A Nugget-based Test Collection Construction Paradigm

Shahzad 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, pgolbus, jaa}@ccs.neu.edu

ABSTRACT

The problem of building test collections is central to the development of information retrieval systems such as search engines. Starting with a few relevant "nuggets" of information manually extracted from existing TREC corpora, we implement and test a methodology 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. Our methodology produces highly accurate test collections that hold the promise of addressing the issues of scalability, reusability, and applicability.

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

The key limitation of the Cranfield paradigm is that *the information relevant to a topic is encoded by documents.* Thus, assessing the performance of an information retrieval system requires effectively complete relevance judgments.

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. A number of IR systems are used to find documents believed to be relevant to the topics. A "pool" of these documents is selected to be judged; only the relevance of these documents is assessed.

This gives rise to the following problems:

1. **Scalability:** The Cranfield paradigm and its variants cannot scale to large collections and/or topic sets. For example, the query "Barack Obama" yields about 233 million results on Google as of August 2011; it would be impossible to judge any meaningful sample of these documents at a reasonable cost or in a reasonable amount of time.
2. **Reusability:** For static collections, novel systems will retrieve unjudged but relevant documents, and the assessments of these systems will be inaccurate. 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.
3. **Applicability:** It can be difficult to apply a test collection designed for one retrieval task to another retrieval task. The current methodologies used to address the issues of scale and reusability can exacerbate this problem.

Various attempts to address the issues described above have been proposed. Sampling techniques such as statAP [6], the Minimal Test Collection methodology [5] and their variants have been used extensively in various TREC tracks. 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 20 times the size of standard fully-judged pools. However, this is only a stop-gap measure. These methods cannot scale to collections the size of the web and do not address the issue of dynamic collections. Furthermore, they reduce applicability in that the samples drawn and estimates obtained are typically tailored to specific evaluation measures, *e.g.* average precision.

Crowd-sourcing relevance judgments, via, for example, Mechanical Turk, has also been proposed [1]. However, this too is only a stop-gap measure, allowing us to use collections larger 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. The use of click-through data has also been proposed [8], but this is only applicable to the web, and only for those topics and documents with sufficient "clicks".

In order to address the inherent limitations of the Cranfield paradigm, we propose a test collection construction

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

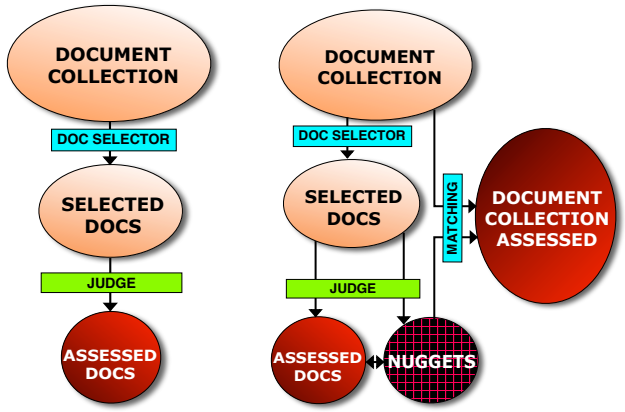

Figure 1: For a given query, selected documents are evaluated as "relevant/nonrelevant". Left: Traditional TREC strategy for relevance. Right: Proposed nuggets methodology.

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 mark the information that is relevant contained within a document, rather than mark the documents containing relevant information. This relevant *information* is used to automatically assign relevance judgments to documents.

2. METHODOLOGY

The Cranfield paradigm and its variants encode the information relevant to a topic by *documents*. Our hypothesis 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. Thus, collecting and encoding the relatively small set of 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.

Figure 1 graphically illustrates the differences between the traditional Cranfield-style 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. These nuggets, the information relevant to a topic, are used to automatically infer the relevance of arbitrary documents in the collection.

3. PILOT IMPLEMENTATION

Building a test collection in our framework consists of three distinct tasks (the rectangular boxes in Figure 1, right): (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 procedure to select the documents to be judged. Generally, documents retrieved by many systems and/or at higher ranks are preferred to documents retrieved by fewer systems and/or at lower ranks. We used the statAP selection mechanism because it has been shown to be an effective document selection procedure in previous TREC ad hoc tracks for system evaluation [6].

Nugget Extraction. Nugget extraction was performed by our internal assessors (primarily graduate students engaged in 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 piece of text that constitutes relevant information in and of itself. However, assessors were 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 collected were relevant information extracts encoded in the form of actual text contained in relevant documents.

Assessors were also given the option of adding topic keywords, which would be used later as a retrieval filter. If a topic has keywords associated with it, a document must contain at least one keyword to be considered relevant for that topic. 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 relevant.

Inferred Relevance Judgements. To test our methodology, we implemented a text-based matching algorithm that automatically infers the relevance of documents given the nuggets extracted. 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 piece of text sufficiently similar to a known relevant information nugget, then the document must necessarily be relevant. Therefore, each document was given a relevance score based on whether it matched any nugget.

The matching algorithm is based on a variant of *shingle matching*, which is often used in near-duplicate detection [4]. 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 [7] to find

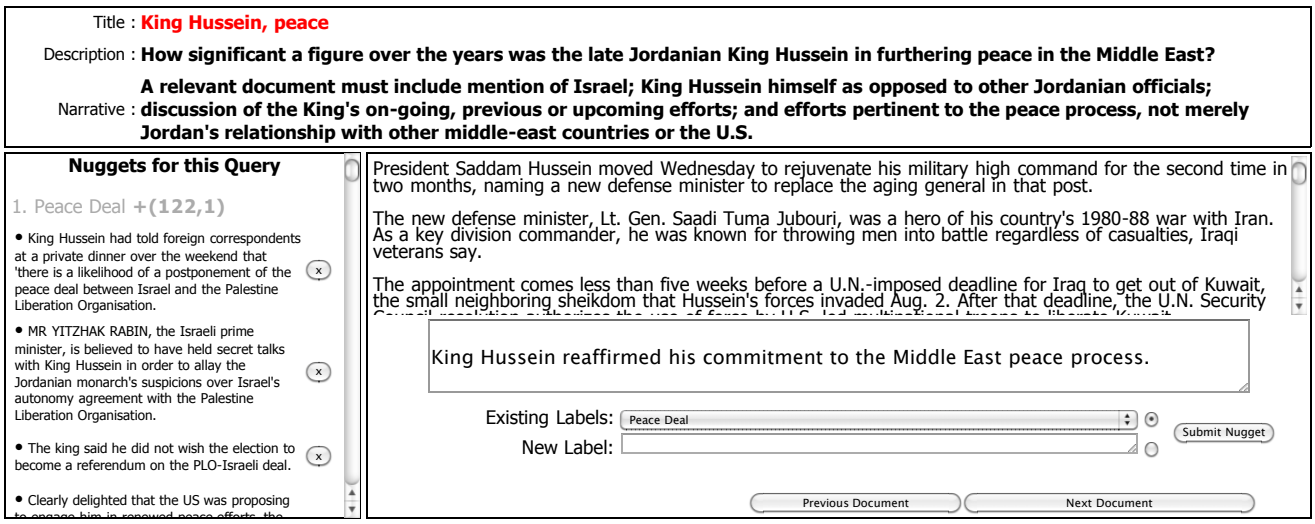

Figure 2: Nugget extraction interface.

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. Note that in contrast to standard shingle matching, we do not require all shingle words to be present in the matching document in the same order or contiguously; by our definition, such a shingle would have a "perfect" shingle score of 1.

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

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

- **Document score:** Since we wish to know if the document matched at least one nugget, the relevance score of each document is defined to be the maximum nugget score of the document:

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

- **Inferred relevance judgment:** We convert a document relevance score to an inferred relevance judgment by simple thresholding. The threshold $\theta = 0.8$ is held constant across all experiments.

4. VALIDATION

In order to show that our methodology produces useful test collections and addresses the issues of scalability, reusability, and applicability, we must show that our inferred relevance judgments are correct and that our system can return far more relevant documents than traditional methods. 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 built around a corpus of about 500,000 clean text newswire articles. It is considered to have effectively complete assessments, with an average of about 1,736 assessed documents for each of 50 queries. 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 diversity task, which uses the ClueWeb09 html collection of about one billion documents. It contains an average of only about 528 documents assessed per query. 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. The TREC 8 ad hoc collection sample, denoted "SampleAdHoc", consists of approximately 11% of the documents assessed by TREC. The TREC09 web sample, denoted "SampleWeb", consists of approximately 38% of the documents 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. (See table below.)

Sample	Relevant Documents	Nuggets
SampleAdHoc	34.02	86.98
SampleWeb	25.18	61.82

Given the nuggets extracted, we employ the matching algorithm to infer binary relevance for all documents not contained in our sample. The relevance judgments produced are referred to by 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 remaining documents.

We demonstrate the correctness of our inferred judgments in two ways: by comparing our judgments to those provided by TREC, and by verifying the inferred relevance of documents that were not assessed by TREC with independent human assessors. However, we must bear in mind that the notion of correctness of relevance judgments is somewhat problematic. Inter-assessor disagreement [3, 9] is a well known phenomenon, as the question of relevance is ambiguous for many documents. Even if our inferences were perfect, we would still expect to see disagreement with our inferences and any other judgments.

After thresholding, we can compare our inferred judgments against the published TREC qrel, measuring precision, recall, F1, etc. (See table below.)

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

For comparison, a previously published comparison of inferred relevance judgments to existing qrels [2] cites an F1 of 0.68 (compared with our F1=0.75). From this we conclude that the relevance judgments produced by our methodology are very similar to those produced by the TREC assessors.

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 validated the correctness of our relevance judgments for documents from the web corpus not assessed by TREC. Our matching algorithm marked an average of about 400 additional documents as relevant per query. We selected a uniform random sample of about 80 per query and performed a validation of these inferred relevance assessments using Amazon's Mechanical Turk service.[1] (See table below.) Note that the "outside qrel" numbers given below are maximum likelihood estimates of the number of documents in each category given the random sample that was judged.

	Within qrel	Outside qrel	Total
Judged Rel	2,969	14,624	17,593
Judged NonRel	411	5,329	5,740
Total	3,380	19,953	23,333
Agreement	87.84%	73.29%	75.40%

This experiment resulted in an agreement of 73.29% between the Mechanical Turk judges and our inferred assessments. Given the overall 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.

Overall, even with nuggets extracted from a small sample of assessed documents, we were able to correctly infer relevance for a large number of relevant documents, with limited false positive mistakes and within the realm of inter-assessor disagreement.

5. CONCLUSION AND FUTURE WORK

Our methodology produces reliable relevance judgments at a substantially reduced cost. Based on reasonable assumptions,[2] we were able to construct SampleAdHoc in one-sixth the time and SampleWeb in half the time that TREC required to create the underlying collections.

Preliminary results with our nugget-based test collections indicate that the relevance judgments inferred from nuggets are also accurate enough to perform reliable and reusable system evaluations; in our pilot study, we obtained system orderings with Kendall's τ at or above 0.9 as compared to ground truth TREC assessments.

In future work, we intend to explore the application of nuggets to areas such as learning-to-rank, novelty and diversity, and so-called "knowledge engines" such as Wolfram|Alpha. We also hope to develop new performance metrics that uses nugget matching to directly measure the information content of the documents in a ranked list, rather than simply measuring the ranks at which relevant documents appear.

Acknowledgment: This material is based upon work supported by the National Science Foundation under Grant No. IIS-1017903. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

[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. Turkers were required to correctly assess 70% of these 10 documents; otherwise, the job was not accepted. Furthermore, some jobs were performed by multiple judges. 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.

[2] Private communication with a TREC organizer.

6. REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] Steve Krenzel. Finding blurbs. Website. http://www.stevekrenzel.com/articles/blurbs.

[8] 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.

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