A Comparison of Classifiers and Document Representations for the Routing Problem

Hinrich Schütze* David A. Hull† Jan O. Pedersen*

*Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304, USA
{schuetze,pedersen}@parc.xerox.com

†Rank Xerox Research Center
6 Chemin de Maupertuis
38240 Meylan, France
hull@xerox.fr

URL: ftp://parcftp.xerox.com/pub/qca/SIGIR95.ps

Abstract

In this paper, we compare learning techniques based on statistical classification to traditional methods of relevance feedback for the document routing problem. We consider three classification techniques which have decision rules that are derived via explicit error minimization: linear discriminant analysis, logistic regression, and neural networks. We demonstrate that the classifiers perform 10-15% better than relevance feedback via Rocchio expansion for the TREC-2 and TREC-3 routing tasks.

Error minimization is difficult in high-dimensional feature spaces because the convergence process is slow and the models are prone to overfitting. We use two different strategies, latent semantic indexing and optimal term selection, to reduce the number of features. Our results indicate that features based on latent semantic indexing are more effective for techniques such as linear discriminant analysis and logistic regression, which have no way to protect against overfitting. Neural networks perform equally well with either set of features and can take advantage of the additional information available when both feature sets are used as input.

1 Overview

Document routing can be described as a problem of statistical text classification. Documents are to be assigned to one of two categories, relevant or non-relevant, and a large sample of judged documents is available for training. This paper will compare traditional relevance feedback approaches to routing with classification based on explicit error minimization.

A central problem in routing is the high dimensionality of the native feature space, where there exists one potential dimension for each unique term found in the collection, typically hundreds of thousands. Standard classification techniques cannot deal with such a large feature set, since computation of the solution is not tractable and the results become unreliable due to the lack of sufficient training data. One solution is to reduce dimensionality by using subsets of the original features or transforming them in some way. Another approach does not attempt dimensionality reduction, but instead employs a learning algorithm without explicit error minimization. Relevance feedback via Rocchio expansion, which has been widely used in IR, is an example of such an approach. We will examine two different forms of dimensionality reduction, Latent Semantic Indexing (LSI) and optimal term selection, in order to investigate which form of dimensionality reduction is most effective for the routing problem.

In routing, the system uses a query and a list of documents that have been identified as relevant or not relevant to construct a classification rule that ranks unlabeled documents according to their likelihood of relevance. We examine a number of different methods of generating the document classifier: relevance feedback via query expansion (QE), linear discriminant analysis (LDA), logistic regression (LR), linear neural networks (LNN), and non-linear neural networks (NNN). The mathematical description of the classification rule is generally expressed as a function $f(x)$, where x is a vector of feature variables. The traditional approach to relevance feedback [30] defines $f(x) = q * x$, where q, the feedback query, is a weighted combination of the original query vector and the vectors of the relevant (and perhaps non-relevant) documents. Methods which use this functional form (QE, LDA, LR, and LNN) are known as linear classifiers. We also look at NNN's to investigate whether adding a non-linear component to the basic model improves performance.

The classification techniques proposed above have significant advantages over query expansion. They perform explicit error minimization using an underlying model with enough generality to take full advantage of the information contained in a large sample of relevant documents. In contrast, query expansion uses a limited probabilistic model that assumes independence between features and the model parameters are often fit in a heuristic manner based on term frequency information from the corpus. This paper will demonstrate that these advantages translate directly into improved retrieval performance for the routing problem. We use the Tipster collection and the TREC-2 and TREC-3 routing tasks to test classifiers and representations [15, 16].

There are some risks associated with using more general models of the relevant document space. On the surface, one might expect that learning algorithms that use more parameters and/or a larger feature space will have an easier time capturing the distinction between relevant and non-relevant documents (cf. Buckley's recent experiments that show better performance with increasing number of terms [4]). However, the improved performance is only guaranteed for the training data, which is simply a sample from the underlying population of relevant documents which may not adequately characterize its true distribution. The more general the model, the more effort it will expend on fitting to specific features of the training documents that will generalize to the full relevant population. A classification technique is said to suffer from *overfitting* when it improves performance over the training documents but reduces performance when applied to new documents, when compared to another method. There is thus a fundamental trade-off between a large feature space with a restrictive learning algorithm and fewer features with a more general learning algorithm. In the past [15], evidence has suggested that a weak learning rule (query expansion) and a high-dimensional feature space (terms) optimizes performance. We will demonstrate that the alternative approach is likely to prove superior in the long run.

Sections 2 and 3 describe and motivate our dimensionality reduction strategies and classification techniques. Sections 4 and 5 present experimental set-up and experimental results. Section 6 analyzes results in detail and section 7 states our conclusions.

2 Dimensionality Reduction

In our work, we will examine two major approaches to dimensionality reduction, loosely described as feature selection and reparameterization. In feature selection, a subset of the most important features are selected from the full feature space for use by the learning algorithm. Most previous work on classification in IR has relied exclusively on this method of dimension reduction. Reparameterization is the process of constructing a new document representation by taking combinations and transformations of the original feature variables.

In our experiments, the most important features are assessed by applying a χ^2-measure of dependence to a contingency table containing the number of relevant and non-relevant documents in which the term occurs (N_{r+} and N_{n+}, respectively), and the number of relevant and non-relevant documents in which the term doesn't occur (N_{r-} and N_{n-}, respectively).

$$\chi^2 = \frac{N(N_{r+}N_{n-} - N_{r-}N_{n+})^2}{(N_{r+} + N_{r-})(N_{n+} + N_{n-})(N_{r+} + N_{n+})(N_{r-} + N_{n-})}$$

For each query, only documents in the *local region* are considered (see below). We settled on the χ^2-statistic as the selection criterion after initial experiments comparing it with term selection according to raw frequency of occurrence and according to the ratio of relevant and non-relevant documents a term occurs in. These alternative measures were less effective than the χ^2-test. The underlying assumption in using the χ^2-test is that features whose frequency depends heavily on whether they occur in a relevant or non-relevant document (defined by a high χ^2-score for constant total frequency) will be useful for measuring the distinction between these two categories.

For reparameterization, we use Latent Semantic Indexing (LSI) [8], a technique that represents features and documents by a low-dimensional linear combination of orthogonal indexing variables. Our use of LSI differs in two important aspects from [8]. We compute a separate representation of terms and documents for each query, focusing on the documents which are most likely to be relevant [19]. We refer to this technique as *Local LSI* since it is only applied to a region of the document space that is in the neighborhood of the query. A second innovation is that the LSI representations are not used to construct a query which is analyzed using the vector space model. Rather, they are used as input parameters to a learning algorithm [19, 34].

LSI works by applying a matrix decomposition to the term by document matrix of the collection, which generates a large number of orthogonal LSI factors. A small number of the most important factors are then selected to approximate the covariance structure of the full collection. We use SVDPACK, a sparse SVD algorithm, for our computations [3]. Even though this algorithm does not need to calculate all orthogonal factors, it is still difficult to compute the LSI solution for the TREC collection, since it contains over a million terms and documents.

A proper model of the full TREC collection would probably require many hundreds of LSI factors, far more than could be successfully modeled by learning algorithms. Furthermore, these factors are capturing the structure of the document collection as a whole and are not tuned for particular queries. Previous work has shown that LSI is more successful when applied to a local region on a query specific basis [19]. Dumais [9] also applies LSI to the routing task, but uses the judged documents for all the queries to generate her reduced representation, a method that corresponds roughly to taking the union of the local LSI regions for each query.

We compute a separate LSI representation for each query using only the documents contained in the *local region* (defined in the next section), retaining the 100 most important factors.[1] These factors should capture the most important local structure, which will be crucial in separating relevant documents from nearby non-relevant documents. This approach differs from the one in [19] in that the local region now contains both relevant and non-relevant documents, which was found to be more effective than using only relevant documents [20].

2.1 Discussion

LSI captures the "theme" (or "latent semantic structure" [8]) of a document by analyzing the patterns of cooccurrence between terms.[2] Focusing on the theme of a document addresses the problems of synonymy and near-synonymy: In a term-based representation scheme, documents that are about the same theme but describe it with different vocabulary are represented in a way that hides their thematic similarity. This makes it difficult to obtain an accurate measurement of relevance. LSI avoids some of this problem by representing the theme of a document rather than specific terms.

At first sight, synonymy seems a minor problem in the routing context where a training set is available. A classifier can be trained to recognize that each of several different ways of expressing a particular theme indicates relevance. Indeed, if there are a few terms that provide reliable evidence for estimating relevance, then the use of LSI is not necessary. For example, consider a mail filter for TREC topic 133, which is about the Hubble Space Telescope. It can do an excellent job by relying on the single term "Hubble", and an LSI analysis will make it more difficult for a classifier to get at the correct information, the presence or absence of the term "Hubble".

However, if there is a great number of terms which all contribute a small amount of critical information, then the combination of evidence is a major problem for a term-based classifier. Consider the example of another TREC query, Topic 124 about "Alternatives to Traditional Cancer Therapies". There are articles about many different alternative cancer therapies in the Tipster collection: gene therapy, immunization, vitamin A therapy, umbilical blood transfusion, etc. Each therapy has terms that are unique to it, so that the joint vocabulary of relevant terms is too large for a learning algorithm based on error minimization (given the small number of positive examples typical in Tipster). There are more than a thousand terms that contribute helpful information, for example "carcinogenesis" and "terminally-ill" have ranks 1010 and 1018, respectively.[3]

Therefore, LSI serves as a means of data compression, capturing the important information contained in a large number of terms with a much smaller number of factors. This is particularly useful for eliminating the redundancy in word features that is due to term dependence, since LSI factors are constructed to be orthogonal. By creating a compact representation of documents, LSI reduces overfitting while still modeling the important structure contained in heterogeneous queries like topic 124 just described.

[1] These computations took less than 5 minutes per query.

[2] We use "theme" rather than "topic" to avoid confusion with the TREC queries which are also called topics.

[3] However, it is difficult to assess by intuition only how useful a given term is. For example, "carcinogenesis" could be perfectly correlated with a term higher up in the list, in which case it would not contribute information.

Non-linear term-based classifiers can also detect dependencies and are an alternative to the particular analysis of term correlations performed by LSI. However, if the amount of training data is comparatively small, a more general classifier may fail to model non-linear dependencies correctly. In our experiments, the more complicated models we have tested don't achieve any gain in performance compared to LSI.

The disadvantage of LSI is that the full discriminatory power of some of the underlying terms may be lost for queries that crucially depend on particular highly informative terms. Term-based methods excel for this kind of query, for example the above mentioned TREC Topic 133 on the Hubble Space Telescope. Our experiments will compare the performance of features based on variable selection to those generated by Latent Semantic Indexing and determine which are more effective for learning algorithms.

3 Learning Algorithms

Previous approaches to routing and text categorization [24] have used classification trees [33, 22], Bayesian networks [6], Bayesian classifiers [22, 23], rules induction [1], nearest-neighbor techniques [25, 36], logistic regression [5], least-square methods [11], discriminant analysis [19], and neural networks [32, 34]. The majority of these algorithms require that the number of feature variables be restricted in some way. The issue of how best to accomplish this dimensionality reduction is one that has been neglected in the research on learning algorithms in information retrieval.

We compare three different classification algorithms, linear discriminant analysis, logistic regression, and neural networks to a baseline constructed by query expansion. The baseline classification vector q is the vector sum of the relevant documents, using conventional term weighting and document normalization strategies. This is equivalent to Rocchio expansion when one assigns a weight of zero to the query and the non-relevant documents. In previous experiments, we found no evidence that negative feedback improved performance.

The other classification rules are obtained by error minimization of an explicit underlying model, but use different models and optimization techniques. LDA can be derived from a normal model for the distribution of relevant and non-relevant documents in feature space (although that is not how it is derived here) and models feature dependence explicitly by using the covariance matrix of each document class. It has a closed form solution that it obtained by inversion of the covariance matrix, as described below. Logistic regression and linear NN's are based on a binomial model of document relevance, which has an iterative solution obtained via numerical optimization. Logistic regression uses the Newton-Raphson technique while neural networks rely an backpropagation (gradient descent).

3.1 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) for the two-group problem can be derived as follows [13]. Suppose that one has a sample of data from two groups with n_1 and n_2 members, with mean vectors $\bar{x}_1$ and $\bar{x}_2$ and covariance matrices S_1 and S_2 respectively. The goal is to find the linear combination of the variables that maximizes the separation between the groups. A reasonable optimization criterion is to maximize the separation between the vector means, scaling to reflect the structure in the pooled covariance matrix. In other words, choose a so that: (T stands for transpose)

$$a^* = \underset{a}{\arg\max} \frac{a^T(\bar{x}_1 - \bar{x}_2)}{\sqrt{a^T S a}}$$

is maximized, where $(n_1 + n_2 - 2)S = (n_1 - 1)S_1 + (n_2 - 1)S_2$.

Since S is positive definite, we can define the Cholesky decomposition of $S = R^T R$. Let $b = Ra$, then the formula above becomes:

$$\underset{b}{\arg\max} \frac{b^T R^{T-1}(\bar{x}_1 - \bar{x}_2)}{\sqrt{b^T b}}$$

which is maximized by choosing $b \propto R^{T-1}(\bar{x}_1 - \bar{x}_2)$, which means then that $a^* = R^{-1} b = S^{-1}(\bar{x}_1 - \bar{x}_2)$. Therefore, the one dimensional space defined by $y = a^{*T} x$ should cause the group means to be well separated. This approach can be generalized to more than two groups and it can be extended to create a non-linear classifier by modeling a separate covariance matrix for each group. LDA has already been applied to the routing problem by Hull [19].

In order to produce a non-linear classifier, one can estimate a separate covariance matrix for each group, rather than using a pooled estimate of the covariance matrix S, an approach known as Quadratic Discriminant Analysis (QDA). However, QDA is only effective when the number of elements in each group is significantly larger than the number of feature variables, which is almost never the case for the routing problem because relevant documents are relatively rare.

There is a more well-behaved alternative known as Regularized Discriminant Analysis (RDA) [10]. RDA uses a pair of shrinkage parameters to create a very general family of estimators for the group covariance matrices. Rather than choosing between the pooled (LDA) and unpooled (QDA) covariance matrices, it looks at a weighted combination of them. RDA selects the optimal values for the shrinkage parameters based on cross-validation over the training set. However, previous experiments have not found much benefit to applying RDA to the routing problem [20].

3.2 Logistic Regression

Logistic regression is a statistical technique for modeling a binary response variable by a linear combination of one or more predictor variables, using a logit link function:

$$g(\pi) = log(\pi/(1 - \pi))$$

and modeling variance with a binomial random variable, i.e., the dependent variable $log(\pi/(1-\pi))$ is modeled as a linear combination of the independent variables. The model has the form $g(\pi) = x_i\beta$ where π is the estimated response probability (in our case the probability of relevance), x_i is the feature vector for document i, and β is the weight vector which is estimated from the matrix of feature vectors. The optimal value of β is derived using maximum likelihood [26] and the Newton-Raphson method of numerical optimization.

Logistic regression has been used for text retrieval in previous experiments [5, 12, 32]. Our approach is similar but all our feature variables are query-specific, i.e. we do not make use of general properties that are common to all queries in the collection. For the document routing problem, where large quantities of training documents are available for each query, such information is likely to be of limited value.

3.3 Neural Networks

A neural network (NN) is a network of units, some of which are designated as input and output units. Neural networks are trained by backpropagation: the activation of each input pattern is propagated forward through the network, and the error produced is then backpropagated and the parameters changed so as to reduce the error [28].

The strength of neural networks is that they are robust, i.e., they have the ability to fit a wide range of distributions accurately. For example, any member of the exponential family can be modeled

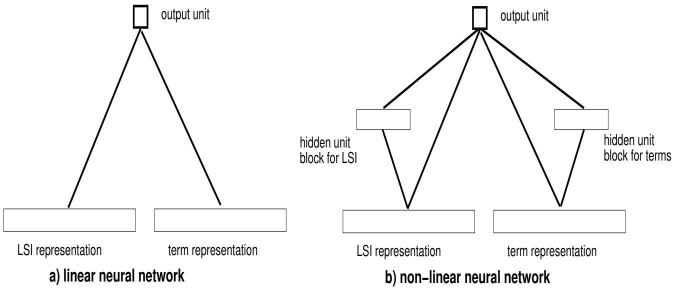

Figure 1: Linear and non-linear neural network.

[29]. Unfortunately, this capacity leads to the danger of *overfitting*. Neural networks can produce a model which fits the training data too precisely and does not generalize to the full population. In previous experiments, we found that logistic regression performed poorly when used with large numbers of features variables, and the most likely culprit is overfitting.[4]

Our neural networks protect against overfitting by using a validation set. Two thirds of the training data is used for model selection, while the remaining third is set apart for validation. At each iteration, the parameters of the model are updated and the error on the validation set computed. Training continues until the error on the validation set goes up, which indicates that overfitting has set in. This procedure establishes the number n of iterations of training that improve generalization. The final parameters of the model are then computed by training on the entire training set for n iterations. We chose this procedure rather than systematic cross-validation since the latter would have been computationally too expensive.

For the validation procedure described above, it is useful to have an optimization strategy that changes the parameters by small amounts at each iteration so that it does not overshoot the optimal point and overfit the training data. Backpropagation (gradient descent), as implemented in our neural networks, acts in just this fashion.

The architectures of the neural networks used in our experiments are shown in Figure 1. There is only one output unit whose activation models probability of relevance. The linear network consists only of input and output units. The non-linear network additionally has two blocks of 3 hidden units each of which are connected to both input and output units. (The figure shows the network architectures for dual input (LSI and terms). The architectures with only one input realize only the corresponding half of the architectures.) In both architectures, all input units are directly connected to the output unit. Relevance for a document is computed by setting the activations of the input units to the document's representation and propagating the activation through the network to the output unit, then propagating the error back through the network, using a gradient descent algorithm [28].

We chose the sigmoid:

$$f(x) = \frac{e^x}{1 + e^x}$$

as the activation function f for the units of the network, It can be shown [29] that in this case backpropagation minimizes the same error as the logistic regression, the cross-entropy error:

$$L = -\sum_i t_i \log o_i + (1 - t_i) \log(1 - o_i)$$

[4] Table 1 confirms this result: precision for logistic regression decreases when more features are added.

where t_i is the relevance for document i and o_i is the estimated relevance (or activation of the output unit) for document i. The definition of the sigmoid is equivalent to $x = log(f(x)/(1 - f(x)))$, which is the same as the logit link function. This means that linear neural networks (architecture (a) in Figure 1) and logistic regression both perform maximum likelihood estimation of the same model. The main difference lies in the optimization algorithm, Newton--Raphson for the logistic regression and backpropagation for neural networks.

Apart from gradient descent, another difference between logistic regression and neural networks is that the latter have a non-linear extension (architecture (b) with hidden units in Figure 1). Hidden units can be interpreted as feature detectors that estimate the probability of a feature being present in the input. This estimate is then propagated to the output unit and can contribute to a better estimate of relevance.

We focus on the learning aspect of neural networks, in particular explicit error minimization. In contrast, other work on neural networks in IR has been closely related to the vector space model [35] or relevance feedback [2]. Kwok's work in [21] bears most similarity with our approach. However, apart from the standard learning algorithm we use, our input consists of reduced representations (either by feature selection or reparameterization). This representational scheme substantially reduces training time, and is less prone to overfitting, because there are fewer parameters. An interesting innovation of Kwok's approach that we are planning to integrate into our model is the non-random initialization of weights, which reflects prior knowledge about terms and documents.

In summary, there are two reasons why we use neural networks as a statistical technique for routing. First we would like to protect against overfitting. Linear neural networks and logistic regression have the same probabilistic model, but validation combined with gradient descent (used to train neural networks) is better suited to avoid overfitting. Secondly, we would like to explore the use of non-linear classifiers in routing. In analogy to the way that non-linear RDA generalizes linear LDA, linear neural networks have a simple non-linear extension: neural networks with hidden units, corresponding to feature detectors.

4 Experimental Set-Up

We use the Tipster corpus for our experiments. It consists of 3.3 gigabytes of text in over one million documents from several different sources: newswire, patents, scientific abstracts, and the Federal Register [14]. There are also 200 Tipster queries, detailed statements of information need that are called *topics*.

We preprocess the corpus using the TDB system [7], performing document parsing, tokenization including stemming using a two-level finite-state morphology, and removal of terms from a 951 word stop-list. Our *terms* consisted of single words and two-word phrases that occur over five times in the corpus (where phrase is defined as an adjacent word pair, not including stop words). This process produced over 2.5 million terms. We also break up documents into chunks of about 250 terms, called text-tiles [17]. Only the tile with the highest proximity to the topic (i.e. the highest correlation in the vector space model) is selected and used for all subsequent experiments (both in training and test).

For our routing runs, we replicate the routing setup at the second and third TREC conferences. Disks 1 and 2 (about two gigabytes) are the training set for our run, Disk 3 (about one gigabyte) is the test set. Each combination of classifier and input representation is run for two sets of topics: 51–100 (corresponding to the routing task in TREC 2 [15]) and 101–150 (corresponding to the routing task in TREC 3 [16]). Our goals in these experiments are (1) to demonstrate that classification techniques work better than query expansion, (2) to find the most effective classification technique for the

classifier	input	precision for Topics 51–100			precision for Topics 101–150			average
		average	% change	at 100	average	% change	at 100	change
baseline	expansion	0.3678	+0.0%	0.4710	0.3705	+0.0%	0.4194	+0
	LSI	0.3240	-11.9%	0.4210	0.3268	-11.8%	0.3908	-12
	200 terms	0.3789	+3.0%	0.4824	0.3712	+0.2%	0.4440	+2
	LSI + 200 terms	0.3359	-8.7%	0.4426	0.3358	-9.4%	0.3928	-9
logistic regression	LSI	0.3980	+8.2%	0.5108	0.4057	+9.5%	0.4802	+9
	200 terms	0.3654	-0.7%	0.4788	0.3637	-1.8%	0.4434	-1
	LSI + 200 terms	0.3494	-5.0%	0.4652	0.3457	-6.7%	0.4168	-6
LDA	LSI	0.4139	+12.5%	0.5166	0.4230	+14.2%	0.4870	+13
	200 terms	0.3966	+7.8%	0.4916	0.3841	+3.7%	0.4586	+6
	LSI + 200 terms	0.3973	+8.0%	0.5034	0.3910	+5.5%	0.4616	+7
linear network	LSI	0.4098	+11.4%	0.5094	0.4211	+13.7%	0.4830	+13
	200 terms	0.4209	+14.4%	0.5044	0.4121	+11.2%	0.4742	+13
	LSI + 200 terms	0.4273	+16.2%	0.5180	0.4302	+16.1%	0.4908	+16
non-linear network	LSI	0.4110	+11.7%	0.5090	0.4208	+13.6%	0.4834	+13
	200 terms	0.4210	+14.5%	0.5026	0.4115	+11.1%	0.4740	+13
	LSI + 200 terms	0.4251	+15.6%	0.5204	0.4318	+16.5%	0.4882	+16

Table 1: Non-interpolated average precision, precision at 100 documents and improvement over expansion for routing runs on TREC data.

routing problem, and (3) to make sure that our comparison between LSI and term-based methods is not based on the idiosyncrasies of a particular learning algorithm.

4.1 Query-specific screening of the collection

The sheer size of the TREC collection makes it difficult to apply learning methods to the full training set from a purely computational standpoint. Furthermore, all documents are not of equal value for training. Relevant documents are relatively rare, which means that they are much more valuable for training than non-relevant documents. These considerations motivate an initial screening of documents before applying our classification algorithms.

For each query, we apply an initial screening process designed to identify documents that are clearly not relevant so that they can be excluded from further analysis. We define the local region for a query as the 2000 nearest documents, where similarity is measured using the inner product score to the Rocchio-expansion of the initial query vector [4], corresponding to our baseline feedback algorithm.[5] The documents in the local region are then used as the training set for the learning algorithms. The documents in this region for which relevance judgements do not exist are treated as not relevant.

There are a number of advantages to training over the local region. First, the size of the training set is substantially reduced, so it is possible to attack the problem using computationally intensive learning algorithms. Second, the density of relevant documents is much higher in the local region than in the collection as a whole. Third, the non-relevant documents selected for training are those which are most difficult to distinguish from the relevant documents. These non-relevant documents are clearly among the most valuable ones to use as training data for a learning algorithm.

The screening process is also applied to the test set before evaluation to avoid extrapolating beyond the region defined by the training set. A threshold derived from the training set is applied to all documents in the test set. Documents with a query-correlation higher than the threshold are automatically ranked ahead of those that fall outside the local region.

[5] Only the one-thousand highest-weighted terms were used which may partly explain why our performance is not as good as the one in [4].

5 Experimental Results

Table 1 presents routing results for 5 different classifiers and 4 different representations. The representations are:

a) relevance feedback via query expansion

b) LSI (100 factors from a query-specific local LSI)

c) 200 terms (200 highest ranking terms according to χ^2-test)

d) LSI + terms (100 LSI factors and 200 terms).

The classifiers are:

a) baseline (the vector space model: documents ranked according to proximity to query vector for "LSI", "200 terms", and "LSI + 200 terms" and proximity to expanded query vector "for expansion")

b) logistic regression

c) linear NN (architecture (a) in Figure 1)

d) non-linear NN (architecture (b) in Figure 1)

e) LDA (linear discriminant analysis)

The run "expansion" was tf-idf weighted [31], and terms in the baseline runs were idf-weighted. Inverse document frequency (idf) weights are derived from the entire training set, not from the local region. All other runs on terms were not weighted: the input was 1 if the term occurred in the document and 0 otherwise. This strategy was motivated by poor results for runs in which terms were weighted according to frequency of occurrence and a desire to let the learning algorithms select the proper weight for each term.

These experimental results are analyzed using ANOVA and the Friedman Test [18] to measure their statistical significance. ANOVA determines that one method is significantly better than another if the average difference in performance is large compared to its variability, correcting for differences between queries. The Friedman test conducts a similar analysis, but it uses only the ranking of the methods within each query.

From Table 1 we can draw the following conclusions:

Classification vs. Expansion. More advanced learning algorithms increase performance by 10 to 15 percent over query expansion. LDA and neural networks perform significantly better than the baseline experiments, regardless of representation. Logistic regression only performs better when using an LSI representation (significant difference $\approx$ .02).

LSI vs. Selected terms. LDA and logistic regression work significantly better with LSI features than with term features. Neural networks work equally well with either LSI or term-based features, and significantly better with a combination of LSI and term-based features (significant difference $\approx$ .01).

Logistic Regression vs. Other Classifiers. For LSI features, logistic regression is less effective than the other learning algorithms according to the Friedman Test, although the magnitude of the difference is small. For word or combined features logistic regression performs a lot worse than either LDA or neural networks.

Linear vs. Non-linear neural networks. The results suggest that there is no advantage to adding non-linear components to the neural network. (see Section 6 for discussion)

LDA vs. Neural networks. For LSI features, LDA and neural networks perform about the same. Neural networks are superior to LDA for the other representations. The best neural network performance (combined features) is slightly better than the best LDA performance (LSI features), but not enough to be statistically significant.

The sharp observer will note that the magnitude of the significant difference changes, depending on the experiment. This occurs because the variability between learning algorithms is greater than the variability between representations. Therefore, comparisons between experimental runs using the same learning algorithm can detect the significance of a smaller average difference.

The most important conclusion is that advanced learning algorithms capture structure in the feature data that was not obtained from query expansion. It is also interesting that the linear neural network works better than logistic regression, since they are using exactly the same model. This indicates that the logistic model is overfitting the training data, and the ability of the neural network to stop training before convergence is an important advantage. NN's can also benefit from the additional information available by combining the word and LSI features unlike the other classification techniques. Evidence of overfitting for logistic regression can be found by observing that performance decreases when going from LSI or term features to a combined representation. Using a more general feature space should only increase performance over the training set, yet it hurts performance in the final evaluation. The price for better protection against overfitting in neural networks is their slower speed of convergence, since backpropagation (gradient descent) requires more time to converge than Newton-Raphson.

Linear discriminant analysis also suffers from overfitting, which explains why it works most successfully with the compact LSI representation. One might be able to improve performance for word-based features by applying regularized discriminant analysis [10], which uses cross-validation to adjust for this problem. However, we did not conduct such an experiment here, due to the prohibitive computational cost of cross-validation for large IR problems. Previous work [20] suggests that RDA does not improve performance when applied to the LSI representation. To the best of our knowledge, the results given here for LDA and neural networks are at least as good as the best routing results published for TREC-2 [4] and TREC-3 [27].

Selection of the best routing technique in an operational system may depend on efficiency as well as IR performance. When computed using a Sparc 10, the neural network solution requires 3 hours per query, logistic regression requires 2-10 minutes per query, LDA requires 0.5-5 minutes, and query expansion (limited to 1000 terms) requires considerably less than a minute. This does not include the time to compute the LSI solution which is less than 5 minutes. However, there are several other important factors. One generally assumes that the routing query is a standing profile which can be computed once in advance, and is not subject to the same time constraints which apply to other search problems.

The experimental set-up of the TREC routing problem is unusual in that all the relevance judgements in the training set are presented initially rather than coming in gradually over time. Iterative algorithms (and query expansion) are well-equipped to deal with new training data as the new solution can be computed from the previous optimal setting of the parameters, and convergence times should be much reduced. There also exist updating algorithms which can be used to compute a revised solution for linear discriminant analysis. However, the LSI solution must be recomputed from scratch, and it is unclear how neural networks would protect against overfitting in this context.

6 Query Analysis

While the average performance scores presented in the previous section are quite informative, they do not provide a complete picture of the experimental results. Similar average scores can conceal large differences in performance for individual queries. In this section, we examine the experimental results in more detail on a query by query basis in order to gain a better understanding of the observed differences between methods and representations.

We focus on three specific issues. First, when do our classification techniques perform better (or worse) than relevance feedback via query expansion? Second, does the optimal choice of representation depend on some characteristic of the query? Third, while linear and non-linear neural networks perform equally well on average, perhaps there are individual queries where non-linearity can be helpful.

Query expansion vs. Linear neural network. Table 2 examines the difference between query expansion and the linear neural network with terms as input; and presents the queries with the largest differences between the two methods. The neural network performs better than expansion in 71 of the 100 queries with an average improvement of .047. Note that despite the high standard deviation of .090, the average difference between expansion and the neural network (as well as LDA) is significant according to both ANOVA and Friedman test.

We hypothesized that the queries where expansion was more successful than learning algorithms might be ones where the use of feature selection resulted in a loss of information. We tested this hypothesis by looking at the baseline scores for these queries using expansion and word based features. However, there was no correlation between poor performance of the neural networks and poor performance of the feature selection algorithms. So far, we have been unable to find any patterns that indicate which characteristics of the query (or its relevant documents) make it more (or less) amenable to learning algorithms.

LSI vs. Term Features. Table 3 compares performance of the linear neural network for LSI and terms. The queries with the largest differences between the two methods are presented. Average precision for LSI is better for 56 queries and worse for 39 queries with 5 ties. Although there is virtually no difference in average performance (-0.0010), the differences for individual topics are large: There are 24 topics with a difference of more than 5%.

We analyzed the top ten documents of four of the topics (51, 133, 72, 134) for both representations to determine possible reasons for the large individual differences.

Topic 51 "Airbus Subsidies" specifies that relevant articles describe either government assistance or a dispute between a European and an American manufacturer. The term-based method did a better job at capturing this condition in the decision rule. It ranked

	TREC topic	Δ	expansion	terms
134	The Human Genome Project	0.1283	0.5833	0.4550
140	Political Impact of Islamic Fundamentalism	0.0846	0.2773	0.1927
143	Why Protect U.S. Farmers?	0.0633	0.5686	0.5053
68	Health Hazards from Fine-Diameter Fibers	0.0557	0.7203	0.6646
73	Demographic Shifts Across Boundaries	0.0390	0.4568	0.4178
92	International Military Equipment Sales	-0.1815	0.1239	0.3054
144	Management Problems at the United Nations	-0.2291	0.1334	0.3625
61	Israeli Role in Iran-Contra Affair	-0.2470	0.1871	0.4341
133	Hubble Space Telescope	-0.2914	0.3763	0.6677
51	Airbus Subsidies	-0.6363	0.2271	0.8634
	mean (100 topics)	-0.0473	0.3692	0.4165
	std. dev. (100 topics)	0.0902	0.2197	0.2159

Table 2: Query expansion vs. linear network with terms as input. 10 topics with the greatest differences in non-interpolated average precision.

	TREC topic	Δ	LSI	terms
134	The Human Genome Project	0.1501	0.6051	0.4550
72	Demographic Shifts in the U.S.	0.0945	0.4471	0.3526
124	Alternatives to Traditional Cancer Therapies	0.0923	0.5396	0.4473
136	Diversification by Pacific Telesis	0.0812	0.5204	0.4392
63	Machine Translation	0.0786	0.5642	0.4856
131	McDonnell Douglas Contracts for Military Aircraft	-0.0741	0.0494	0.1235
144	Management Problems at the United Nations	-0.1004	0.2621	0.3625
61	Israeli Role in Iran-Contra Affair	-0.1309	0.3032	0.4341
133	Hubble Space Telescope	-0.1630	0.5047	0.6677
51	Airbus Subsidies	-0.2606	0.6028	0.8634
	mean (100 topics)	-0.0010	0.4155	0.4165
	std. dev. (100 topics)	0.0519	0.2138	0.2159

Table 3: Linear network with input LSI vs. selected terms. 10 topics with the most marked differences in non-interpolated average precision.

	TREC topic	Δ	non-linear	linear
61	Israel and Iran-Contra	0.0345	0.4686	0.4341
122	RDT&E of New Cancer Fighting Drugs	0.0309	0.5240	0.4931
106	U.S. Control of Insider Trading	0.0299	0.2204	0.1905
144	Management Problems at the United Nations	0.0192	0.3817	0.3625
77	Poaching	0.0144	0.5005	0.4861
139	Iran's Islamic Revolution	-0.0093	0.1861	0.1954
138	Iranian Support for Lebanese Hostage-takers	-0.0152	0.2041	0.2193
60	Merit-Pay vs. Seniority	-0.0364	0.1962	0.2326
107	Japanese Regulation of Insider Trading	-0.0375	0.2051	0.2426
124	Alternatives to Traditional Cancer Therapies	-0.0536	0.3937	0.4473
	mean (100 topics)	-0.0002	0.4163	0.4165
	std. dev. (100 topics)	0.0240	0.2152	0.2159

Table 4: Linear vs. Non-linear Neural Networks. 10 topics with the most marked differences in non-interpolated average precision.

many relevant articles higher than the LSI-based method because they contained good indicators for subsidies or trade conflicts. Examples of documents and highly weighted terms indicating subsidies or trade conflicts: AP900907-0243: u.s.-manufacturer, airbus-industrie; SJMN91-06169017: competitive-advantage; SJMN91-06176191: u.s.-aircraft, airbus-industrie. Conversely, the LSI-based classifier ranked non-relevant documents high that were about government involvement, but not about the precise kind of involvement required by Topic 51, namely subsidies. Examples include: AP900330-0107 and AP900501-0186 (corruption charges concerning one of Airbus' deals with India), and SJMN91-06320105 (Taiwan government in talks with McDonnell Douglas). Apparently, only the exact term features succeeded at differentiating different kinds of government involvement.

Relevance to Topic 133 "Hubble Space Telescope" seems to depend on a small number of highly weighted terms like "hubble-telescope" or "defect" (many articles are about the Hubble's defective mirror). Since theme-based features don't capture helpful information in this case, LSI is at a disadvantage in this example.

Topic 72 is about demographic shifts in the U.S. with economic impact. The condition that there be an economic impact of the shift can be expressed in many different ways. In particular, if there are many numbers in a text, that is a good indication for economic data. Two such articles that were ranked higher in the LSI-based scheme are SJMN91-06113204 (population growth in the San Francisco Bay Area) and AP900611-0055 (job growth in California). In general, the 200 top-ranked terms included in the term-based representation seem insufficient for this topic. For example, the following highly relevant sentence does not contain any of these terms: "The nation grew to 249.6 million people in the 1980s as more Americans left the industrial and agricultural heartlands for the South and West." Consequently, only the LSI classifier ranked the relevant document AP901227-0006 that contains it high.

For Topic 72, non-relevant articles were ranked high by the term-based classifier even if they did not mention economic consequences. The following three articles about reapportionment don't cover economic implications, but still receive high ranks of 5, 7, and 8: SJMN91-06005094 (Massachusetts loses seat), SJMN91-06293056 (Montana loses seat to California), SJMN91-06137238 (Redistricting in San Francisco Bay Area). There were not enough clues in the pool of 200 terms to make reliable decisions as to whether an article covered economics or not.

Insufficient coverage of the relevant vocabulary also seems to explain the poor performance of term-based classification for Topic 134 "The Human Genome Project". For example, document ZF32-037-119 is about a donation by Microsoft to the University of Washington. Due to a passing reference to the Human Genome Initiative, it contains many indicators that mislead the term-based classifier which gives it rank 5. The LSI-based representation captures the medical rather than microbiological theme of the article and gives it rank 31.

In summary, if there is a small number of good terms that reliably indicate relevance, term-based methods are superior to LSI since an LSI-based classifier can infer the presence of these individual terms only indirectly from the LSI features that are linear combinations of all terms. In contrast, if the number of indicators is large, than LSI is superior because it can integrate information from many terms.

Linearity vs. Non-linearity. Table 4 looks at the difference in performance between linear and non-linear neural networks. The non-linear network performs better than the linear network in 43 of the 100 queries (with 11 ties) and the average difference between the methods is basically zero. The standard deviation of the differences is only .024 and the extreme differences are relatively small. The differences are distributed in a fashion which suggests that they are only the result of noise.

To obtain further evidence, we examined the most extreme topics (61 and 124) a bit more closely. For topic 61, there are 35 relevant documents in the local region. The non-linear network ranks 19 of them higher and there are 3 ties. For topic 124, there are 68 relevant documents in the local region. The linear network ranks 32 of them higher and there are 3 ties. Neither of these results are close to being statistically significant according to the sign test. Since there is no difference in within-query performance for even the most extreme topics, it is safe to conclude that the non-linear component to the neural network provides absolutely no advantage, even for individual queries.

For large numbers of input variables, there is often no advantage to modeling non-linearity, because there is insufficient training data (even in the IR context).

7 Conclusions

In this paper, we compare two approaches to document routing, relevance feedback via query expansion and statistical classification with error minimization. We show that advanced classification algorithms perform 10-15% better than relevance feedback on the Tipster document collection. Since learning algorithms based on error minimization and numerical optimization are computationally intensive and prone to overfitting in a high dimensional feature space, it is necessary to apply some method of dimensionality reduction. We examine two different approaches, latent semantic indexing and feature selection of terms using a χ^2-test of non-independence.

Our experiments indicate that latent semantic indexing is more effective for classification techniques such as linear discriminant analysis and logistic regression, which have no way to protect against overfitting. Neural networks perform equally well with either set of features and can take advantage of the additional information available when both terms and LSI factors are used as input. We also provide evidence that non-linear extensions of the classifiers (RDA and non-linear neural networks) do not improve performance, probably because there is not enough information in the Tipster data collection to accurately learn complex models.

Past evidence [15] has suggested that a weak learning algorithm (relevance feedback) and a high-dimensional feature space (terms) optimizes performance. We interpret the results in this paper as evidence that the alternative approach, complex learning algorithms and a reduced feature space is both practical and beneficial for the routing problem.

Acknowledgments. We are indebted to Michael Berry for SVDPACK, to Marti Hearst for implementing the text-tiling algorithm, to Jerry Friedman for advice about regularized discriminant analysis, and to John Tukey for helpful comments. We would also like to thank three SIGIR reviewer for their excellent comments.

References

[1] C. Apte, F. Damerau, and S.M. Weiss. Towards language independent automated learning of text categorization models. In *Proc. 17th Int'l Conference on R&D in IR (SIGIR)*, pages 23–30, 1994.

[2] Rik K. Belew. Adaptive information retrieval: Using a connectionist representation to retrieve and learn about documents. In *Proceedings of SIGIR '89*, pages 11–20, Cambridge MA, 1989.

[3] Michael W. Berry. Large-scale sparse singular value computations. *The International Journal of Supercomputer Applications*, 6(1):13–49, 1992.

[4] Chris Buckley, Gerard Salton, and James Allan. The effect of adding relevance information in a relevance feedback environment. In *Proceedings of SIGIR '94*, pages 292–300, 1994.

[5] Wm. S. Cooper, Aitao Chen, and Fredric C. Gey. Full text retrieval based on probabilistic equations with coefficients fitted by logistic regression. pages 57–66, 1994. In [15].

[6] W. B. Croft, J. Callan, and J. Broglio. Trec-2 routing and adhoc retrieval evaluation using the INQUERY system. 1994. In [15].

[7] Douglass R. Cutting, Jan O. Pedersen, and Per-Kristian Halvorsen. An object-oriented architecture for text retrieval. In *Conference Proceedings of RIAO'91, Intelligent Text and Image Handling, Barcelona, Spain*, pages 285–298, April 1991. Also available as Xerox PARC technical report SSL-90-83.

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

[9] Susan T. Dumais. Latent semantic indexing (lsi) and trec-2. In *The Second Text REtrieval Conference (TREC-2)*, pages 105–115, 1993.

[10] Jerome H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

[11] Norbert Fuhr. Optimum polynomial retrieval funcions based on the probability ranking principle. *ACM Transactions on Information Systems*, 7(3):183–204, 1989.

[12] Norbert Fuhr and U. Pfeifer. Probabilistic information retrieval as a combination of abstraction, inductive learning, and probabilistic assumptions. *ACM TOIS*, 12(1), Jan 1994.

[13] R. Gnanadesikan. *Methods for Statistical Data Analysis of Multivariate Observations*. John Wiley & Sons, New York, 1977.

[14] Donna Harman. Overview of the first trec conference. In *Proceedings of SIGIR '93*, 1993.

[15] Donna Harman, editor. *Proceedings of the 2nd Text Retrieval Conference (TREC-2)*, 1994.

[16] Donna Harman, editor. *Proceedings of the 3rd Text Retrieval Conference (TREC-3)*, 1995. to appear.

[17] Marti A. Hearst. Multi-paragraph segmentation of expository discourse. In *Proceedings of the 32nd Meeting of the Association for Computational Linguistics*, June 1994.

[18] David Hull. Using statistical testing in the evaluation of retrieval performance. In *Proc. of the 16th ACM/SIGIR Conference*, pages 329–338, 1993.

[19] David Hull. Improving text retrieval for the routing problem using latent semantic indexing. In *Proceedings of SIGIR '94*, pages 282–289, 1994.

[20] David A. Hull. *Information Retrieval using Statistical Classification*. PhD thesis, Stanford University, 1995.

[21] K. L. Kwok. Experiment with a component theory of probabilistic information retrieval based on single terms as document components. *ACM Transactions on Information Systems*, 8(4):363–386, 1990.

[22] David Lewis and Marc Ringuette. A comparison of two learning algorithms for text categorization. In *Symposium on Document Analysis and Information Retrieval*. University of Nevada, Las Vegas, 1994.

[23] David D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *SIGIR*, pages 37–50, 1992.

[24] David D. Lewis and Philip J. Hayes. Special issue on text categorization. guest editorial. *ACM Transactions on Information Systems*, 12(3):231, 1994.

[25] Brij Masand, Gordon Linoff, and David Waltz. Classifying news stories using memory based reasoning. In *SIGIR*, pages 59–65, 1992.

[26] P. McCullagh and J.A. Nelder. *Generalized Linear Models*, chapter 4, pages 101–123. Chapman and Hall, 2nd edition, 1989.

[27] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Text Retrieval Conference 3 (preproceedings)*, 1994.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In David E. Rumelhart, James L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing. Explorations in the Microstructure of Cognition. Volume 1: Foundations*. The MIT Press, Cambridge MA, 1986.

[29] David E. Rumelhart, Richard Durbin, Richard Golden, and Yves Chauvin. Backpropagation: The basic theory. In Yves Chauvin and David E. Rumelhart, editors, *Back-propagation: Theory, Architectures, and Applications*. Lawrence Erlbaum, Hillsdale NJ, 1995.

[30] Gerard Salton and Chris Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[31] Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[32] Hinrich Schütze, Jan O. Pedersen, and Marti A. Hearst. Xerox TREC 3 report: Combining exact and fuzzy predictors. 1995. In [16], to appear.

[33] Richard M. Tong and Lee A. Appelbaum. Machine learning for knowledge-based document routing (a reprot on the trec-2 experiment). pages 253–264, 1994. In [15].

[34] Erik Wiener, Jan Pedersen, and Andreas S. Weigend. A neural network approach to topic spotting. In *Fourth Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas NV, 1995. To appear.

[35] Ross Wilkinson and Philip Hingston. Using the cosine measure in a neural network for document retrieval. In *SIGIR*, pages 202–210, Chicago, 1991.

[36] Yiming Yang. Expert network: Effective and efficient learning form human decisions in text categorization and retrieval. In *Proceedings of SIGIR '94*, pages 13–22, 1994.