Automatic Construction of Known-Item Finding Test Beds

Leif Azzopardi
Dept. of Computer and Information Sciences
University of Strathclyde, Glasgow G1 1XH
leif@cis.strath.ac.uk

Maarten de Rijke
ISLA, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam
mdr@science.uva.nl

Categories and Subject Descriptors

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

General Terms

Experimentation, Measurement

Keywords

Test collection formation, evaluation, simulation

1. INTRODUCTION

This work is an initial study on the utility of automatically generated queries for evaluating known-item retrieval and how such queries compare to real queries. The main advantage of automatically generating queries is that for any given test collection numerous queries can be produced at minimal cost. For evaluation, this has huge ramifications as state-of-the-art algorithms can be tested on different types of generated queries which mimic particular querying styles that a user may adopt. Our approach draws upon previous research in IR which has probabilistically generated simulated queries for other purposes [2, 3].

2. QUERY GENERATION

To create simulated queries, we model the following querying behavior of the user applying techniques similar to those applied in speech recognition (i.e., probabilistic generative language models). We assume that the user wants to retrieve a particular document that they have seen before in the collection, because some need has arisen calling for this document. This assumption eliminates the need for explicit relevance judgments as the known-item is the relevant document. The user then tries to re-construct or recall terms, phrases and features that would help identify this document, which they pose as a query. We model the actual process with the following algorithm.

- Initialize the empty query set $q = \{\}$
- Select the document d to be the known-item with probability $p(d)$
- Select the query length k with probability $p(k)$
- Repeat k times:
  - Select a term t from the document model of d with probability $p(t|\theta_d)$ (we assume query terms are drawn independently, though this need not be the case.)
  - Add t to the query set q.
- Record d and q to define the known-item/query pair.

By repeatedly running this algorithm we can create numerous queries. Before doing so, the probability distributions $p(d)$, $p(k)$ and $p(t|\theta_d)$ need to be defined. This is where we simulate the thought and behavior of the user by using different distributions to characterize the various types and styles of queries issued. The distribution with the most influence is the definition of the user's language model of the document, from which we sample query terms. Assume, for instance, that $p(t|\theta_d)$ is defined as a mixture of the maximum likelihood estimate of term occurring in a document and a background model $p(t)$, as in (1). Then we can directly influence the quality of the query. As λ tends to zero, the user's recollection of the original document improves. Conversely, as λ tends to one, the user's memory of the document degrades. If $\lambda = 1$, the user knows the document exists by they have no idea as to which terms appear in the document (and randomly select query terms).

$$p(t|\theta_d) = (1 - \lambda)p(t|d) + \lambda p(t) \quad (1)$$

Different query types can be generated by using different information to estimate the probability of a term being recalled by the user from that document $p(t|d)$. Examples include: (*popular*) using term frequency assumes that the user will recall the most popular or common terms in the document; (*discrimitive*) according to the inverse document frequency, where it assumes that the user will recall the most discriminative terms in the document; and (*uniform*) if a uniform distribution is used, it is assumed that the user will indiscriminately recall terms in the document.

3. EXPERIMENTS

We conducted several retrieval experiments to examine the differences using simulated queries versus real queries. We used the email sub-collection in the W3C corpus (called "lists"), which contains approximately 170,000 emails posted to the W3C forums over several years. The TREC topics KI1-25 and KI26-KI50 were concatenated to form 150

SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.

known-item (*real*) queries. We then generated one thousand queries for each of the three different styles of queries (*popular*, *discriminative* and *uniform*), using the following parameter settings: The probability $p(d)$ was uniform. The query length was constrained to a minimum of 3 and maximum of 7, with the length selected at random. The probability $p(t|\theta_d)$ was set proportional to the term frequency in the document (*popular*); the term frequency divided by the collection term frequency (*discriminative*); and uniform for terms occurring in the document (*uniform*), and $\lambda = 0$. Terms that were less than three characters in length were discarded.

To examine how the simulated queries perform on different retrieval models against the real queries, we used three language models previously used on this task at TREC [1]: a standard language modeling approach, a fielded language modeling approach, and a combination language modelling approach, where the later two are structured language models. Our aim is to determine if the simulated queries provide a comparable indication of the system performance as reported by real queries and whether the simulated queries are helpful in identifying differences between models.

4. RESULTS AND DISCUSSION

Table 1 provides examples of real and generated queries, where clearly, the terms in each query style vary with some more realistic than others. The *popular* method appeared to produce reasonably natural queries. The *discriminative* method produced queries which tended to be rather artificial, with some bearing little resemblance to actual queries posed. The *uniform* method provided a random selection of query terms that gave the impression of more realistic queries than the discriminative method. It is unlikely, however, that a user submitting a query will randomly select the query terms from the known-item. This is because it is more likely that the most memorable terms in the document are issued as query terms, which are generally those that are more discriminative, more frequent or more obvious (because of location, for instance). Presumably users will combine popular and discriminative terms within a query, which may give the impression of query terms appearing more or less random.

Table 2 shows the performance of the each retrieval model for the different types of simulated queries in terms of Mean Reciprocal Rank (MRR). Significance differences between models on the MRR is indicated by the letters a to c, which correspond to a particular retrieval model in the table. If the MRR of the model in that tuple was significantly better than another the letter of the model is shown. (All tests were performed using the paired Wilcoxon Signed Rank Test, where the significance level was set to 5%).

Table 1: Examples of Real and Generated Queries

Method	Query Terms
real	html access improvement draft Judy Brewer
	releas owl recommend
	addition new button bar beneath Amaya's menu bar
popular	web folder incompatible Henrik
	entity Gavin Nicol
	possible thing September
discrim.	markupdec Fred
	fit taxonomy sucess targtyp Len Bullard
	element value 6a0700000010 691c0000000d
uniform	more interest erratum rule XPath
	pdf Tuesday David Burdett
	valid system access communication read Friday

Table 2: Retrieval Performance in MRR

Model	*Real*	*Popular*	*Discrim.*	*Uniform*
(a) Stand. LM	0.47	0.30	0.72	0.28
(b) Field. LM	0.55a	0.31	0.67	0.31a
(c) Comb. LM	0.63ab	0.37a	0.74ab	0.37ab

From the results, the performance of the different query generation methods provide a similar ordering of the different retrieval strategies. The *discriminative* queries tend to overestimate the performance of model (a), but show that model (c) is still clearly superior. The *uniform* and *popular* queries tend to underestimate the performance of the models. However, the *uniform* queries do provide a similar indication of significance between retrieval strategies as the *real* queries. This suggests that using the *uniform* simulated queries for comparing systems will provide a good indication of the relative performance of systems. From a diagnostic perspective, we can see that model (c) is very robust to the different query generation styles. The explicit control gained through the simulated queries provides a finer grained analysis of retrieval models. With more sophisticated query generators many other user querying styles can be modeled, including bi-grams, translations, noise, structure, document priors, and so forth. This would provide a suite of user query types to evaluate known-item finding techniques in a fine-grained and comprehensive manner.

5. CONCLUSIONS AND FURTHER WORK

The different sampling methods, while an abstraction of the actual querying process, accentuate the different styles of queries that a user may issue. This is useful for developing, training, and evaluating retrieval models and techniques, especially since a sufficiently large number of queries can be generated cheaply and quickly to examine how the different styles will affect different retrieval models. Our initial results motivate further work to be directed in three main areas: (1) understanding how simulated queries can be applied in the context of IR evaluation; (2) developing a suite of user querying models including methods for evaluating whether the queries produced are realistic; and (3) application to other IR tasks such as ad hoc retrieval.

6. ACKNOWLEDGMENTS

This research was supported by the Netherlands Organization for Scientific Research (NWO), project 220-80-001.

7. REFERENCES

[1] L. Azzopardi, K. Balog, and M. de Rijke. Language Modeling Approaches for Enterprise Tasks. In *Proceedings TREC 2005*, 2006.

[2] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings SIGIR-99*, pages 222–229, Berkeley, CA., 1999.

[3] J. P. Callan and M. E. Connell. Query-based sampling of text databases. *Information Systems*, 19(2):97–130, 2001.