User Lenses – Achieving 100% Precision on Frequently Asked Questions

Christopher C. Vogt[1], Garrison W. Cottrell[1], Richard K. Belew[1], and Brian T. Bartell[2]*

[1] Department of Computer Science and Engineering, University of California, San Diego, CA, USA
[2] Conceptual Dimensions, Inc., San Diego, CA, USA

Abstract. The concept of a "user lens" is introduced. The lens is a sequence of linear transformations used to reweight the vectors which represent documents or queries in information retrieval systems. It is trained automatically via relevance data provided by the user. Experiments verify the lens can improve performance on training data while not degrading test data performance, and that larger lenses result in nearly perfect performance on the training set. The lens provides a mechanism for automatically capturing long-term, user-specific information about an improved representation scheme for document vectors.

1 Introduction

Information Retrieval (IR) is the task of finding documents which are relevant to a user's query. It is typified in the World Wide Web context by search engines and filtering agents. Search engines are approaches for handling what is known as the "adhoc" task – a user issues a new query against a static collection of documents. Filtering agents, on the other hand, embody the "routing" task – the user has a standing query against which new documents (e.g., newswire articles) are compared.

The division of the IR task into adhoc and routing subtasks is one which researchers naturally make to help them analyze their approaches. Realistically, however, no IR system solves either of these tasks alone, since neither document nor query collections are really static. Viewing each task separately has lead to some standard approaches to one task which are not compatible with the other, making combination more difficult. For example, when solving the adhoc task, traditional IR systems use methods of relevance feedback like term reweighting and term expansion in a short-term fashion. Once the user has finished with a query, the system forgets about the feedback he or she has provided. Systems solving the routing task generally do not fall prey to this problem, since they create and maintain fine-tuned versions of each of the user's queries. On the other hand, each such query is viewed in isolation from the others, with the system making no attempt to create an overall picture of the user's "view of the world." Construction of such a view could make it easier for the system to satisfy future queries (i.e., solve the adhoc task).

Another seemingly unrelated, yet perennial problem in the IR community is that of choosing term weighting schemes. A weighting scheme is just a method for determining how much each term (i.e., word or phrase) in the document will contribute to the representation of the document. These representations are typically in a vector form, with one component of the vector representing each term, and the value of that component (i.e., its weight) indicating how much

* This research was supported by NSF grant IRI 92-21276.

the document is about that term. These weights are almost always a function of how often the term occurs in the document (its frequency). Countless researchers have tried to find the best scheme for a particular problem or domain, or even just the best overall scheme (Salton and Buckley, 1988). Searching for the best overall scheme is clearly not the best approach, since the scheme will necessarily be suboptimal in some situations. Likewise, the manual search for the best weighting scheme for a particular domain seems grossly inefficient and unlikely to produce the optimal scheme. A more reasonable approach would be to determine the weighting scheme automatically.

The research presented here addresses both of these problems, and introduces a new way of modeling the user. We introduce the idea of a "user lens." The lens is merely a set of IR system parameters which are automatically adjusted according to the user's relevance feedback. The lens is used to modify the default vector-space weighting scheme, creating a new scheme for representing documents and/or queries. Furthermore, because a modified vector-space approach is used, the system can be used for both routing and adhoc tasks. The lens indirectly preserves all of the user's feedback while simultaneously representing his or her view of the world. This lets the system leverage off of what it has learned from either task when performing the other. The lens presents a simple yet powerful way of modeling the user's behavior, and thus his or her preferences.

2 User Lenses

A "user lens" is simply a sequence of matrices by which the document or query vectors are multiplied to obtain a new representation. The document representations are then compared to the queries using the standard inner-product or cosine measures. The entries in the lens matrices are adjusted automatically using the information gained via relevance feedback. We preface the term lens with "user" to emphasize that each user (or possibly group of users) could have their own lens which gets used whenever they use the system, which they train with their own feedback, and which represents in a crude way their wants, needs, and usage idiosyncrasies. Thus, the lens can be viewed as a rough model of the cognitive processes of the user when he or she is creating the query or interpreting a document.

Figure 1 summarizes our concept of a user lens. We use Latent Semantic Indexing (Deerwester et al., 1990) as our representation. LSI takes the very large document and query vectors and maps them into a smaller dimensional space. We use this technique because if a square matrix were used to modify the original document/query vectors, it would have millions or billions of entries to adjust. Reduced LSI vectors make the lens more manageable, with tens of thousands of entries. The figure is meant to show the entire process through which a ranking score is derived from the document and query vectors. First, each is reduced to a much lower dimensional vector using the LSI matrix S. Then each is optionally multiplied by one of the matrices L_d or L_q. The final score R for the document with reference to the query is calculated using these transformed vectors. Mathematically, this is:

$$R(d, q) = (L_d S d)^T (L_q S q)$$

assuming both lenses are used and the inner-product is the comparison operator. Since there are no restrictions on the lens matrices, any linear transformation can be used to reweight the input vectors.

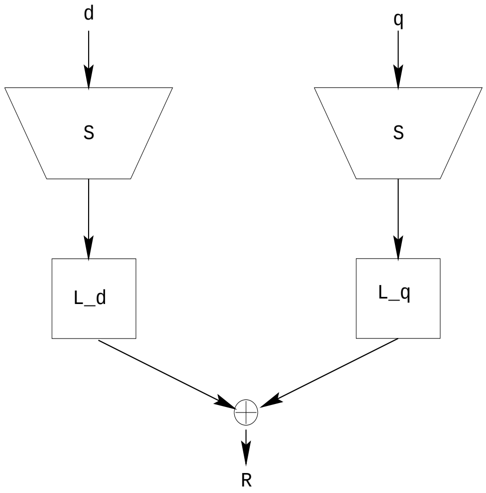

Figure 1. The “user lens” – document and query are first reduced using LSI matrix S, then possibly multiplied by a matrix before being combined into a ranking score R.

Our technique for adjusting the entries in the lens matrices is based on work done by Bartell et al. (1995), (1994). Bartell defines a criterion (hereafter called J) based on Guttman’s Point Alienation statistic as follows:

Defn: the ranking function implemented by an IR system is

$$R_{\Theta} : \Theta \times D \times Q \rightarrow \Re$$

where
Θ =the set of system parameters
D=the set of document vectors
Q=the set of query vectors

Defn: Bartell’s J criterion is:

$$J(R_{\Theta}) = \frac{1}{|Q|} \sum_{q \in Q} \frac{\sum_{d \succ_q d'} (R(\Theta, d, q) - R(\Theta, d', q))}{\sum_{d \succ_q d'} |R(\Theta, d, q) - R(\Theta, d', q)|}$$

where $d \succ_q d'$ indicates the user prefers document d to document d' on query q.

Note that J has a maximum value of 1 when the numerator and denominator are the same (i.e., the IR system ranks documents exactly as the user would), and a minimum value of -1 when the opposite is true. Also note that in order to calculate J, we need to have feedback from a user – for each query, the user needs to label *every* document as either relevant or not relevant to the query. This time consuming and tedious process necessarily limits the amount of training data available.

In our formulation, the entries in the lens matrices L_q, L_d, and S are the system parameters Θ. We then use a multivariate optimization technique like gradient descent or conjugate gradient to determine those values of the matrices which minimize J, by taking the partial derivative of J with respect to the system parameters.

3 Experiments

We have performed two related series of experiments to verify the validity of the lens idea on the adhoc task. The first series verifies the lens as capable of improving system performance, the second series attempts to pinpoint which configuration of lenses produces the most improvement. Because we are examining the adhoc task, two types of improvement are possible: on the training data (this is equivalent to getting better on frequently asked questions) or on the test data (getting better on new queries).

The two most accepted measures of IR system performance are precision and recall. **Precision** is the percentage of retrieved documents which are relevant: $\frac{|R_{EL} \cap R_{ET}|}{|R_{ET}|}$ (where R_{EL} is the set of all relevant documents and R_{ET} is the list of retrieved documents). Precision is a measure of the quality of retrieval. **Recall** is the percentage of relevant documents which have been retrieved: $\frac{|R_{EL} \cap R_{ET}|}{|R_{EL}|}$. Recall measures the coverage of retrieval. Maximizing one of these two measures typically minimizes the other. This tradeoff can be seen in what is a common view of IR system performance: the precision versus recall graph. Precision is calculated at different levels of recall, and these scores are then graphed. Note that high precision corresponds to low recall and vice-versa. The ideal IR system would have a horizontal line across the graph at precision level 1.00, which would correspond to retrieving all and only the relevant documents.

3.1 System Verification

In these experiments, we use both the CISI and MED corpora (distributed with the SMART system (Salton, 1971)). Respectively, these are corpora of Information Science and Medical abstracts, each containing about 1000 documents. Only the query lens L_q is trained, with a cosine comparison operator. The CISI corpus is indexed using the SMART *atc* weighting scheme, and MED is indexed using *ntc*. In both cases, document and query vectors are reduced to 100 dimensions using LSI. The number of training queries is 56 for CISI and 20 for MED, with 10 and 5 test queries respectively. Relevance feedback on all of these 66 and 25 queries are distributed along with the corpora. In both cases, eight different random partitions of the queries in training and testing sets are made, and the average precision/recall figures over these eight partitions are reported below. L_q is initialized to the identity matrix before being trained. Gradient descent with a learning rate of 0.1 is used to minimize J by adjusting the entries of L_q.

For the CISI data, a number of different lenses are trained, each corresponding to a different number of iterations of the gradient descent algorithm. Precision/recall curves corresponding to 200, 1000, and 2000 iterations are shown in Figure 2, along with a baseline of no lens. It is clear from these graphs that a lens can indeed improve performance on the training data and yet not affect performance on the test data. However, as the 1000 and 2000 iteration results show, the large improvement gained by longer training comes at a cost of degradation on the test data. This suggests that a hold-out set be used to determine when to stop training. More importantly, though, this may mean that the lens does not have quite enough power to produce the best possible weighting scheme. This latter idea is explored in the second set of experiments, where we actually train the entries of the LSI matrix S. Because S is so much larger than the L matrices, we hypothesize that training it will allow much better performance on the training data and possibly even some improvement on the test data.

To verify that the results on CISI are not specific to that corpus, we also train a query lens L_q for 2000–5000 iterations of gradient descent on the MED corpus. The results in Figure 3 show the same pattern of improvement on the training data and no effect on the test data performance.

3.2 Finding the Best Configuration

As described above, many options are available when using a lens. In this set of experiments, we explore several configurations in an attempt to determine which results in the best performance on the CISI corpus. The three configurations we use are: training the document lens L_d only, training the document and query lenses simultaneously so that $L_q = L_d$, and as mentioned above, training the LSI matrix S alone. Training the query lens L_q results in performance virtually identical to that of L_d, so it is not shown on the graphs below. Also, in the first two configurations, gradient descent with 200 iterations is used, whereas the faster converging conjugate gradient is used to train S because it is so large.

Figure 4 shows the results of these experiments. As expected, performance on the test data does not change after training, but the performance on the training data improves. Even more importantly, we see that running both the document and the query through a lens results in even better performance. This behavior is not only expected, but desirable, since if a document is used as a query, one would expect the highest possible ranking score when it is compared to itself. Finally, if we are willing to spend the extra time training a much larger set of parameters (i.e., the LSI matrix), then we can get *nearly perfect precision at all levels of recall* on the training data while marginally improving the test data performance (although, this latter improvement is probably not statistically significant). This supports the hypothesis suggested by the first set of experiments that the query lens alone was just not large enough to maximize performance. However, another possibility is that good solutions are not even capable of being represented by transforming the reduced space constructed by LSI. This is an issue we will have to explore in future experiments.

4 Discussion and Conclusions

Use of relevance feedback information as a means of modifying a single query to improve a system's retrieval is becoming a common feature of many IR systems. The second set of experiments imply that modifying the *document* representation, in addition to the query representation, can generate significant improvements. In fact, with the right model (e.g., training the entire LSI matrix), modifying both document and query representations can result in nearly perfect performance on the training data without degrading performance on the test data, producing 100% precision on frequently asked questions.

The two experiments raise a number of interesting and important points. First, we reconfirm Bartell's results: Optimization according to the J criterion results in improved performance as measured by precision. In related work, we have found that optimizing J also worked well in some larger scale experiments (Vogt et al., 1997). Together with the results here, it seems to

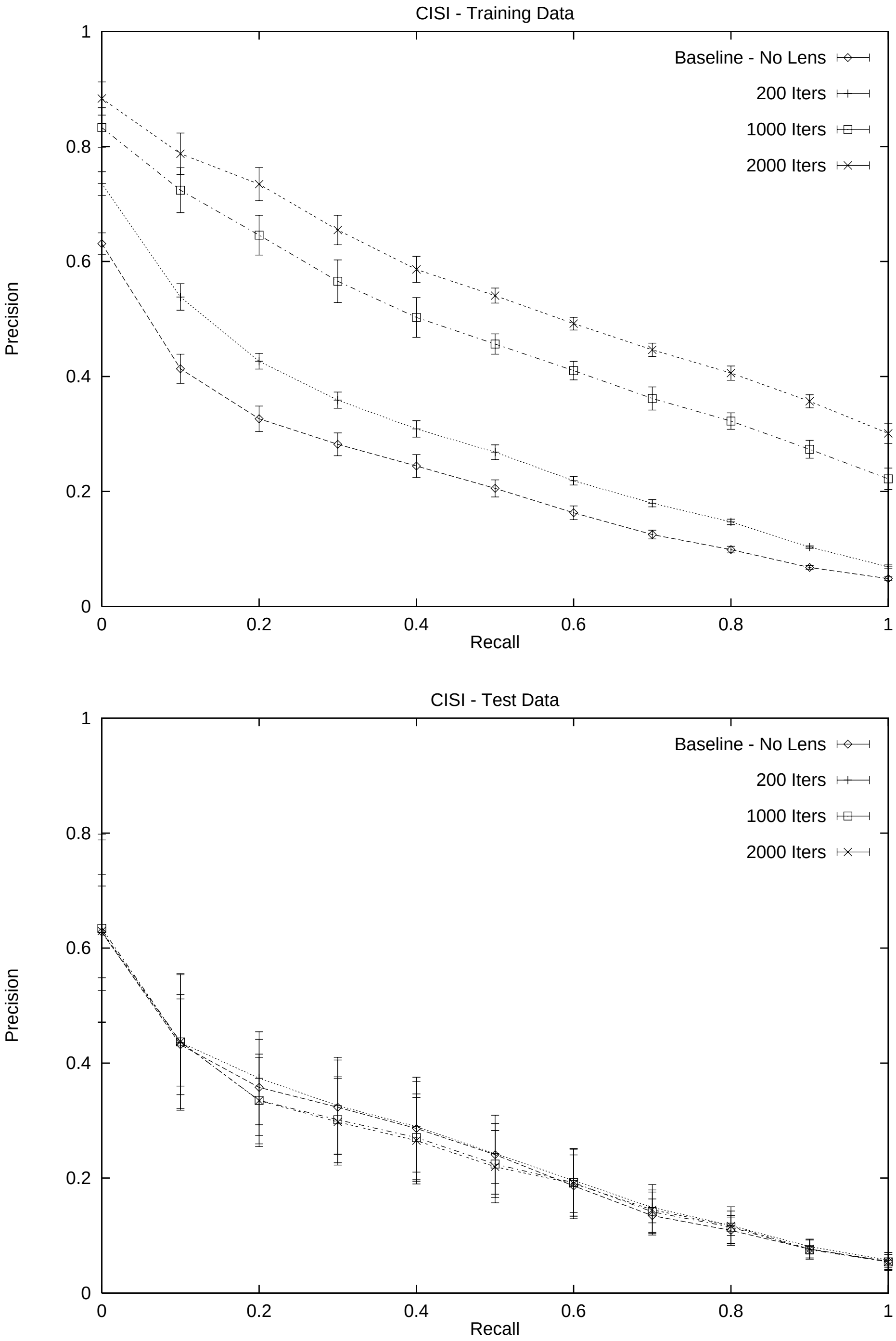

Figure 2. Results of system verification on the CISI training and test data – Comparison of different number of iterations of gradient descent averaged over 8 Runs (error bars are one standard deviation).

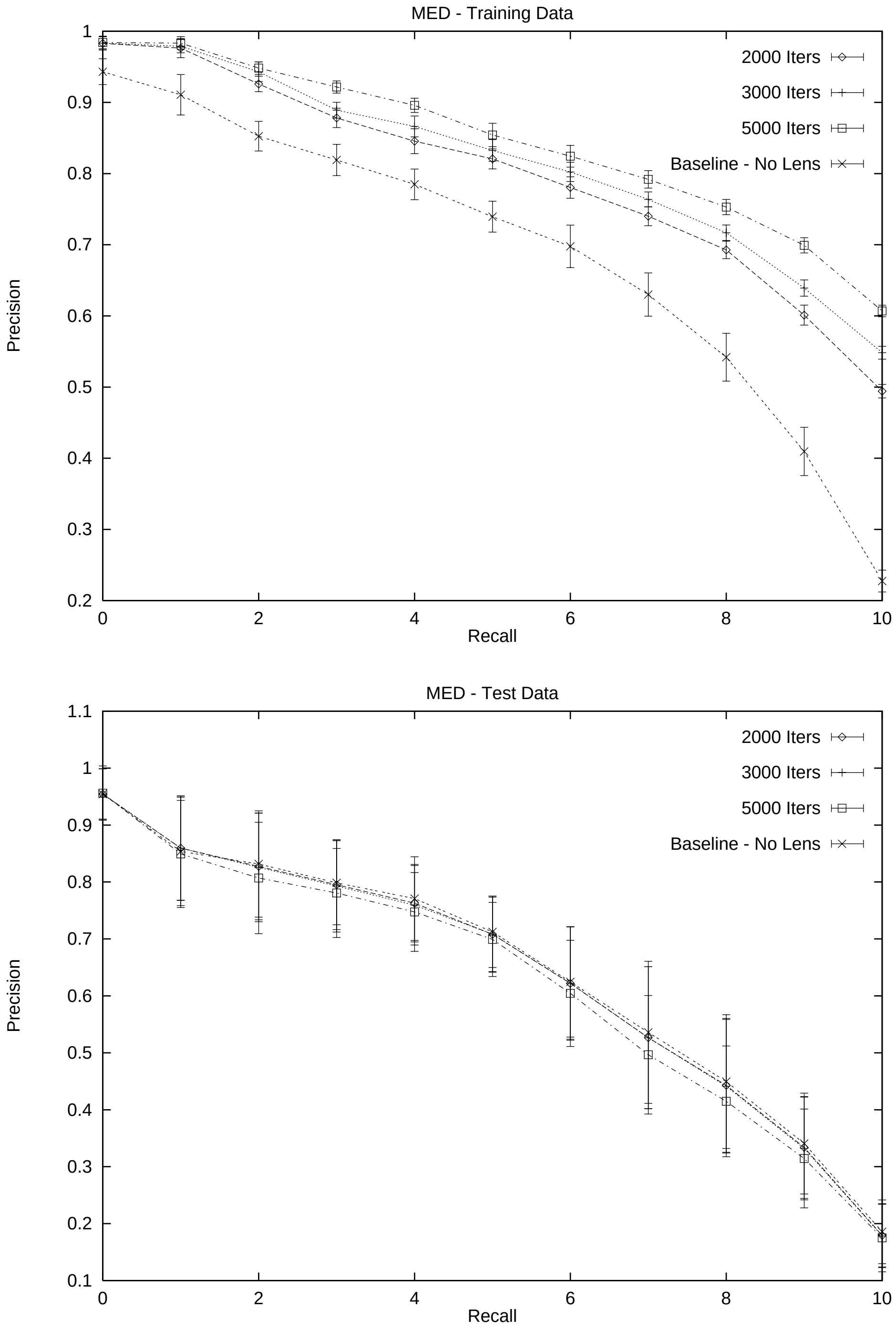

Figure 3. Results of system verification on the MED training and test data, averages over 8 runs (error bars are one standard deviation).

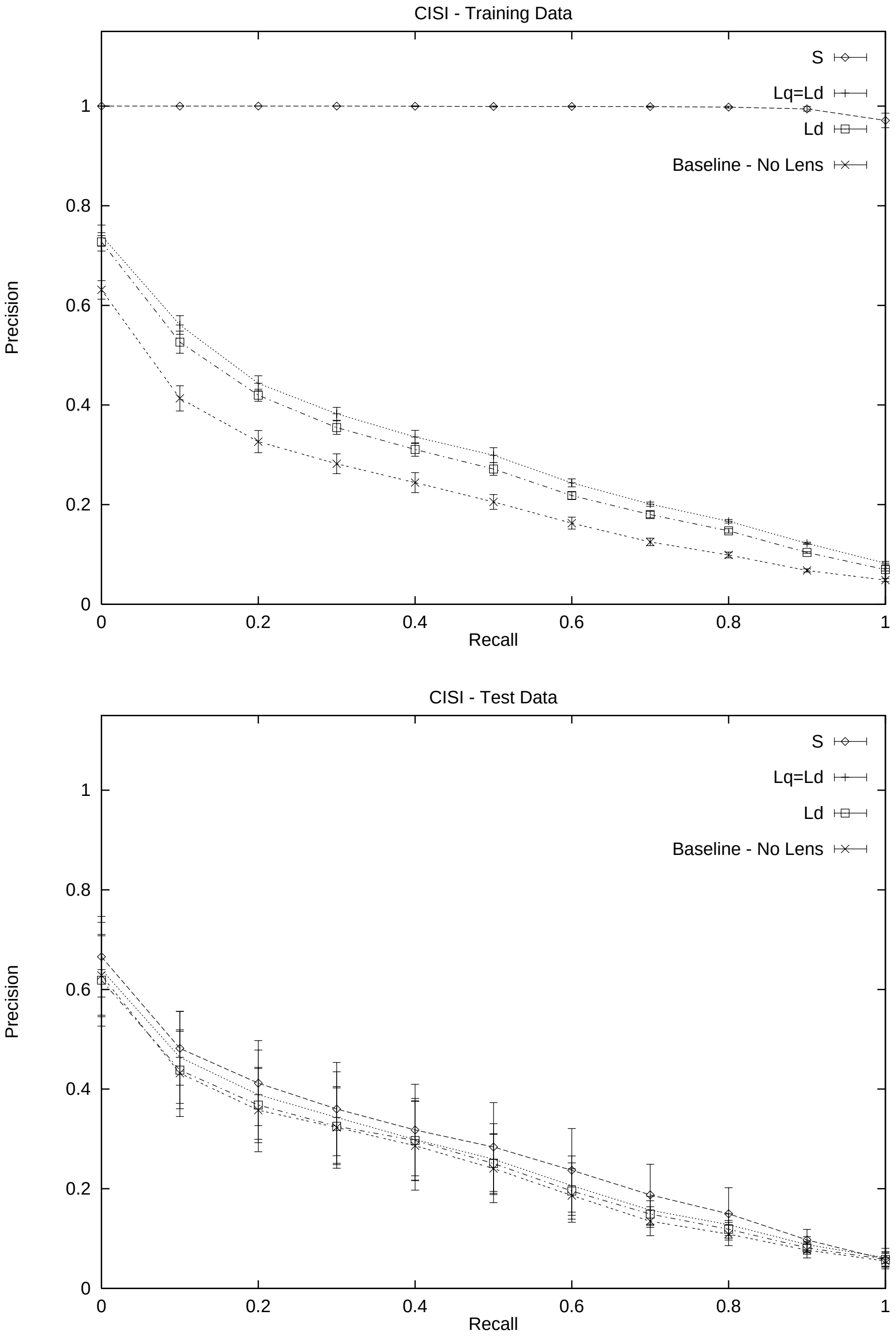

Figure 4. Results of configuration experiment, averages over 8 runs (error bars are one standard deviation).

be a robust approach to determining IR system model parameters. This makes sense, since users typically care most about, and can provide reliable relevance feedback concerning, the *rank order* of retrieved documents rather than their absolute ranking scores. The J criterion is sensitive to just this nonmetric information. Furthermore, we note that optimizing J can be adopted by any adaptive approach to IR, not just user lenses.

Our use of J to adjust the entries in a lens matrix, incorporating a user's preferences, provides a method for improving any existing vector-space weighting scheme. Any document or query, whether part of the original corpus or training set or new and unanticipated, can be "warped" by the lens into a representation more consistent with those they have successfully used in the past. User lenses are also compatible with more traditional forms of feedback, since they can be applied directly to the reweighted or term-expanded form of a query. In terms of the two canonical IR tasks, this suggests improved performance on both the adhoc and routing tasks.

5 Future Work

The work reported here is in its earliest stages and a great deal remains to be done, ranging from straight-forward extensions of experiments done here to much larger designs. First, additional testing must be done to confirm our interpretation of the enormous advantage of training the full S matrix. Second, all of our experiments to date have used more analytically tractable *linear* transformations, but much of this work has grown out of an interest in nonlinear, neural network-style learning methods. We continue to believe that additional benefits may be gained by such nonlinear transformations, and intend to explore these extensions as well. Important questions also remain with regards to the capacity of the lens. That is, how many documents and queries can we reasonably expect the lens to be able to accurately capture?

We are currently in the process of reproducing these results on the the TREC data set (Harman, 1997), and expect that our techniques will scale up to this much larger corpus. We are also formulating a series of experiments that verify that a lens can improve (or at least not degrade) the performance of a routing system.

Ultimately, we envision multiple varieties of lenses, some corresponding to individual users and some to *groups* of users with shared, consensual interpretations of what words and documents mean. Important issues remain concerning how lenses can be trained independently, how users' collective relevance feedback can be aggregated in the appropriate lens, and how compound lenses might be formed automatically, to shape the world of documents each user sees.

References

Bartell, B. T., Cottrell, G. W., and Belew, R. K. (1995). Representing documents using an explicit model of their similarities. *Journal of the American Society for Information Science* 46(4).

Bartell, B. T. (1994). *Optimizing Ranking Functions: A Connectionist Approach to Adaptive Information Retrieval*. thesis, Department of Computer Science and Engineering, The University of California, San Diego, CSE 0114, La Jolla, CA 92093.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41(6):391–407.

Harman, D. K., ed. (1997). *The Fifth Text REtrieval Conference (TREC5)*. Gaithersberg, MD: National Institute of Standards and Technology. NIST Special Publication 500-238.

Salton, G., and Buckley, C. (1988). Term weighting approaches in automatic text retrieval. *Information Processing and Management* 24:513–23.

Salton, G., ed. (1971). *The SMART Retrieval System – Experiments in Automatic Document Retrieval*. Englewood Cliffs, N.J.: Prentice-Hall Inc.

Vogt, C., Cottrell, G., Belew, R., and Bartell, B. (1997). Using relevance to train a linear mixture of experts. In Harman, D. K., ed., *The Fifth Text REtrieval Conference (TREC5)*, 503–515. Gaithersberg, MD: National Institute of Standards and Technology. NIST Special Publication 500-238.