Document Routing as Statistical Classification *

David Hull
Rank Xerox Research Centre
6, chemin de Maupertuis, 38240 Meylan, France
{david.hull}@xerox.fr

Jan Pedersen and **Hinrich Schütze**
Xerox Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304
{pedersen,schuetze}@parc.xerox.com

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.

The Routing Problem

Of the two classical information retrieval tasks[1] document routing is most amenable to machine learning. A fixed, standing query, and a training collection of judged documents[2] is provided and the task is to assess the relevance of a fresh set of test documents. This can clearly be approached as a problem of statistical text classification: documents are to be assigned to one of two categories, relevant or non-relevant, and inference is possible from the labeled documents. In contrast, the classical ad-hoc search problem presumes only a query and an unlabelled collection is provided.

The standard approach to document routing models document content as a bag-of-words, represented as a sparse, very high-dimensional vector, with one component for each unique term in the vocabulary (Salton, Wong, & Yang 1975). Vector weights are proportional to term frequency and inversely proportional to collection frequency.[3] The general technique is to score test documents with respect to their closeness to the query (also represented a sparse, high-dimensional vector), where closeness is measured by the cosine between vectors. A modified and expanded query is learned from the training set via Rocchio-expansion Relevance Feedback (Buckley, Salton, & Allan 1994), which essentially constructs a linear combination of the query vector, the centroid of the relevant documents and, occasionally, the centroid of select irrelevant documents[4]. The net result is a scored list of test documents, which may be ranked in decreasing score order for the purposes of presentation and evaluation. Evaluation typically proceeds by averaging precision[5] at a number of recall[6] thresholds.

Rocchio-expansion Relevance Feedback employs a weak learning method. However, the application of stronger methods faces two problems: the very high dimensionality of the native feature space (typically hundreds of thousands of dimensions for moderate sized text collections), and the relatively small number of positive examples in the training collection. The first difficulty may be addressed through dimensionality reduction techniques, and the second by constraining the training domain. One key issue, then, is whether the improved classification power of strong learning methods can compensate for the loss of information due to dimensionality reduction and local analysis.

We have examined two dimensionality reduction techniques, term selection and a variation of latent semantic indexing (Deerwester *et al.* 1990), against a number of different error-minimizing classifiers: linear discriminant analysis (LDA), logistic regression

*Authors listed in alphabetic order.

[1] as defined and evaluated by the TREC conferences(Harman 1994; 1995)

[2] Actually, only a few documents are explicitly labeled, including most of the relevant documents and a few of the irrelevant documents. All other documents are implicitly assumed to be irrelevant.

[3] The exact expression varies across systems, but is typified by

$$d_i(j) = f_i(w_j) \log (N/N(w_j))$$

where $f_i(w_j)$ is the frequency of word w_j in document i, N is the total number of documents, and $N(w_j)$ the number of documents containing w_j.

[4] those explicitly judged irrelevant

[5] the proportion of relevant documents in the documents seen so far

[6] the proportion of relevant documents seen so far out of the total number of relevant documents

(LR), linear neural networks (LNN), and non-linear neural networks (NNN) (Schütze, Hull, & Pedersen 1995). These techniques offer significant theoretical advantages over Rocchio Expansion because they explicitly minimize misclassification error based on an underlying model with enough generality to take full advantage of the information contained in a large sample of relevant documents. In contrast, Rocchio Expansion assumes independence between features and fits its parameters 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 (Harman 1994; 1995).

Methodology

Conventional classification techniques must be adapted for the routing problem. Traditional learning algorithms do not scale to the native feature set, either due to computational tractability issues or because of the risk of overfitting. Similarly, the training collection is simply too large, and relevant documents too rare to efficiently learn the optimal classification rule. Therefore, we have developed a three-step algorithm to approach the routing problem.

Step 1: Local Regions

In the first stage, standard Rocchio Expansion[7] is applied to form a modified query vector. All documents in the training collection are ranked according to their similarity to the expanded query, and the top 2000 documents are selected to define the *local region*. These documents then form the training set.

There are a number of advantages to constraining the training domain in this manner. First, the size of the training set is controlled, making it possible to apply computationally intensive variable selection techniques and learning algorithms in a reasonable length of time. Second, the density of relevant documents is much higher in the local region than in the collection as a whole, which should improve classifier performance. 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.

[7] without considering irrelevant documents

Step 2: Document Representations

To reduce the dimensionality of the native feature space, which for the Tipster collection includes 2.5 million unique words and selected word pairs, we consider two separate approaches: optimal term selection and reparameterization of the document space.

The process of optimal term selection consists of identifying the words most closely associated with relevance. Our approach is to apply a χ^2 test to the 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).

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

The higher the score, the greater the association between that term and the relevant documents.[8] We select the 200 terms with the highest scores for input into our learning algorithms.

As an alternative, we apply Latent Semantic Indexing (LSI) (Deerwester *et al.* 1990) to represent documents by a low-dimensional linear combination of orthogonal indexing variables. The LSI solution is computed by applying a singular value decomposition to the sparse document by term matrix constructed from the local region. We then select the 100 most important LSI factors to serve as an alternative document representation.

One expects optimal term selection to work best if there is a relatively small specific vocabulary associated with the query while one expects LSI to performs well of the query has a very large vocabulary which can be organized into a smaller number of general concepts.

Step 3: Classification Algorithms

We have run a number of classification techniques against the reduced features computed in step 2. These are briefly described below.

Linear discriminant analysis (LDA) finds a linear combination a of the feature elements which maximizes the separation between the centroids of the relevant $\bar{x}_r$ and non-relevant $\bar{x}_n$ documents: $a = S^{-1}(\bar{x}_r - \bar{x}_n)$, where S is the pooled within-group covariance matrix. Logistic regression (LR) fits the log odds of the probability of relevance as a linear function of the features using maximum likelihood estimation. The linear neural network (LNN) (no hidden units) uses backpropagation to iteratively fit a logistic model. To protect

[8] It is also theoretically possible to have a high score if term is highly associated with *irrelevant* documents, however since irrelevant documents are so common this is not observed in practice

against overfitting, and to distinguish it from the logistic regression classifier, the network is not iterated to convergence, rather a validation set is used to determine the optimal number of training iterations. The non-linear neural net (NNN) adds a layer of hidden units to potentially capture non-linear structure.

To minimize misclassification errors, a binary classifier is simply required to estimate whether the probably of relevance is greater than one half. However, the evaluation criterion for the routing task requires the test set to be ranked. This can be achieved by scoring documents by probability of relevance. Note, this is not guaranteed to be optimal except under certain conditions, e.g. if the evaluation criterion is a linear utility function (Lewis 1995).

Summary of Routing Algorithm

We present a simple flow chart that describes the training and testing process of our routing algorithm. The training process:

- Compute 2000 nearest documents to Rocchio expanded query (local region).
- Compute LSI decomposition of local region and select 100 largest factors.
- Compute χ^2 statistic to select 200 most valuable terms in local region.
- Apply classification technique to documents in local region using the appropriate representation.

The testing process:

- Select test documents whose similarity to the Rocchio expanded query exceeds a threshold similarity score, placing it in the local region.
- Obtain LSI document vectors for the selected test documents.
- Obtain selected term representation for the selected test documents.
- Score selected test documents using the probability output from the trained classification rule.
- Rank the test documents by descending probability score. All test documents in the local region are ranked ahead of those that fall below the threshold similarity score.

Experimental Set-Up

The Tipster corpus used for our experiments consists of 3.3 gigabytes of text in over one million documents from several different sources: newswire, patents, scientific abstracts, and the Federal Register (Harman 1993). In addition, there are 200 judged Tipster topics, detailed statements of information need, that can serve a routing queries.

We preprocessed the corpus using the TDB system (Cutting, Pedersen, & Halvorsen 1991), to parse and tokenize documents, and to stem and remove stop words. Indexed *terms* consisted of single words and two-word phrases that occurred over five times in the corpus (where a phrase is defined as an adjacent word pair, not including stop words). This process produced over 2.5 million *terms*. Then, each document was partitioned into overlapping sections of average length 300 words with an average overlap of approximately 250 words. The highest scoring section of each document (with respect to the expanded query that defines the local region) was used in all further processing.

For our routing runs, we replicated the routing setup for the second and third TREC conferences. Disks 1 and 2 (about two gigabytes) are the training collection, 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 (Harman 1994)) and 101–150 (corresponding to the routing task in TREC 3 (Harman 1995)). Our goals in these experiments were (1) to demonstrate that strong learning methods can perform better than Rocchio expansion, (2) to find the most effective classification technique for the 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.

Experimental Results

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

a) Rocchio-expansion Relevance Feedback

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 (documents ranked by closeness to query vector for "LSI", "200 terms", and "LSI + 200

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.

terms" and closeness to expanded query vector "for expansion")

b) logistic regression

c) linear NN

d) non-linear NN

e) LDA (linear discriminant analysis)

The run "expansion" was tf-idf weighted, 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 (Hull 1993) 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.

Conclusions

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.

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. Linear discriminant analysis also suffers from overfitting, which explains why it works most successfully with the compact 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 (Buckley, Salton, & Allan 1994) and TREC-3 (Robertson *et al.* 1994).

References

Buckley, C.; Salton, G.; and Allan, J. 1994. The effect of adding relevance information in a relevance feedback environment. In *Proc. 17th Int'l Conference on R&D in IR (SIGIR)*, 292–300.

Cutting, D. R.; Pedersen, J. O.; and Halvorsen, P.-K. 1991. An object-oriented architecture for text retrieval. In *Conference Proceedings of RIAO'91, Intelligent Text and Image Handling, Barcelona, Spain*, 285–298. Also available as Xerox PARC technical report SSL-90-83.

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

Harman, D. 1993. Overview of the first trec conference. In *Proceedings of SIGIR '93*.

Harman, D., ed. 1994. *The Second Text REtrieval Conference (TREC-2)*. NIST.

Harman, D., ed. 1995. *The Third Text REtrieval Conference (TREC-3)*. NIST.

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

Lewis, D. 1995. Evaluating and optimizing autonomous text classification systems. In *Proc. 18th Annual Int'l ACM SIGIR Conference on R&D in IR*, 246–255.

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

Salton, G.; Wong, A.; and Yang, C. 1975. A vector space model for automatic indexing. *Communications of the ACM* 18(11):613–620.

Schütze, H.; Hull, D.; and Pedersen, J. 1995. A comparison of document representations and classifiers for the routing problem. In *Proc. 18th Int'l Conference on R&D in IR (SIGIR)*, 229–237.