# DSO at TREC-8: A Hybrid Algorithm for the Routing Task

**Hwee Tou Ng**
**Huey Ting Ang**
**Wee Meng Soon**

DSO National Laboratories
20 Science Park Drive, Singapore 118230
{nhweetou,ahueytin,sweemeng}@dso.org.sg

**Abstract**

In this paper, we describe a new hybrid algorithm that we used for the routing task at TREC-8. The algorithm combines the use of Rocchio's formula for term selection, and an improved variant of the perceptron learning algorithm for tuning the term weights. This algorithm is able to give good performance on TREC-8 test data. We also achieved a slight improvement in average uninterpolated precision by using Dynamic Feedback Optimization (DFO) as another weight tuning algorithm and combining the ranked list generated by DFO with that of perceptron.

## 1 Introduction

DSO is a first-time participant in TREC. We only participated in the routing task at the TREC-8 filtering track.

Broadly speaking, there are two popular approaches to the routing task. The first approach uses the Rocchio algorithm (Rocchio, 1971), and has its root in the information retrieval community. Recently, a number of extensions have been made to this approach. These include the use of better document representation (Singhal et al., 1996), better non-relevant document selection (Singhal et al., 1997), and Dynamic Feedback Optimization (DFO) for weight tuning (Buckley and Salton, 1995). This approach has yielded very good results and is used by a number of TREC participants, including Cornell (Buckley et al., 1998), AT&T (Singhal, 1998), and NTT DATA (Nakajima et al., 1999).

The second approach treats the routing task as supervised learning from training data and has its root in the traditional machine learning community. This approach is exemplified by the work of Xerox (Schütze et al., 1995; Hearst et al., 1996; Hull et al., 1997) in the context of the routing task in TREC, as well as most of the past research on text categorization (Apte et al., 1994; Cohen and Singer, 1996; Dagan et al., 1997; Lewis et al., 1996; Ng et al., 1997; Yang, 1999). In particular, our own previous research on perceptron learning for text categorization (Ng et al., 1997)) falls under this approach.

A natural question arises as to which of these two approaches is better at the routing task. One may wonder whether the success of the Rocchio formula is due to the selection of a good set of terms, or due to the assignment of a good set of weights. Although the machine learning approach for text categorization has reported good results, most of the work were only tested on the Reuters corpus (Lewis, 1992) but not on TREC data sets. Xerox has reported good results in TREC-4 routing using the machine learning approach (Hearst et al., 1996). However, their subsequent result at TREC-5 routing (Hull et al., 1997) was not as good as groups using the Rocchio approach. That this question is still unresolved is evidenced by a recent paper (Schapire et al., 1998) which attempted to compare the performance of the Rocchio approach with Adaboost, a recently developed learning algorithm from the machine learning community.

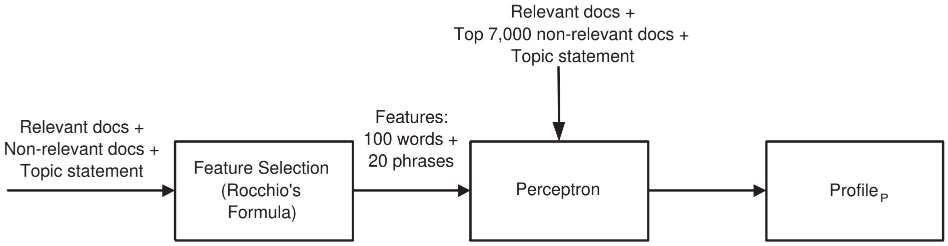


Figure 1: Training phase of our submitted run dso99rt1


Figure 2: Test phase of our submitted run dso99rt1

In this paper, we present a new hybrid algorithm used at TREC-8 for the routing task that combines these two approaches. We treat the generation of the profile for a topic as a two-step process: first select a set of terms, and then assign appropriate weights to these selected terms. Our algorithm uses Rocchio's formula to select the terms, but after the terms are chosen, the weights assigned by Rocchio's formula are discarded. It then reverts to the use of an improved variant of the perceptron learning algorithm for tuning the weights of the selected terms. Our first submitted run dso99rt1 uses this hybrid algorithm.

To further improve accuracy, our second submitted run dso99rt2 attempts to combine two weight tuning algorithms, namely perceptron and Dynamic Feedback Optimization (DFO) (Buckley and Salton, 1995). Both submitted runs start with an identical set of terms selected by Rocchio's formula, but each algorithm separately tunes the weights and two profiles are generated per topic. The final ranked list of test documents is produced by merging the individual ranked lists of the two profiles.

## 2 The First Submitted Run: dso99rt1

For each topic, our routing algorithm learns a profile, which is a set of selected terms where each term is assigned a numeric weight. A term can be a word or a phrase, where a phrase is defined as any two consecutive words in a document that are both non-stop words. Once a profile is learned, it is used to rank all test documents, using the dot product score.

Our first submitted run, dso99rt1, is generated by our new hybrid routing algorithm. This algorithm consists of two parts: feature selection and weight tuning. Feature selection picks a set of terms using Rocchio's formula, whereas weight tuning is achieved by an improved variant of the perceptron learning algorithm. The broad outline of the training and test phase of our method is shown in Figure 1 and Figure 2, respectively.

Our algorithm also incorporates recent advances made in the area of document representation (Singhal, 1998; Singhal et al., 1996) and non-relevant document selection (Singhal, 1998; Singhal et al., 1997).

- Document representation: We used the document representation scheme employed in (Singhal, 1998). Each document or topic statement can be represented in ltu, Ltu, or Lnu form. These document representation forms take into account the number of times a term appears in a document, the number of documents in the training collection that contain the term, as well as a document length normalization factor.
- Non-relevant document selection: For each TREC topic, there are many non-relevant documents available for training. A non-relevant document may be one explicitly judged as non-relevant by a human assessor, or it may be considered as non-relevant based on the "complete" judgment assumption made in TREC. To be computationally tractable as well as to give high routing accuracy, it is important to select only a good subset of non-relevant documents for training. In our algorithm, there are two places where non-relevant document selection takes place. (1) During feature selection using Rocchio's formula, the non-relevant documents are selected using the "query zone" method of (Singhal, 1998; Singhal et al., 1997). (2) In selecting the non-relevant documents for perceptron learning to tune the weights, the top 7,000 non-relevant training documents are chosen by explicitly using a learning algorithm (perceptron) to learn a classifier to rank the potential non-relevant documents.

Our TREC-8 training document collection $T$ consists of all TREC documents (minus the 1993 and 1994 Financial Times documents) that have been explicitly judged as relevant or non-relevant to any of the topics 351-400. All documents are preprocessed where stop words and punctuation symbols are removed, and the terms are stemmed using Porter's algorithm.[1]

In the remainder of this section, we describe the feature selection and the weight tuning components of our algorithm.

### 2.1 Feature Selection

This component is the same as that used in (Singhal, 1998).

#### 2.1.1 Non-relevant Document Selection

First, the topic statement is represented in ltu form. Then each training document in the training document collection $T$ is represented in Lnu form. Each training document is ranked by its dot product score with the topic statement. Let $R$ be the set of all documents judged to be relevant to a topic $i$ by the human assessors. Then $T - R$ is the set of all potential non-relevant documents of topic $i$. All non-relevant documents in $T - R$ that are ranked within the top 5,000 training documents by the dot product score, as well as all relevant training documents $R$ of a topic are selected for use in the computation of Rocchio's formula.

[1] We were not aware of the presence of the so-called "controlled language" fields in the Financial Times documents, and so the contents of these fields are used.

### 2.1.2 Rocchio's Formula

Each of the selected training documents is then represented in Ltu form. The topic statement is still represented in ltu form. The following vector denoted by the Rocchio formula is then computed:

$$\alpha \times \textit{topic statement vector} + \beta \times \textit{average relevant vector} - \gamma \times \textit{average nonrelevant vector}$$

where average relevant (non-relevant) vector is the average vector of all the selected relevant (non-relevant) training documents. In the resultant vector, we only consider the words or phrases that are present in the topic statement, or the words that occur in at least 10% of the relevant training documents, or the phrases that occur in at least 5% of the relevant training documents. We then select the top 100 words and the top 20 phrases with the highest positive weights from the set of eligible words and phrases in the resultant vector. We use the parameters $\alpha = 8$, $\beta = 64$, and $\gamma = 64$.

## 2.2 Weight Tuning Algorithm: Perceptron

Perceptron is the learning algorithm used in our past work on text categorization (Ng et al., 1997). Given a set of terms as features, and a set of training documents represented as feature vectors using the selected features, the perceptron algorithm can learn a set of weights that effectively discriminate the relevant from the non-relevant documents.

### 2.2.1 Weight Tuning

The input to the perceptron algorithm is the set of 100 words and 20 phrases selected by Rocchio's formula. However, all the weights determined by Rocchio's formula are discarded. Instead the perceptron algorithm determined from scratch the best weights of the selected terms.

All relevant training documents of a topic are used by the perceptron algorithm. In addition, 15 copies of the topic statement are added to form 15 additional relevant training documents. A subset $N_2$ comprising the top 7,000 non-relevant training documents are selected and used by the perceptron algorithm. The method of selecting these 7,000 non-relevant documents is described in the last part of this section. The set of training documents is represented in Lnu form. The maximum number of epochs that the perceptron algorithm iterates is 100.

We made two changes to the standard perceptron algorithm that resulted in significant improvement to the average uninterpolated precision (AUP):

1. In the standard perceptron algorithm, the final weights chosen for the selected features are those of the epoch at which the number of misclassified training documents (whether relevant or non-relevant) is minimized. However, since the number of relevant training documents in TREC is a lot less than the number of non-relevant training documents (7,000) we used, minimizing the total number of misclassified training documents tends to neglect the relevant documents. We have devised a metric to deal with the skewed distribution of relevant versus non-relevant training documents. Let $R$ ($N$) be the total number of relevant (non-relevant) training documents, and let $r$ ($n$) be the number of misclassified relevant (non-relevant) training documents at an epoch. We choose the feature weights of the epoch at which the metric value $r/R + n/N$ is minimized.

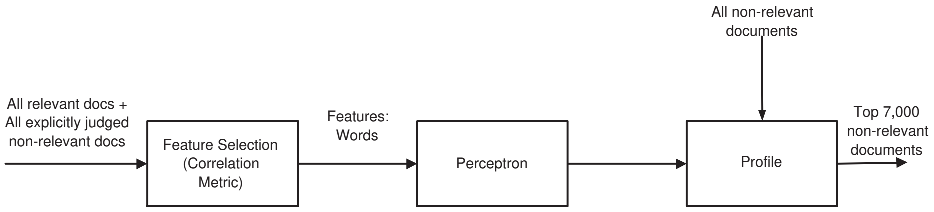


Figure 3: Selection of top 7,000 non-relevant training documents

2. Having settled on the epoch at which the weights are chosen, all terms with negative weights at this epoch are discarded before forming the final profile. This is analogous to discarding terms in the Rocchio formula with negative weights. In effect, the perceptron algorithm further prunes the set of terms chosen by the Rocchio formula, in addition to setting the weights of the remaining terms.

### 2.2.2 Non-relevant Document Selection

We are now left with describing the method of selecting the top 7,000 non-relevant documents to complete the description of the submitted run dso99rt1.

In our own work, we also found that the choice of the non-relevant training documents has a significant impact on the accuracy of the routing task, confirming the findings of (Singhal et al., 1997). However, instead of using the dot product score with the topic statement to rank the potential non-relevant training documents, we explicitly learned a classifier (using the perceptron algorithm) to rank the potential non-relevant documents. The approach is outlined in Figure 3.

We start with the set of all relevant documents $R$ of a topic $i$, as well as the set $N_1$ of all documents that have been explicitly judged as non-relevant to topic $i$ by the human assessors. Given $R$ and $N_1$, we use the correlation metric of (Ng et al., 1997) to dynamically select a set of words as features.[2] We first compute the correlation metric of any word which occurs more than 5 times in the relevant training documents. The average correlation metric value of all words with positive metric values is then computed. A word is selected as a feature if its correlation metric value is greater than the average.

This set of chosen words, as well as the training documents $R$ and $N_1$, are used by the perceptron algorithm to learn a profile. The maximum number of epochs that the perceptron algorithm iterates is 300. The learned profile is then used to rank all potential non-relevant training documents $T - R$, and the top 7,000 non-relevant training documents are selected to form the set $N_2$ of non-relevant documents mentioned earlier in this section.

[2]This metric has also been independently proposed by (Ballerini et al., 1997) for use in the routing task. The metric is termed U-measure in their work.

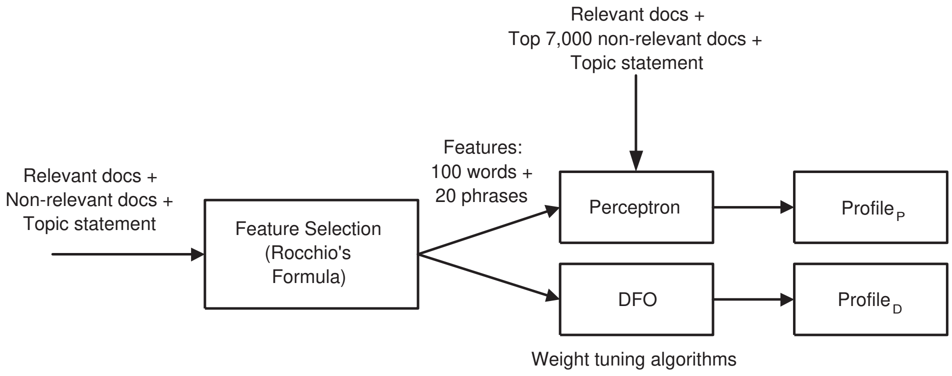


Figure 4: Training phase of our submitted run dso99rt2

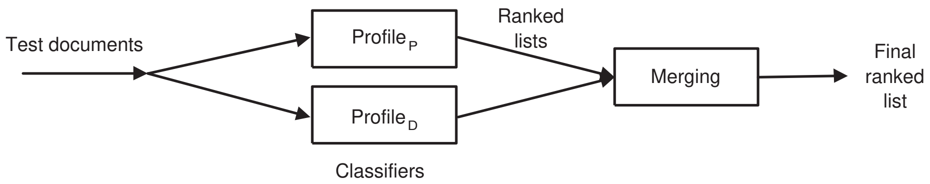


Figure 5: Test phase of our submitted run dso99rt2

## 3 The Second Submitted Run: dso99rt2

To further improve accuracy, our second submitted run dso99rt2 attempts to combine two weight tuning algorithms, namely perceptron and Dynamic Feedback Optimization (DFO) (Buckley and Salton, 1995). Both submitted runs start with an identical set of 100 words and 20 phrases selected by Rocchio's formula, but each algorithm separately tunes the weights and two profiles are generated per topic. The final ranked list of test documents is produced by merging the individual ranked lists of the two profiles. The broad outline of the training and test phase of our second submitted run dso99rt2 is shown in Figure 4 and Figure 5, respectively.

### 3.1 Weight Tuning Algorithm: Dynamic Feedback Optimization

Dynamic Feedback Optimization (DFO) has been used in conjunction with Rocchio's formula to tune the weights of selected terms in the work of (Singhal, 1998). The DFO algorithm is described in detail in (Buckley and Salton, 1995). The algorithm starts with the initial weights assigned by Rocchio's formula. It proceeds in three passes. In each pass $k$, the algorithm traverses the terms in ascending order of their weights. For each term, its weight is increased by a factor $R_k$. If the new set of weights (in which one term weight is increased by a factor of $R_k$) gives a higher AUP on the training documents, the new set of weights is kept, else the original set is retained. We use the increment factor $R_1 = 2.0$, $R_2 = 1.5$, and $R_3 = 1.25$.

### 3.2 Merging

All test documents are represented in Lnu form. Each of the two profiles generated by the two weight tuning algorithms will be used to produce a separate ranked list of the top 1,000 documents per topic. The rank of a test document is determined by the dot product score $s$ assigned by a profile to the document.

To generate a final ranked list of top 1,000 documents, all scores assigned by each profile have to be normalized to range between 0 and 1. Let $max$ ($min$) be the maximum (minimum) score assigned by a profile to the top 1,000 documents. Then the normalized score of a raw score $s$ is $(s - min)/(max - min)$. The final score of a test document is the average of the two normalized scores from the two profiles. The final ranked list is then determined by the final score of the test documents.

## 4 Results

There were six groups of participants with a total of eleven submitted runs to the routing task at TREC-8 (Each group can submit up to two runs). Our two submitted runs achieved the top two scores among the eleven runs, as measured by the official metric of average uninterpolated precision (AUP).

Our two submitted runs give very close performance. The AUP score of dso99rt1 is 45.1%, while that of dso99rt2 is 46.2%. Thus, dso99rt2 is slightly better than dso99rt1, by 1.1% in AUP. Of the 48 topics with at least one relevant test document, dso99rt2 achieves scores equal to or above the median for 46 of these 48 topics. Furthermore, the maximum scores of 15 topics were contributed by dso99rt2, and the maximum scores of 11 *additional* topics were contributed by dso99rt1.

In addition, we have tested a variant of our hybrid algorithm on 4 past year TREC data sets (TREC-3, TREC-5, TREC-6, and TREC-7), and the algorithm is able to achieve very competitive scores. Our evaluation indicates that merging multiple weight tuning algorithms is able to improve the AUP score in general.


## 5 Acknowledgements

We would like to thank Hinrich Schütze for helpful discussions.


## References


Chidanand Apte, Fred Damerau, and Sholom M. Weiss. 1994. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, July.

Jean-Paul Ballerini, Marco Buchel, Ruxandra Domenig, Daniel Knaus, Bojidar Mateev, Elke Mittendorf, Peter Schauble, Paraic Sheridan, and Martin Wechsler. 1997. SPIDER retrieval system at TREC-5. In *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*, pages 217–228.

Chris Buckley and Gerard Salton. 1995. Optimization of relevance feedback weights. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 351–357.

Chris Buckley, Mandar Mitra, Janet Walz, and Claire Cardie. 1998. Using clustering and superconcepts within SMART: TREC 6. In *Proceedings of the Sixth Text REtrieval Conference (TREC-6)*.

William W. Cohen and Yoram Singer. 1996. Context-sensitive learning methods for text categorization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Ido Dagan, Yael Karov, and Dan Roth. 1997. Mistake-driven learning in text categorization. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 55–63.

Marti Hearst, Jan Pedersen, Peter Pirolli, Hinrich Schütze, Gregory Grefenstette, and David Hull. 1996. Xerox site report: Four TREC-4 tracks. In *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 97–119.

David A. Hull, Gregory Grefenstette, B. Maximilian Schulze, Eric Gaussier, Hinrich Schütze, and Jan O. Pedersen. 1997. Xerox TREC-5 site report: Routing, filtering, nlp, and spanish tracks. In *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*, pages 167–180.

David D. Lewis, Robert E. Schapire, James P. Callan, and Ron Papka. 1996. Training algorithms for linear text classifiers. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

David Lewis. 1992. *Representation and Learning in Information Retrieval*. Ph.D. thesis, Department of Computer and Information Science, University of Massachusetts at Amherst.

Hiroyuki Nakajima, Toru Takaki, Tsutomu Hirao, and Akira Kitauchi. 1999. NTT DATA at TREC-7: system approach for ad-hoc and filtering. In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*.

Hwee Tou Ng, Wei Boon Goh, and Kok Leong Low. 1997. Feature selection, perceptron learning, and a usability case study for text categorization. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 67–73.

J. J. Rocchio. 1971. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System — Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, Inc., Englewood Cliffs, NJ.

Robert E. Schapire, Yoram Singer, and Amit Singhal. 1998. Boosting and Rocchio applied to text filtering. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 215–223.

Hinrich Schütze, David A. Hull, and Jan O. Pedersen. 1995. A comparison of classifiers and document representations for the routing problem. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Amit Singhal, Chris Buckley, and Mandar Mitra. 1996. Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29.

Amit Singhal, Mandar Mitra, and Chris Buckley. 1997. Learning routing queries in a query zone. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 25–32.

Amit Singhal. 1998. AT&T at TREC-6. In *Proceedings of the Sixth Text REtrieval Conference (TREC-6)*.

Yiming Yang. 1999. An evaluation of statistical approaches to text categorization. *Information Retrieval*, May.