IR System Evaluation using Nugget-based Test Collections

Virgil Pavlu Shahzad Rajput Peter B. Golbus Javed A. Aslam
College of Computer and Information Science, Northeastern University
{vip, rajput, pgolbus, jaa}@ccs.neu.edu*

ABSTRACT

The development of information retrieval systems such as search engines relies on good test collections, including assessments of retrieved content. 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 "nuggets" are collected, our matching method [23] can assess any document for relevance with high accuracy, and so any retrieved list of documents can be assessed for performance. In this paper we analyze the performance of the matching function by looking at specific cases and by comparing with other methods. We then show how these inferred relevance assessments can be used to perform IR system evaluation, and we discuss in particular reusability and scalability. 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.

Categories and Subject Descriptors

H.3 [**Information Systems**]: Information Storage and Retrieval; H.3.4 [**Systems and Software**]: Performance Evaluation

General Terms

Measurement, Performance

1. INTRODUCTION

Much thought and research has been devoted to each of the steps of evaluation of IR systems: constructing a test collection of documents and queries, judging the relevance of the documents to each query, and assessing the quality of the ranked lists. The annual text retrieval conference TREC [17] is central to the standardization of these tasks.

For large collections of documents and/or topics, it is impractical to assess the relevance of each document to each topic. Instead, only a small subset of the documents 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. Shallower and deeper pools have been studied [32, 17] 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: In Phase 1, "Collection Construction", documents, topics, and relevance assessments are all gathered. Following Phase 1, we have a *test collection* that can be used to evaluate the performance of systems in Phase 2, "Evaluation". 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

*We gratefully acknowledge the support provided by the NSF grant IIS-1017903.

WSDM'12, February 8–12, 2012, Seattle, Washington, USA.
Copyright 2012 ACM 978-1-4503-0747-5/12/02 ...$10.00.

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 associated "subtopics" [13], and the information retrieved by a system is encoded by documents and either their "marginal utility" with respect to previously retrieved documents or some measure of the coverage of those documents over the associated subtopics.

The central issue with the Cranfield paradigm and with its variants described above is that the information relevant 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. Hundred of thousands of documents were 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 233 million results on Google as of August 2011, 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 "address" the scalability and reusability issues described above using current methodologies. This issue is discussed below.

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". 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. 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 evaluate retrieval systems.

1.2 Related Work

Various attempts to address the issues described above have been proposed. (1) Sampling techniques such as infAP [31], statAP [12], 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 233 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 [11] 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 [2]. 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 [22], but this is only applicable to the web and only for those queries and documents with sufficient "clicks". Evaluating IR systems without relevance assessments has also been the subject of research [25, 26, 24]; however, these methods tend to lack the capability of inferring the relevance of arbitrary documents outside the pool, limiting reusability. Quite similar to our work, [4] proposes using "Trels", sets of relevant and non-relevant keywords, to assess arbitrary documents. We believe nuggets can capture more information than simple keywords.

Finally, we note that nuggets of a somewhat different kind are widely used in other contexts. For example, the evaluation of question answering systems [19, 20, 15, 28, 21] 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 [30, 29]. Conceptual nuggets are currently used in novelty and diversity evaluation: subtopics can be thought of as nuggets [14], or systems can be evaluated on coverage of both subtopics and nuggets [6]. However, in none of these contexts are nuggets used to infer relevance automatically. In an effort to improve retrieval under resource restricted conditions (e.g. small screen mobile devices) INEX focuses on retrieving relevant elements/passages from XML documents. However, properly evaluating the effectiveness in XML-IR remains an ongoing research question at INEX [5]. Additionally, the collection used is a set of documents from Wikipedia, which is much cleaner and more structured than arbitrary HTML document on the web. All of the above are still susceptible to the limitations of the Cranfield paradigm.

To address the limitations of the Cranfield paradigm and to achieve test collection scalability, reusability, and applicability, we propose a test collection construction methodology based on nuggets. In previous work [23], we describe a methodology for collecting nuggets, a technique for assessing the relevance of documents given these nuggets, and a preliminary analysis of the quality of these relevance assessments. These results are summarized in Section 2 and parts of Section 3 below. In this work, we provide a detailed exposition of our methodology, a thorough analysis of the quality of our inferred relevance assessments, and a demonstration that these relevance assessments and the test collection thus inferred addresses the issues of scalability, reusability, and applicability for information retrieval evaluation.

2. METHODOLOGY

Consider for example the web query "Barack Obama". The vast majority of the potentially 233 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.

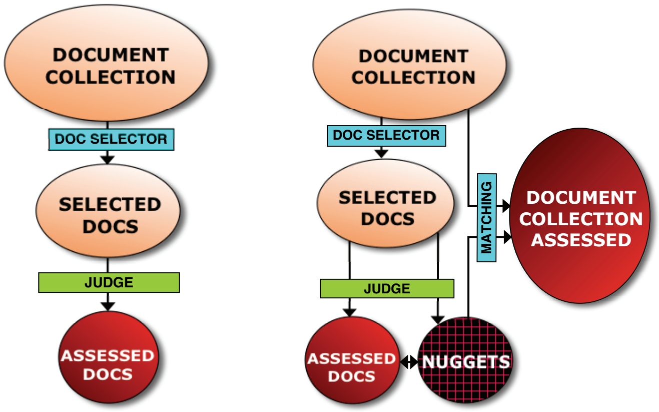

Figure 1: For a given query, selected documents are evaluated as "relevant/nonrelevant". Left: Traditional TREC strategy for relevance. Right: Proposed nuggets method.

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. Once documents are automatically assessed, all standard measures of retrieval performance (such as Average Precision) can be computed for any ranked list. In Section 4 we demonstrate that runs submitted to TREC can be effectively evaluated using collected nuggets.

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 previously 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.

2.1 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 particular methodology, previously summarized in [23].

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/consensus 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 [12].

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 assessors were asked to extract the relevant nuggets. They were instructed to find the smallest portion of text that constitutes relevant information in and of 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 information encoded in the form of verbatim text, e.g. "A healthy diet with enough calcium helps reduce high risk of osteoporosis".

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, for the topic "JFK assassination", an assessor might add the keyword "`Kennedy`": if a document does not contain this term, it is promptly not relevant.

Inferred Relevance Judgements. According to the typical TREC definition of relevance for ad hoc retrieval, a document is considered relevant if it contains at least one 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 more sophisticated 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 [9, 10]. 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 used the algorithm presented in [18] to find the shortest span of shingle words in a text document in linear time. We define the shingle score as follows

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

  where λ is a fixed decay parameter. We found $\lambda = 0.95$ to be an effective value. 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.

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

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

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

Sample	Docs/Relev	Assess time	Nuggets	Extract time
AdHoc	200/34	3.3 hours	86.98	1.7 hours
Web	200/25.18	3.2 hours	61.82	1.3 hours

Table 1: Sample statistics (query average)

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`".

- **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 $\theta = 0.8$, then the document is inferred to be relevant. This threshold is optimized empirically, but it is constrained to be a constant across all experiments; better performance can be obtained by setting the threshold differently for each query or experiment. Such variable thresholds could be learned from data.

3. EXPERIMENTS

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 of 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 [1]: 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 [13], 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).

The notion of correctness of relevance judgments is somewhat problematic. Inter-assessor disagreement [8, 27] is a well known phenomenon — the question of relevance is ambiguous for many documents. Bearing this in mind, we next analyze the quality of our inferred relevance judgments in several ways: (1) sort documents by their document relevance scores and measure Mean Average Precision (MAP) of the induced retrieval performance, (2) threshold relevance scores to create a qrel file and compare directly with TREC qrel file in terms of precision, recall, F1; (3) human verification of documents inferred relevant but not judged by TREC, and (4) use our inferred qrel to evaluate retrieval systems and compare system ranking with published TREC rankings in terms of Kendall's τ.

3.1 Performance: Finding Relevant Documents

Treating the relevances inferences as a retrieval function, and restricting to only those documents judged by TREC, we compute an average precision of AP=0.75 (Table 2) or better for our ranked list, which implies the vast majority of relevant documents are ranked higher than non-relevant ones. To our knowledge MAP=0.75 compares very favorably to any previous adhoc/web retrieval method, including machine learning and relevance feedback mechanisms (see 3.2). Some of these results we stated in a previous work [23].

Next, we use a threshold derived by trial and error 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.

After thresholding ($\theta = 0.8$), 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 [7] achieves an F1 of 0.68 (compared with our F1=0.75), and uses significant extra ranking information which is highly contextual and may not be available.

	Judged Rel	Judged NonRel	Total	Agreement
AdHoc	2464	337	2801	87.96%
Web	2969	411	3380	87.84%

Truncated Result List	MAP	Precision	Recall	F1
SampleAdHoc	n/a	0.18	0.47	0.26
SampleAdHoc+InfRel	0.76	0.88	0.65	0.75
SampleWeb	n/a	0.23	0.25	0.24
SampleWeb+InfRel	0.75	0.88	0.60	0.71

Table 2: Documents inferred relevant from TREC qrel, compared to TREC assessments.

3.2 Comparison with Learning and Relevance Feedback Methods

In Table 3, we show comparison with other approaches that also use manual intervention. The training or relevance feedback set used by these methods is the same as that used by the Nuggets method. For reference we include basic results without relevance feedback (BM25, Language Model with Dirichlet smoothing). The learning algorithms (Linear regression, SVM, and RankBoost) use for each query the sampled documents as training; note that this is training per query, as opposed to the traditional learning-to-rank setup where different sets of queries are used for training and testing. RankBoost achieves the best result of the comparison methods (MAP=0.484 for ad hoc, MAP=0.661 for web) but still significantly lower than our Nuggets method.

Method	Ad Hoc	Web
Boolean Retrieval (Keyword Filtering)	0.111	0.219
BM25	0.204	0.293
BM25 + Keyword Filtering	0.198	0.292
BM25 + Relevance Feedback	0.387	0.428
LM Dirichlet	0.152	0.266
LM Dirichlet + Keyword Filtering	0.151	0.266
LM Dirichlet + Relevance Feedback	0.272	0.418
TFIDF	0.204	0.293
TFIDF + Keyword Filtering	0.199	0.292
TFIDF + Relevance Feedback	0.390	0.428
Learning(per query) Regression	0.311	0.466
Learning(per query) SVM+RBF kernel	0.476	0.632
Learning(per query) RankBoost	0.484	0.661
Nuggets Method	0.746	0.754
Nuggets Method + Keyword Filtering	**0.755**	**0.755**

Table 3: MAP Comparison with Learning Methods.

3.3 Failure Analysis

	Sources of Errors	Ad Hoc(%)	Web(%)
1	Assessor disagreement	46 (–)	71 (–)
2	Spam filter [16]	N/A	45 (–)
3	Missing information	64 (50.3%)	47 (55.9%)
4	Textual ambiguity	12 (9.5%)	2 (2.4%)
5	Overly general nuggets	39 (30.7%)	4 (4.7%)
6	Non-atomic nuggets	7 (5.5%)	14 (16.7%)
7	Text string	5 (3.9%)	11 (13.1%)
8	HTML parsing	N/A	6 (7.1%)

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

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 significant disparities in terms of document score and TREC relevance assessment. We describe the main eight reasons below:

1. Assessor disagreement: 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 [16]. 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 information: There may be relevant information not present in the collected nuggets. Documents that only discuss this unrepresented information will not be found. This can be addressed by extracting nuggets from more documents.
4. Textual ambiguity: Either the match is not exact due to limitations of our particular matching function (for example not recognizing synonyms or coreferences) or the text is inherently ambiguous. Solution: improved NLP into the matching function.
5. Overly general nuggets: A nugget could be rendered meaningless by stopping and stemming, or it could be so vague as to apply to many topics. This can be eliminated with proper assessor training and keyword filtering.
6. Non-atomic nuggets: Some nuggets as collected actually contained several nuggets. A document might match some part of these nuggets, 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 a more appropriate choices for nuggets.
8. Nugget does not 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 our largest cause of error is missing information, which can be addressed by using a more diverse document sample. Some of our errors are perhaps unavoidable, e.g. spam, or not really errors, e.g. disagreement. Many of the remaining errors can be corrected simply with technical fixes such as keyword filters and better HTML parsing, or the proper training of assessors.

3.4 Effort vs. Nuggets vs. Information

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 nuggets are extracted from non-relevant documents.

TREC assessors judge between 50 and 100 documents an hour.[1] For the sake of comparison, 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.

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.

In Figure 2, the solid blue curve shows the rate of finding nuggets per unit of human effort (including assessment for relevance and nugget extraction from relevant documents). Since the order of processing documents reflected on the x-

[1] Private communication with TREC organizer.

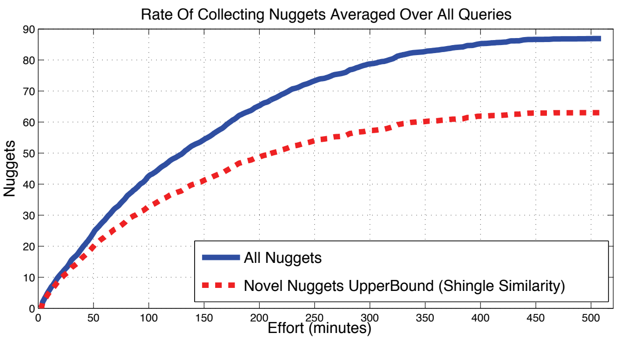

Figure 2: Rate of collecting (all and novel) nuggets averaged over all queries compared with the effort spent over the documents in the SampleAdHoc.

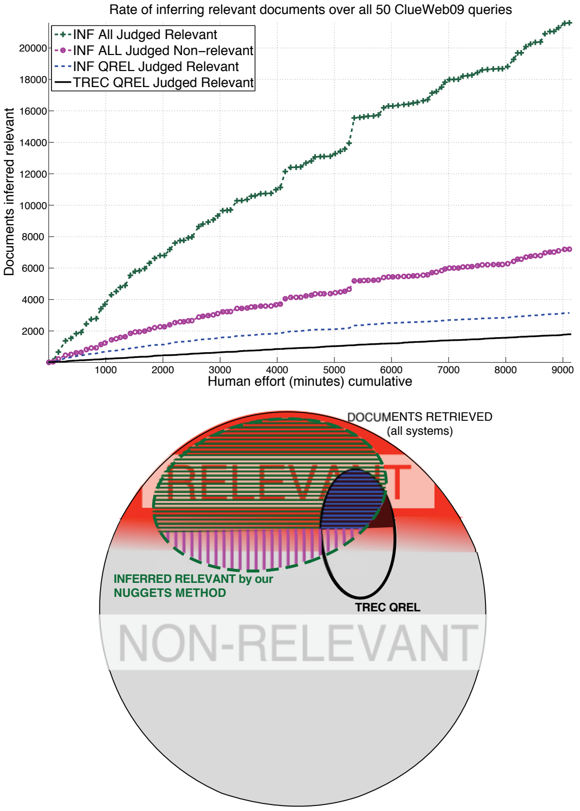

Figure 3: The top plot shows the rate of finding relevant documents per unit of time: TREC qrel (black), nuggets inferred from TREC qrel (blue), nuggets inferred–assessor disagreement (pink) and nugget inferred–assessor agreement (green). The result of this process is shown in the bottom diagram, which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

axis is "most likely relevant first", the curve is initially steep for the first few hours of effort and then it starts to flatten, ultimately asymptoting at approximatley 90 nuggets on average per query. This in part validates our hypothesis that while there may be vast numbers of relevant documents, there are likely far fewer relevant pieces of information. In fact, there are likely even fewer relevant pieces of information than indicated by the solid blue curve in Figure 2, since multiple nuggets may encode the same underlying information. To assess this phenomenon, we employed shingle matching on the extracted nuggets with a conservative threshold of $\theta = 0.3$ that yields virtually no false positive similarity mistakes, as validated on a separate entailment data set [3]. The resulting dashed red curve is an *upper bound* on the number of novel (semantically distinct) nuggets found, asymptoting at just over 60 novel nuggets per query on average. (In further analyzing the nuggets manually, we believe that the true value is closer to 40.) In short, this plot confirms our thesis that after relatively few documents selected well (or at least not selected badly), an assessor sees most of relevant information.

3.5 Many More Relevant Documents

To test the hypothesis that our method finds many additional relevant documents, as well as to further test the correctness of our inferred relevance judgments, we also used the nuggets extracted from SampleWeb to infer the relevance of documents retrieved within the top 300 ranks by any web system (depth-300 pool) — about 5891 documents per query. About 460 documents were marked relevant per query; on average 400 of these were unjudged by TREC.

Validation of the inferred relevance assessments outside the TREC qrel was performed by taking a uniform random sample of about 80 out of the 400 inferred-relevant, TREC-unjudged, documents per query, and having them assessed for relevance by humans using Amazon's Mechanical Turk service.[2] (Note that this test is not part of the nugget-based methodology, but is instead intended as a validation of our hypothesis and implementation.) 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 unjudged documents (outside TREC qrel).

	Judged Rel*	Judged NonRel*	Total	Agreement
Web	14624	5329	19953	73.29%

(* denotes the extrapolation from about 4000 documents assessed by Turks to all 19953 documents inferred relevant.)

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 14,049, or about 281 per query (maximum likelihood estimate is 14,624, or 292 per query, on average).

Overall, using 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). Figure 3 shows of grand overview of the documents inferred relevant vs. the ones marked relevant by TREC (bottom diagram), for the Web collection. Note

[2] 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.

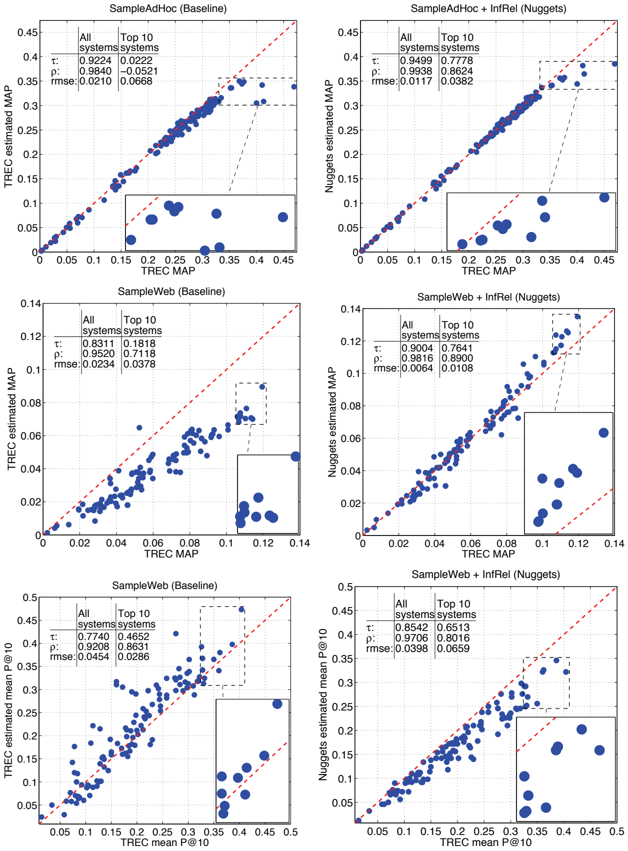

Figure 4: Evaluation comparison. Left: evaluations obtained using only the assessed sample; Right: evaluations obtained using the assessed sample and the inferred documents. Top: evaluations obtained using SampleAdHoc; Middle: evaluations (MAP) obtained using SampleWeb; Bottom: evaluations (P@10) obtained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

also the vast number of documents inferred relevant outside the TREC qrel (green). On the top plot, the number of documents inferred total (green) and incorrectly (pink) is compared with the TREC qrel size (black) per unit of human effort of assessment/extraction.

4. IR SYSTEM EVALUATION

Notwithstanding the incompleteness of the published TREC qrels (see previous section), 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. The linear correlation coefficient ρ measures linear agreement, i.e. the goodness of fit to some straight line, which 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.

While the baseline evaluation using the relevance judgments of the sample already performs very well, our methodology using nuggets and inferred relevance judgments shows definite improvement. 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 significantly 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 0.07 to 0.04. 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 shows the most dramatic gains for recall-oriented tasks and metrics. 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 36 to 8. For MAP on web runs, we reduced the total absolute rank difference for top the 10 systems from 28 to 10. For P@10 on web runs, we reduced the total absolute rank difference for top the 10 systems from 22 to 18.

4.1 Reusability: Systems Not Part of the Pool

The reusability of a test collection is its ability to accurately assess the performance of *unseen* systems that did not contribute to its own construction. Reusability can be estimated by holding out a set of *seen* systems—eliminating their contribution to the test collection—and assessing the ability of the modified test collection to accurately assess both the held-out systems and the systems that yet remain.

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	9(-7)	3(-1)	uogTrDYCcsB	9(-7)	3(-1)	uogTrDPCQcdB	2(0)	1(1)
3	iit99ma1	4(-1)	2(+1)	udelIndDRSP	5(-2)	2(1)	NeuDiv1	3(0)	2(1)
4	READWARE	10(-6)	7(-3)	uogTrDPCQcdB	2(2)	5(-1)	uogTrDYCcsB	8(-4)	3(1)
5	CL99XTopt	2(+3)	4(+1)	UMHOOsd	8(-3)	7(-2)	MSDiv3	6(-1)	8(-3)
6	CL99XT	3(+3)	6(0)	UMHOOsdp	7(-1)	6(0)	MSRACS	9(-3)	7(-1)
7	CL99SDopt1	1(+6)	5(+2)	NeuLMWeb600	4(3)	8(-1)	MSRAACSF	10(-3)	9(-2)
8	CL99SD	6(+2)	8(0)	NeuDivW75	3(5)	4(4)	UCDSIFTslide	4(4)	5(3)
9	CL99SDopt2	7(+2)	9(0)	udelIndDMRM	10(-1)	9(0)	UCDSIFTdiv	5(4)	6(3)
10	8manexT3D1N	8(+2)	10(0)	NeuLMWeb300	6(4)	10(0)	MSDiv2	7(3)	10(0)
	Total absolute difference	**(36)**	**(8)**		**(28)**	**(10)**		**(22)**	**(18)**

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

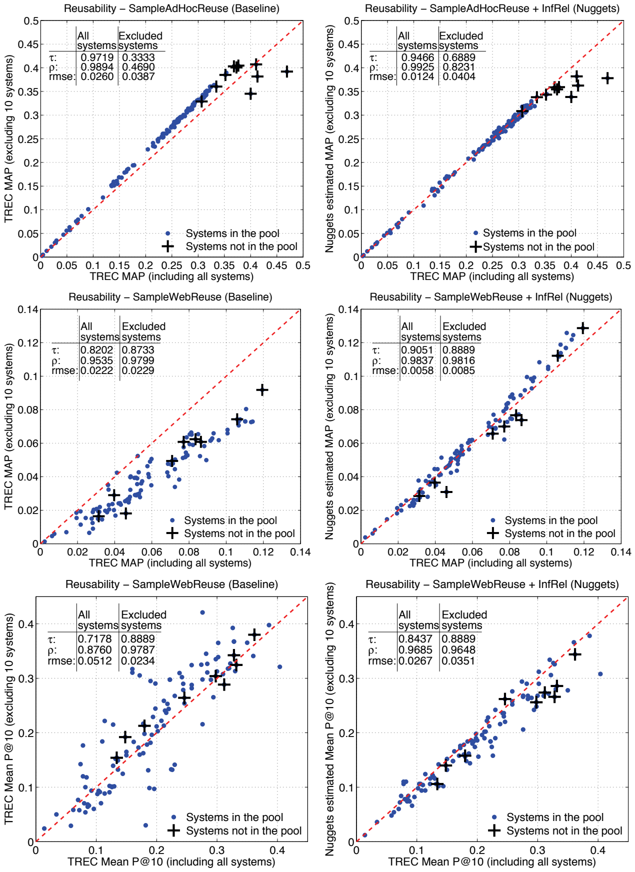

Figure 5: Reusability comparison; "+" denotes systems not contributing to selection of documents. Left: evaluations obtained using only the assessed sample - self-stability of TREC evaluations; Right: evaluations obtained using the assessed sample and the inferred documents - stability of Nuggets evaluations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

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 these systems. We pick 10 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. In order to make an analysis, we need to compare the reusability plots with corresponding evaluation plot. The effect can be seen for systems having higher MAP values (higher than 0.15). 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 significantly under-evaluates the top removed systems (black 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. P@10 on web runs shows similar effects.

5. CONCLUSIONS AND FUTURE WORK

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. While the documents retrieved by our system are necessarily similar to those in our sample, our method still demonstrates that a large number of relevant documents will not be assessed by the Cranfield paradigm.

Pool Bias. All test collection methodologies, including ours, are biased towards documents rich in query terms. We believe that nuggets-based test collections will in the future be able to find many more non-obvious documents.

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 learning (e.g. regression) and discriminative learning (e.g. boosting or support vector machines).

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.

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).

6. REFERENCES

[1] *The Eighth Text REtrieval Conference (TREC-8)*. U.S. Government Printing Office, 2000.

[2] *33rd ACM SIGIR Workshop on Crowdsourcing for Search Evaluation*, Geneva, Switzerland, 2010.

[3] P. Achananuparp, X. Hu, and X. Shen. The evaluation of sentence similarity measures. DaWaK '08, Berlin, Heidelberg, 2008.

[4] E. Amitay, D. Carmel, R. Lempel, and A. Soffer. Scaling IR-system evaluation using term relevance sets. SIGIR '04, New York, NY, USA, 2004.

[5] P. Arvola, S. Geva, J. Kamps, R. Schenkel, A. Trotman, and J. Vainio. Overview of the INEX 2010 ad hoc track. In *Preproceedings of the INEX 2010 Workshop*, Vught, The Netherlands, 2010.

[6] A. Ashkan and C. L. Clarke. On the informativeness of cascade and intent-aware effectiveness measures. WWW '11, 2011.

[7] J. A. Aslam and E. Yilmaz. Inferring document relevance via average precision. SIGIR '06, 2006.

[8] P. Bailey, N. Craswell, I. Soboroff, P. Thomas, A. P. de Vries, and E. Yilmaz. Relevance assessment: Are judges exchangeable and does it matter? SIGIR '08, New York, NY, USA, 2008.

[9] A. Z. Broder. Identifying and filtering near-duplicate documents. COM '00, London, UK, 2000.

[10] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. WWW'97.

[11] B. Carterette, J. Allan, and R. K. Sitaraman. Minimal test collections for retrieval evaluation. SIGIR'06.

[12] B. Carterette, V. Pavlu, E. Kanoulas, J. A. Aslam, and J. Allan. Evaluation over thousands of queries. SIGIR'08, 2008.

[13] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2009 web track, 2009.

[14] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. SIGIR'08, 2008.

[15] H. T. Dang, J. J. Lin, and D. Kelly. Overview of the TREC 2006 question answering track 99. In *TREC*, 2006.

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

[17] D. Harman. Overview of the third text REtreival conference (TREC-3). In *Overview of the Third Text REtrieval Conference (TREC-3)*. U.S. Government Printing Office, Apr. 1995.

[18] S. Krenzel. Finding blurbs. Website. http://www.stevekrenzel.com/articles/blurbs.

[19] J. Lin and D. Demner-Fushman. Automatically evaluating answers to definition questions. HLT '05, Morristown, NJ, USA, 2005.

[20] J. Lin and D. Demner-Fushman. Will pyramids built of nuggets topple over? HLT'06, Morristown, NJ, USA, 2006.

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

[22] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? CIKM '08, New York, NY, USA, 2008.

[23] S. Rajput, V. Pavlu, P. B. Golbus, and J. A. Aslam. A nugget-based test collection construction paradigm. CIKM '11, New York, NY, USA, 2011.

[24] T. Sakai and C.-Y. Lin. Ranking Retrieval Systems without Relevance Assessments — Revisited. In *the 3rd International Workshop on Evaluating Information Access (EVIA) — A Satellite Workshop of NTCIR-8*, Tokyo, Japan, 2010.

[25] I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. SIGIR '01, New York, NY, USA, 2001.

[26] A. Spoerri. Using the structure of overlap between search results to rank retrieval systems without relevance judgments. *Inf. Process. Manage.*, 43, 2007.

[27] E. M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *Inf. Process. Manage.*, 36, 2000.

[28] E. M. Voorhees. Question answering in TREC. CIKM '01, New York, NY, USA, 2001.

[29] Y. Yang and A. Lad. Modeling expected utility of multi-session information distillation. ITCIR'09, 2009.

[30] Y. Yang, A. Lad, N. Lao, A. Harpale, B. Kisiel, and M. Rogati. Utility-based information distillation over termporally sequenced documents. SIGIR'07, 2007.

[31] E. Yilmaz and J. A. Aslam. Estimating average precision when judgments are incomplete. *Knowledge and Information Systems*, 16(2), 2008.

[32] J. Zobel. How reliable are the results of large-scale retrieval experiments? SIGIR'98, Aug. 1998.