. We produce two separate qrels based on our sample, one containing the TREC assessments of documents in our sample, denoted "Sample (Baseline)", and the other containing those judgments as well as the judgments inferred using the extracted nuggets, denoted "Sample + InfRel (Nuggets)". Using these qrels, we evaluated all systems submitted to the

TREC 8 adhoc track over all 50 queries, and separately all systems submitted to the TREC 09 web track, also over all 50 queries. The results of this experiment are shown in Figure 4, with each data point representing an IR system. Perfect performance would be indicated by all data points coinciding with the line $y = x$.

While the scatter plots are largely qualitative, we also compute several statistics. Kendall's τ is a measurement of rank agreement. ρ is a linear correlation coefficient, which measures linear agreement, i.e. the goodness of fit to some straight line, and implies rank correlation. We also compute the root mean square error, the difference of our scores compared to the actual scores, which implies both linear and rank correlation.

Our methodology using nuggets and inferred relevance judgments clearly out-performs the baseline evaluation using the relevance judgments of the sample. For the ad hoc experiment, using inferred relevance increases Kendall's τ from 0.92 to 0.95, linear correlation from 0.98 to 0.99, and decreases RMS error from 0.02 to 0.01. For web, MAP evaluation with inferred relevance increases Kendall's τ from 0.83 to 0.90, linear correlation from 0.95 to 0.98, and decreases RMS error from 0.02 to 0.01. Also for web, P@10 evaluation with inferred relevance increases Kendall's τ from 0.77 to 0.85, linear correlation from 0.92 to 0.97, and decreases RMS error from 0.05 to 0.04.

In most circumstances, evaluation accuracy matters most for the top systems. For this reason, it is important to note that the baseline evaluation wildly under-evaluates the top 10 systems. To make this point clear, we zoom in on the top 10 systems in each plot. Also of interest is the fact that, for ad hoc runs, the nugget-based evaluation of the top systems is much better than the baseline evaluation of the same systems. This is significant since the TREC ad hoc assessments (based on depth-100 pooling) are far more complete than the web ones (based on depth-10 pooling): using the SampleAdHoc and the inferred relevant documents and limiting our analysis to the top 10 systems, we obtain a dramatic increase of Kendall's τ from 0.02 to 0.78, linear correlation from -0.05 to 0.87, and a decrease of RMS error from 0.04 to 0.07. MAP evaluations of the top 10 systems using the SampleWeb and the inferred relevant documents shows similar results. However, for P@10 on web runs, the Kendall's τ is not much better than the baseline evaluation. We believe this is due to the following factors: (1) P@10 is more sensitive to judging disagreements than MAP, and many such cases exist in the web qrel, (2) web text matching is more difficult, and (3) in general, the nuggets-based framework is more appropriate for recall-oriented tasks.

Table 5 shows the comparison of the rankings of the top 10 systems in the various systems. Rankings based on nugget-inferred relevance are generally more consistent with the TREC rankings than their baseline counterparts. For ad hoc systems, using inferred relevance reduced the total absolute rank difference for top the 10 systems from 46 to 8. For MAP on web runs, we reduced the total absolute rank difference for top the 10 systems from 32 to 10. For P@10 on web runs, the absolute difference in ranking is actually slightly worse for reasons given above.

While our evaluation results for the top 10 systems are not perfect (a Kendall's τ closer to 0.9 is desirable), even nuggets extracted by untrained judges from a small sample of documents, our matching algorithm produces inferred relevance score that lead to very good evaluations.

5.1 Reusability: Systems Not Part of the Pool

Pretend that a system that contributed to the document pool did not exist when documents were being selected for nugget extraction. Would we still be able to evaluate this system? This is the reusability task. In order to test the reusability of the inferred relevance assessments, we remove the nuggets extracted from several systems, as well as any relevance assessments for documents only returned by this system. We pick these systems greedily based on their high number of unique relevant documents not retrieved by other systems. Together the removed systems contributed 1306 unique relevant documents to the full ad hoc qrel of 4728 relevant documents; the removed systems contributed 72 out of 1701 relevant documents to SampleAdhoc. We call the new sample, with the corresponding documents and nuggets removed, "SampleAdHocReuse". Similarly, 126 out 1260 relevant documents were removed from SampleWeb, producing "SampleWebReuse".

The results of removing these relevant documents and their nuggets from the samples are shown in Figure 5. Note that the baseline evaluation over-evaluates the majority of the systems. These systems were penalized in the original TREC assessment for not retrieving these unique relevant documents now removed from the qrel. The baseline also massively under-evaluates the removed systems (red pluses), because of the unique relevant documents these systems retrieve, which are considered not relevant since they are not assessed. However, the nuggets-based evaluations (right plot) are very stable. This is due to the ability of our method to infer relevance on most missing documents.

6. CONCLUSIONS AND FUTURE WORK

We have described a method for building Test Collections for IR systems on the basis on relevant information, as opposed to the Cranfield paradigm which is based on relevant documents. This "nugget" approach has the potential to solve important problems like scalability, reusability, and applicability, as well as the potential to improve IR measurement technology itself. We showed that starting with a few relevant documents, by carefully collecting relevant facts, a simple matching function can recover the vast majority of assessed relevant documents and a great many other unassessed yet relevant documents. We also showed that these inferred-relevant documents can be successfully used for IR system evaluation. The method also demonstrates that a large number of relevant documents will not be assessed by the Cranfield paradigm.

There are many other important applications of such nugget-based relevance approach, some of which we mention below:

Learning-to-rank. Recently, much effort has been devoted to applying machine learning techniques to creating ranking functions via training on assessed (query, document) pairs. Using the inferred relevance, we can create much, much larger training sets. Even if the inferred relevance is not 100% accurate, larger training sets are likely to improve both quantitative learnine (e.g. regression) and discriminative learning (e.g. boosting or support vector machines).

Post-processing and organization of nuggets. Once we have extracted nuggets, several steps are necessary to organize them. Nuggets can be too long, too short, too focused, or too general. Stop word elimination and stemming

TREC	SampleAdHoc			SampleWeb (MAP)			SampleWeb (P@10)		
Rank (qrel)	System Name	Rank (Baseline)	Rank (Nuggets)	System Name	Rank (Baseline)	Rank (Nuggets)	System Name	Rank (Baseline)	Rank (Nuggets)
1	READWARE2	5(-4)	1(0)	NeuDiv1	1(0)	1(0)	uwgym	1(0)	4(-3)
2	orcl99man	13(-11)	3(-1)	uogTrDYCcsB	10(-8)	3(-1)	uogTrDPCQcdB	3(-1)	1(1)
3	iit99ma1	4(-1)	2(+1)	udelIndDRSP	6(-3)	2(1)	NeuDiv1	5(-2)	2(1)
4	READWARE	16(-12)	7(-3)	uogTrDPCQcdB	2(2)	5(-1)	uogTrDYCcsB	13(-9)	3(1)
5	CL99XTopt	2(+3)	4(+1)	UMHOOsd	9(-4)	7(-2)	MSDiv3	8(-3)	15(-10)
6	CL99XT	3(+3)	6(0)	UMHOOsdp	8(-2)	6(0)	MSRACS	19(-13)	13(-7)
7	CL99SDopt1	1(+6)	5(+2)	NeuLMWeb600	4(3)	8(-1)	MSRAACSF	26(-19)	16(-9)
8	CL99SD	6(+2)	8(0)	NeuDivW75	3(5)	4(4)	UCDSIFTslide	6(2)	6(2)
9	CL99SDopt2	7(+2)	9(0)	udelIndDMRM	11(-2)	9(0)	UCDSIFTdiv	7(2)	8(1)
10	8manexT3D1N	8(+2)	10(0)	NeuLMWeb300	7(3)	10(0)	MSDiv2	11(-1)	19(-9)
	Total absolute difference	(46)	(8)		(32)	(10)		(42)	(44)

Table 5: TREC top 10 systems ranking (ranking difference in parenthesis)

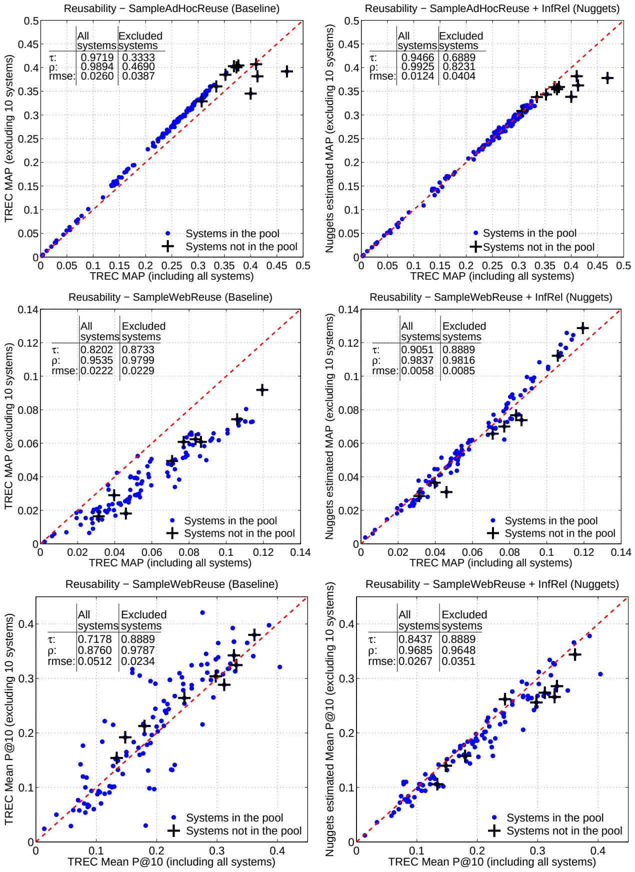


Figure 5: Reusability comparison; "+" denotes systems not contributing to selection of documents. Left: evaluations obtained using only the assessed sample - self-stability of TREC evaluations; Right: evaluations obtained using the assessed sample and the inferred documents - stability of Nuggets evaluations with respect to TREC; Top: evaluations obtained using SampleAdHocReuse; Middle: evaluations (MAP) obtained using SampleWebReuse; Bottom: evaluations (P@10) obtained using SampleWebReuse.

might cause some nuggets to be unusable. Some nuggets might require additional keywords, and so on. A significant step is *clustering* of the nuggets, necessary for at least two reasons: (1) Matching functions that check a document against all nuggets extracted may use an score accumulator, or in the case of probabilistic matching, a product of probabilities; such methods usually make a natural assumption of independence between nuggets, which is not always realistic since many nuggets might contain the same information. Clustering will solve this by creating independent subsets of nuggets. (2) Diversity tasks rely on detecting common information between documents; clustering of the nuggets will allow us to do just that, by indicating that different nuggets matched against two documents belong to the same cluster. We generally think of clustering as an automatic procedure, but some supervision might be necessary.

Nuggets may not be all equal (although they all represent relevant information). Each nugget may have a quality or "importance", which the similarity or matching functions can leverage; such a weighting scheme may be used to model what in Information Retrieval is often termed "graded relevance": document quality that can range from "entirely irrelevant" to "perfectly relevant".

Performance measure. Most current performance measures assume that the document is an atomic unit: it is either relevant or non-relevant (or relevant to some degree), and it is effectively assumed to take a fixed constant effort to read and understand. This is, of course, incorrect in practice: short documents that contain large fractions of relevant information are far superior to long documents containing relatively small fractions of relevant information, though both may equally be assessed "relevant". Given relevant information encoded as nuggets, we could potentially assess the fraction of a document that is relevant and the fraction of the relevant information that it contains (information precision and information recall), thus obtaining an overall measure of performance much more closely matching user utility. We propose to develop and test just such measures.

Summarization and canonical document evaluation. Finally, one can envision entirely new evaluation tasks and methodologies using the techniques that underly the nugget-based evaluation proposed above. For example, how could one evaluate the quality of the canonical Wikipedia page on Barack Obama or the output of a "knowledge engine" such as Wolfram Alpha? Given the information relevant to a query, as encoded by nuggets, one could potentially assess the fraction of relevant information found in the output (information recall) and the fraction of information in the output that is relevant (information precision). This would move us many steps toward *information* retrieval evaluation, as opposed to *document* retrieval evaluation.

7. REFERENCES

[1] *33rd ACM SIGIR Workshop on Crowdsourcing for Search Evaluation*, Geneva, Switzerland, 2010.

[2] Azin Ashkan and Charles L.A. Clarke. On the informativeness of cascade and intent-aware effectiveness measures. In *Proceedings of the 20th Annual Internatual World Wide Web Conference*, pages 407–416, 2011.

[3] Javed A. Aslam and Emine Yilmaz. Inferring document relevance via average precision. In Susan Dumais, Efthimis N. Efthimiadis, David Hawking, and Kalervo Jarvelin, editors, *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 601–602. ACM Press, August 2006.

[4] Peter Bailey, Nick Craswell, Ian Soboroff, Paul Thomas, Arjen P. de Vries, and Emine Yilmaz. Relevance assessment: Are judges exchangeable and does it matter? In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 667–674, New York, NY, USA, 2008. ACM.

[5] Andrei Z. Broder. Identifying and filtering near-duplicate documents. In *Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching*, COM '00, pages 1–10, London, UK, 2000. Springer-Verlag.

[6] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the web. In *Selected papers from the sixth international conference on World Wide Web*, pages 1157–1166, Essex, UK, 1997. Elsevier Science Publishers Ltd.

[7] Ben Carterette, James Allan, and Ramesh K. Sitaraman. Minimal test collections for retrieval evaluation. In *Proceedings of SIGIR*, pages 268–275, 2006.

[8] Ben Carterette, Virgil Pavlu, Evangelos Kanoulas, Javed A. Aslam, and James Allan. Evaluation over thousands of queries. In Sung-Hyon Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat-Seng Chua, and Mun-Kew Leong, editors, *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 651–658. ACM Press, July 2008.

[9] Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of SIGIR*, pages 659–666, 2008.

[10] Hoa Trang Dang, Jimmy J. Lin, and Diane Kelly. Overview of the TREC 2006 question answering track 99. In *TREC*, 2006.

[11] Charles L. A. Clarke Gordon V. Cormack, Mark D. Smucker. Efficient and effective spam filtering and re-ranking for large web datasets. University of Waterloo, 2010.

[12] Donna Harman. Overview of the third text REtreival conference (TREC-3). In D.K. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 1–19. U.S. Government Printing Office, April 1995.

[13] Jimmy Lin and Dina Demner-Fushman. Automatically evaluating answers to definition questions. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 931–938, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[14] Jimmy Lin and Dina Demner-Fushman. Will pyramids built of nuggets topple over? In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 383–390, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[15] Gregory Marton and Alexey Radul. Nuggeteer: Automatic nugget-based evaluation using descriptions and judgements. In *Proceedings of NAACL/HLT*, 2006.

[16] Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality? In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 43–52, New York, NY, USA, 2008. ACM.

[17] Ellen M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *Inf. Process. Manage.*, 36:697–716, September 2000.

[18] Ellen M. Voorhees. Question answering in TREC. In *Proceedings of the tenth international conference on Information and knowledge management*, CIKM '01, pages 535–537, New York, NY, USA, 2001. ACM.

[19] Yiming Yang and Abhimanyu Lad. Modeling expected utility of multi-session information distillation. In *Proceedings of the 2nd Annual International Conference on the Theory of Information Retrieval*, 2009.

[20] Yiming Yang, Abhimanyu Lad, Ni Lao, Abhay Harpale, Bryan Kisiel, and Monica Rogati. Utility-based information distillation over termporally sequenced documents. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 31–38, 2007.

[21] Emine Yilmaz and Javed A. Aslam. Estimating average precision when judgments are incomplete. *Knowledge and Information Systems*, 16(2):173–211, August 2008.

[22] Justin Zobel. How reliable are the results of large-scale retrieval experiments? In W. Bruce Croft, Alistair Moffat, C. J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 307–314, August 1998.