Relevance Feedback and Personalization:
A Language Modeling Perspective

W. Bruce Croft, Stephen Cronen-Townsend and Victor Lavrenko
Computer Science Department
University of Massachusetts
Amherst, MA 01003-4610
{croft, crotown, lavrenko}@cs.umass.edu

Abstract

Many approaches to personalization involve learning short-term and long-term user models. The user models provide context for queries and other interactions with the information system. In this paper, we discuss how language models can be used to represent context and support context-based techniques such as relevance feedback and query disambiguation.

1. Overview

From some perspectives, personalization has been studied in information retrieval for some time. If the goal of personalization is to improve the effectiveness of information access by adapting to individual users' needs, then techniques such as relevance feedback and filtering would certainly be considered to support personalization. There has also been considerable research done, mostly in the 1980s, on user modeling for information retrieval. This research had essentially the same goal as current research on personalization, which is to build a model of a user's interests and preferences over time. Filtering systems, too, emphasize learning a user's interest profile (or profiles) over time. Relevance feedback, on the other hand, has the goal of learning a model of the user's interest in a single search session. This is short-term personalization. Starting with the same query, two users could end up with very different documents or answers depending on their feedback. Put another way, the system uses the user's feedback to learn the specific *context* they have in mind for the initial query.

Query expansion or local feedback techniques are also related to personalization of context. Based on the user's query and the document corpus, possible contexts for the query are inferred and used to suggest additional query terms. Although "true" personalization would make use of a long-term profile or user model to choose contexts, and therefore additional query terms, for each individual user, we believe that the representation of context and how it can be inferred is a basic component of any approach to content personalization. By studying these issues in the short-term (a single session), we may learn how to handle context effectively in the long-term (over multiple sessions).

Relevance feedback, despite its long history in information retrieval research, has not been successfully adopted in Web-based search engines. The closest feature found in some search engines is "find more documents like this". Query expansion techniques have been used in a number of systems to suggest additional search terms, with limited success. There are a number of reasons for the apparent failure of relevance feedback in current systems. The primary one that is usually mentioned is the difficulty of getting users to provide the relevance information. Simply providing "relevant" and "not relevant" buttons in the interface does not seem to provide enough incentive for the user. For this reason, a number of researchers are investigating techniques to infer relevance through passive measures such as time spent browsing a page or number of links followed from a page. Even if the relevance information was provided, however, there is a significant problem with using the current feedback techniques. With full text and limited relevance information, the relevance feedback techniques developed in the 70's and 80's are simply not as reliable as the experiments with collections of abstracts had indicated. In other words, identifying the correct context is not simple. Experiments have shown that if a user can indicate relevant sections or even phrases in a document, relevance feedback is more accurate. This, however, seems to imply that we will need more input from the users rather than less.

In summary then, relevance feedback and query expansion are personalization techniques that attempt to infer the context of a user's query from top-ranked documents and additional feedback. The successful application of relevance feedback is not easy and involves both sophisticated interface design and good algorithms for inferring context. In this paper, we focus on the algorithms for inferring context. In particular, we describe how a language modeling approach to the problem can lead to new perspectives on relevance feedback and query ambiguity.

2. Relevance Feedback and Language Models

There are a number of formal ways of describing relevance feedback, beginning with the notion of an "optimal query" used in the SMART system (Salton, 1968). The optimal query is defined as a vector obtained by taking the difference between the relevant and non-relevant sets of documents, also represented as vectors. This query vector can be shown to be the "best", under some assumptions, for distinguishing relevant and non-relevant documents. This relevance feedback approach evolved to a query modification process where the old query is modified by a weighted average of the identified relevant documents and, in some versions of the algorithm, the identified non-relevant documents. The revised query is then used to produce a new document ranking. Another common way of describing relevance feedback is to a Bayesian classification model of retrieval (Van Rijsbergen, 1979). In this approach, identified relevant documents are used to estimate the characteristics (probabilities of occurrences of words) of the relevant class of documents for a query. The corpus is used to estimate probabilities for the non-relevant class. The revised estimates are used to produce a new document ranking based on the probability of belonging to the relevant class. Both of these approaches can be viewed as applications of different machine learning techniques to the problem of identifying relevant documents based on training data. There have been a number of other recent experiments with machine learning techniques, but these have not in general shown significant improvements over the approaches already described (for example, Schapire et al, 1998). Any of these techniques are always more effective when applied to document filtering rather than relevance feedback, because in that application there is significantly more training data.

The language model approach to feedback does not at first appear to lend itself to relevance feedback. In the basic approach, first suggested by Ponte and Croft (1998), each document is represented by a document language model. The query is treated as a sample of text from a language model, and the documents are ranked according to the probability that the document language model could generate the query text. This simple model produces surprisingly good retrieval results, and the model has been extended in a variety of ways. For example, documents have been modeled as mixtures of topics (Hoffman, 1999) and translation probabilities have been introduced to deal with synonymy and cross-lingual retrieval (Berger and Lafferty, 1999).

In terms of relevance feedback, the basic approach has been to modify the initial query using words from top-ranked (for query expansion) or identified relevant documents. Ponte (2000) simply adds some additional words to the query based on the log ratio of the probability of occurrence in the model for relevant documents to the probability in the whole collection. The model for relevant documents was taken as the sum of the individual document models. In Miller et al (1999), words from relevant documents were added to the query and probabilities in their model adjusted by training over queries. Although both of these approaches produced good results, they are not very satisfactory models from the point of view of describing and defining query contexts and user models, which are central to personalization.

In order to better capture the important processes behind relevance feedback and query expansion, we believe it is important to view the query as a sample of text from a model of the information need. That is, documents are generated from document language models associated with authors and queries are generated by information need language models associated with individual users. This raises the interesting possibility that users, like documents, could be represented as a mixture of topic language models generated from previous interactions and other sources. From this perspective, the task of the relevance feedback or query expansion component of the system is to infer the language model associated with the query. Given this query model, retrieval could then be done either by ranking the documents according to their probability of being generated *by the query model*, or the query model and the document models could be directly compared and documents ranked by the similarity of these models. From the language model perspective, inferring the query language model is what is meant by inferring the context of the query. Give the query model, we can predict good suggestions for additional search terms and produce better results by personalizing the search. The problem, of course, is that we still have to estimate the query model from very limited data. This is discussed in the next section.

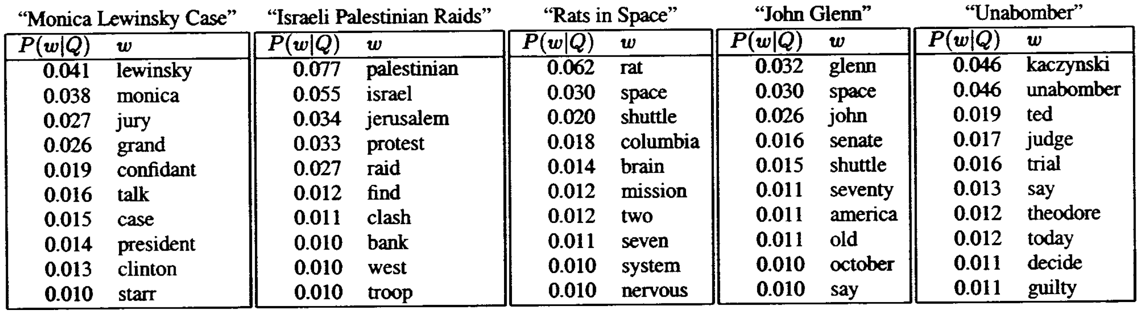
3. Inferring Language Models

To infer the query model from either top-ranked documents (in the case of query expansion) or identified relevant documents (in the case of relevance feedback) requires an approach where document text can be viewed as samples from the query model. In a recent paper, Lavrenko and Croft (2001) show how a relevance-based probabilistic model of retrieval can be described in language modeling terms. Documents are ranked by the ratio *P(D|R)/P(D|G)* where *P(D|R)* is the probability of generating the document text *D* given the relevance (query) model *R* and *G* is a global or corpus model. They then describe a technique for constructing a query model with no relevance data. This technique uses the document models associated with the top-ranked documents to estimate the language model probabilities *P(w|R)*, where *w* is a word. This computation involves the expression

$$(1)\ \prod_{q\in Q}\left(\frac{P(M)}{P(w)}\sum_{M}P(q\,|\,M)P(w\,|\,M)\right)$$

where *M* is a possible model, and *Q* is the query. In the paper, the top ranked documents provide the possible models and the process is very similar to successful, ad-hoc query expansion techniques such as LCA (Xu and Croft, 2000). Table 1 shows some examples of word probabilities for the relevance model generated by this approach. Relevance feedback would provide additional information about relevance that could be incorporated with a high prior probability *P(M)*. In other words, relevance feedback would be done by mixing information from relevant documents and top-ranked documents. Our belief is that this will provide more robust performance. This approach also provides a mechanism for incorporating other information about the user into the estimation of context. “Personalized” language models derived from previous interactions or preferences could be mixed with relevance information and top-ranked documents.

Table 1: Sample probabilities from the query-based relevance models on the TDT2 corpus.



“Monica Lewinsky Case”		“Israeli Palestinian Raids”		“Rats in Space”		“John Glenn”		“Unabomber”	
$P(w\|Q)$	w	$P(w\|Q)$	w	$P(w\|Q)$	w	$P(w\|Q)$	w	$P(w\|Q)$	w
0.041	lewinsky	0.077	palestinian	0.062	rat	0.032	glenn	0.046	kaczynski
0.038	monica	0.055	israel	0.030	space	0.030	space	0.046	unabomber
0.027	jury	0.034	jerusalem	0.020	shuttle	0.026	john	0.019	ted
0.026	grand	0.033	protest	0.018	columbia	0.016	senate	0.017	judge
0.019	confidant	0.027	raid	0.014	brain	0.015	shuttle	0.016	trial
0.016	talk	0.012	find	0.012	mission	0.011	seventy	0.013	say
0.015	case	0.011	clash	0.012	two	0.011	america	0.012	theodore
0.014	president	0.010	bank	0.011	seven	0.011	old	0.012	today
0.013	clinton	0.010	west	0.010	system	0.010	october	0.011	decide
0.010	starr	0.010	troop	0.010	nervous	0.010	say	0.011	guilty

4. Quantifying Ambiguity

Many of the queries presented to an information retrieval system are ambiguous. That is, the context of the query is not clear. An often-used example of such a query in Web search is “java”, where the context could be programming languages, Indonesia, or coffee (or possibly others). Given such a query, the only way the ambiguity can be resolved is to ask the user for clarification in some form. Since there will be many thousands of documents on Java programming and many fewer on Indonesia, relying on relevance feedback to resolve the problem is unrealistic. Instead, a better method would be to first determine if a query is ambiguous, then ask the user specific questions to resolve the ambiguity.

The first part of addressing query ambiguity is to define it and quantify it. We are currently developing a language model framework to do this. This approach assigns a "clarity" value to query based on how different its associated language model is from the corpus language model. Clarity is a non-negative number, with a zero being assigned to a maximally vague query whose associated language model is indistinguishable from the corpus language model.

For example, the one-word query "apple" should get a low clarity score in a general news database since the associated language is partly related to apple pies, partly related to New York City (the "Big Apple"), and partly related to Apple Computer Company (assuming capitalization is ignored). The query "Apple Computer Company" would receive a significantly higher score in the same collection, since the associated language is more specialized and not blurred by the competing meanings of apple pie and New York City. However, in a collection of articles consisting solely of technology news the queries "apple" and "apple computer company" would have similar clarity scores.

In our current efforts, we use unigram language models (probability distributions over terms, *w*). Given such a model for the language associated with a query *Q*, *P(w|Q)*, and a such a model, *P(w)*, for the collection as a whole, we define clarity as the Kullback-Liebler (KL) divergence between the collection distribution and the query distribution, namely,

$$(2)\ \ clarity \equiv D(P(w) \,\|\, P(w \mid Q)) = \sum_{w \in V} P(w) \log_2 \frac{P(w)}{P(w \mid Q)}$$

where *V* is vocabulary, the set of all terms occurring in the collection. This is used in preference to the symmetrically-related quantity *D(P(w|Q)||P(w))* since it can be more accurately estimated. Interpreted in terms of coding theory, clarity is the average number of bits that would be wasted encoding word events from the collection model with a code that was optimally designed for the query model.

The main ingredient in computing numerical values for clarity is the query language model, *P(w|Q)*. We use two types of language models that we will refer to as probability-weighted models and relevance models. In the probability-weighted approach the language model for the query is taken to be

$$(3)\ \ P(w \mid Q) \cong \sum_{d \in R} P(w \mid d) P(d \mid Q),$$

where the query, *Q*, is interpreted as in Song and Croft (1999), *R* is the set of retrieved documents, and *P(Q|d)* is the probability that the query is generated by a specific document model. *P(d|Q)* is the Bayesian inversion of *P(Q|d)* with uniform document priors. Individual document models are estimated by fixed parameter linear smoothing with the corpus model

$$(4)\ \ P(w \mid d) = \lambda P_{ML}(w \mid d) + (1 - \lambda) P(w),$$

where $P_{ML}(w|d)$ is simply the maximum likelihood probability estimate computed as the number of occurrences of the term in the document divided by the number of terms in the document and *P(w)* is estimated by the relative frequency of the term in the corpus. We have used $\lambda=0.8$ in this work.

Equation (3) can be efficiently calculated by replacing the distribution *P(d|Q)* with an approximation that uses a fixed number *N* of documents rather than the entire retrieved set. A histogram is formed by sampling from *P(d|Q)* and adding one to the corresponding bin as long as there are less than *N* non-zero bins. As soon as the $N+1^{\text{th}}$ unique document would be sampled the process is stopped and the histogram is normalized to form the an approximate distribution used in place of *P(d|Q)* in (3). A value of *N=500* was found empirically to give accurate results (compared to using the entire retrieved set) with a variance of roughly 1%. This guarantees that system will only have to mix 500 (or less) documents or in the computation, resulting in large performance gains, particularly for large collections. Using these techniques on equations (2)-(4) leads to the values shown in the second column of Table 2.

The second type of query language model we use (and discussed in section 3) is the relevance model of Lavrenko and Croft (2001). These models have been implemented on a sampled set of documents with smoothed models from equation (4) in such a way that the final model is a valid distribution over all terms in the vocabulary. The two types of language models produce the same probability distribution over all terms in the vocabulary in our implementation, as they must. This second type of language model together with equation (2) produces the values shown in the third column of Table 2. These values produce almost the same ranking as in the other model, with the notable exception that adding the unrelated term "alabama" to the query "indonesia opec" causes the clarity score to drop even slightly below the score of "indonesia" alone. This captures the vagueness inherent in adding a term that does not co-occur with the other query terms and seems to favor using relevance models in computing query clarity.

Table 2: Clarity values using two types of query language models in the allNews collection. AllNews contains the ap88-90, ap94-98, ny97-98, latimes, sjmn91, wsj87-89, wsj90-92, and ft91-93 collections from TREC.

Query	**Probability-weighted LM**	**Relevance model**
"apple"	0.47	0.64
"apple computer company"	0.79	0.93
"indonesia"	0.53	0.63
"indonesia opec"	0.86	0.79
"indonesia opec oil"	0.96	1.00
"idndonesia opec alabama"	0.78	0.61

After quantifying the ambiguity of the query, we must then decide how to resolve it. If the query is sufficiently ambiguous (according to our measure), we should be able to identify the most probable contexts (word associations, language models). Given those contexts, we plan to select sample sentences that are representative of them. This means the sentences have high probabilities of generation in those contexts. The sentences would be shown to the user for clarification. This process can be viewed as a generalization of a KWIC index.

By using a language model view of ambiguity, we hope to resolve the context of the query quickly and with minimal user input. The end result of this process will be more accurate search results.

5. Summary

Language models provide a potential representation for users and contexts. We described how relevance feedback and query ambiguity could be described using this approach. We also suggested how additional information about the user could be incorporated into the context estimation process. Much of this is preliminary; and many more experiments need to be done. We are currently doing relevance feedback and query ambiguity experiments using TREC data. In future work, we intend to incorporate long-term user models and task models.

Acknowledgments

This paper describes some of the ongoing research in the Mongrel project, which is a collaboration between Professor Nick Belkin at Rutgers University, and Professors Allan and Croft at the University of Massachusetts. A description of the project can be found at http://www.scils.rutgers.edu/etc/mongrel/. The paper is also related to research done in collaboration with Professor John Lafferty at Carnegie Mellon University. This research is supported in part by the Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623 and in part by NSF Grant number ISS-9907018. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

References

A. Berger and J. Lafferty, “Information retrieval as statistical translation”, Proceedings of ACM SIGIR 99, 222-229, 1999.
T. Hoffman, “Probabilistic latent semantic indexing”, Proceedings of ACM SIGIR 99, 50-57, 1999.
V. Lavrenko and W.B. Croft, “Relevance-based language models”, to appear in ACM SIGIR 2001.
D. Miller, T. Leek and R. Schwartz, “A Hidden Markov Model information retrieval system”, Proceedings of ACM SIGIR 99, 1999.
J. Ponte and W.B. Croft, “A language modeling approach to information retrieval”, Proceedings of ACM 98, 275-28, 1998.
J. Ponte, “Language models for relevance feedback”, in *Advances in Information Retrieval*, ed. W.B. Croft, 73-96, 2000.
C.J. Van Rijsbergen, *Information Retrieval*, Butterworths, 1979.
G. Salton, *Automatic Information Organization and Retrieval*, McGraw-Hill, 1968.
R. Schapire, Y. Singer and A. Singhal, “Boosting and Rocchio applied to text filtering”, Proceedings of ACM SIGIR 98, 1998.
F. Song and W.B. Croft, “A General Language Model for Information Retrieval”, Proceedings of the Conference on Information and Knowledge Management (CIKM), 316-321, (1999)
J. Xu and W.B. Croft, "Improving the Effectiveness of Information Retrieval with Local Context Analysis", *ACM Transactions on Information Systems*, 18(1), 79-112, (2000).