Information Retrieval, 1, 193–216 (1999)

Exploiting Hierarchy in Text Categorization

ANDREAS S. WEIGEND andreas@weigend.com; www.weigend.com
Department of Information Systems, Leonard N. Stern School of Business, New York University, 44 West Fourth Street, New York, NY 10012, USA

ERIK D. WIENER

JAN O. PEDERSEN
InfoSeek Corp., 1399 Moffet Park Drive, Sunnyvale, CA 94089, USA

Received September 15, 1998; Revised April 23, 1999; Accepted May 7, 1999

Abstract. With the recent dramatic increase in electronic access to documents, text categorization—the task of assigning topics to a given document—has moved to the center of the information sciences and knowledge management. This article uses the structure that is present in the semantic space of topics in order to improve performance in text categorization: according to their meaning, topics can be grouped together into "meta-topics", e.g., gold, silver, and copper are all metals. The proposed architecture matches the hierarchical structure of the topic space, as opposed to a flat model that ignores the structure. It accommodates both single and multiple topic assignments for each document. Its probabilistic interpretation allows its predictions to be combined in a principled way with information from other sources. The first level of the architecture predicts the probabilities of the meta-topic groups. This allows the individual models for each topic on the second level to focus on finer discriminations within the group. Evaluating the performance of a two-level implementation on the Reuters-22173 testbed of newswire articles shows the most significant improvement for rare classes.

Keywords: information retrieval, text mining, topic spotting, text categorization, knowledge management, problem decomposition, machine learning, neural networks, probabilistic models, hierarchical models, performance evaluation

1. Introduction

The goal of text categorization is to assign documents to one or more predefined subject categories. This goal is central to information access tasks such as spotting topics, routing documents, assigning subject headings, and for organizing documents into hierarchical catalogs or directory-like structures.

Methods for text categorization differ in the form of the classifier, the technique for training, and the representation of the documents. However, most approaches treat the categorization problem as a set of K independent binary classification tasks, one for each category, where the information used to train each classifier consists of the set of positive and negative example documents for that class. Our baseline results and comparative results from the literature (Yang 1999) show that the best categorization methods differ only slightly in accuracy: It has become very difficult to improve performance significantly when using similar representation of the topics.

This article exploits the internal structure of the categories in order to improve text categorization performance over baseline models that ignore category structure. The categorization problem is decomposed into subtasks based on a *category hierarchy.* Problem decomposition is widely applied to reduce a larger problem into several smaller, hopefully easier, problems. The idea very naturally applies to classification with hierarchical structure in the classes, ranging from personal e-mail folders to Yahoo!-like catalogs of the Web. We implement the model as a hierarchical neural network.

First, we provide the methodological background. Then we explain the architecture and parameter estimation (or training) of the hierarchical neural network. Next we describe several approaches to represent the documents (input space) and discuss their advantages and disadvantages. After showing the basic results of the hierarchical models compared to flat baseline models, we present a few in-depth illustrations of the superior performance of the hierarchical network. The final section concludes with a summary and outlook to further research.

2. Framework

2.1. A Probabilistic Approach

In text categorization, each document d is represented through $\mathbf{x}(d)$, the input vector, and has an associated K-dimensional output vector whose dimensions correspond to a given, fixed set of K possible categories or topics. The target value for a topic is unity if the document has been assigned that topic, and zero otherwise. The goal of text categorization is to obtain a decision rule. This rule is extracted from a training set consisting of N documents with correct assignments for each category. The decision rule, when applied to a new document (presented to the classifier in the same representation as the training documents), predicts for each of the K topics its presence or absence. For example, rule learning methods, such as Swap-1 (Apte et al. 1994) and Ripper (Cohen and Singer 1996) strive to derive sets of simple rules that best separate documents based on topic assignment. Our own focus is on methods that are statistically motivated and can be interpreted in a probabilistic framework.

In particular, we assume a loss function $L_k(i, j)$. It describes the cost of assigning the value i instead of j to the kth output variable, t_k. With these definitions, the task corresponds to minimizing the total expected loss:

$$\sum_k \sum_j L_k(i, j)\ P(t_k = j \mid \mathbf{x}). \tag{1}$$

The individual losses are weighted with the probability of the output t_k taking the value j given the input $\mathbf{x}$), $P(t_k = j \mid \mathbf{x})$. The inner sum extends over the outcomes j (for the kth output), the outer sum over the outputs k. Since there are no interactions between the topics, the overall sum is minimized when each term is minimized. This yields the usual case where each output variable t_k is treated separately. For each topic, the following expression is minimized:

$$\sum_j L_k(i, j)\ P(t_k = j \mid \mathbf{x}).$$

In addition to this separation of the classes, many problems have binary decisions, $i, j \in \{0, 1\}$. Furthermore, assuming no cost for correct classification, $L(0, 0) = L(1, 1) = 0$, and unit cost for both Type I and Type II misclassification, $L(1, 0) = L(0, 1) = 1$, then the optimal decision rule assigns $t_k = 1$ if

$$P(t_k = 1 \mid \mathbf{x}) > P(t_k = 0 \mid \mathbf{x}). \tag{2}$$

Details on statistical decision theoretic can be found in Berger (1985).

Equation (2) shows that the key quantity is the probability $P(t_k \mid \mathbf{x})$. It can be estimated in several conceptually different approaches. The first approach, discussed in detail in Section 4.3.1, frames the problem as function approximation for the posterior probability P: $P(t_k \mid \mathbf{x}) = \mathbf{E}[t_k \mid \mathbf{x}]$. $\mathbf{E}[\cdot]$ denotes the expected value. It depends on the input $\mathbf{x}$ and can, for example, be expressed as a nonlinear neural network, as explained in Section 4.3.2. The second approach is indirect. It begins by estimating the probability of an input vector for each class, $P(\mathbf{x} \mid t_k = 1)$. Bayes rule is then used to flip the conditioning to obtain $P(t_k = 1 \mid \mathbf{x})$ and the decision rule, again assuming unit cost for misclassification, yields

$$P(\mathbf{x} \mid t_k = 1)P(t_k = 1) > P(\mathbf{x} \mid t_k = 0)P(t_k = 0)$$

where $P(t_k = 1)$ is the unconditional probability of class t_k. For high-dimensional input spaces, the indirect approach tends to give inferior results in comparison to the direct approach due to the problems that arise from combining often poorly approximated individual densities. A third approach does not try to give probabilistic estimates but focuses on the boundaries between the classes. For separating hyperplanes, this simplifies to the Fisher linear discriminant, as well as the iteratively estimated perceptron (Hertz et al. 1991, Haykin 1998). Recent approaches include support vector machines that are based on selecting a relevant subset of patterns in high-dimensional spaces, and large margin classifiers that find class boundaries that maximize the distance between classes (Vapnik 1998). Excellent books on classification are Cherkassky and Mulier (1998), Kennedy et al. (1998), and Duda et al. (1999).

2.2. Related Work

Some of the early work in information retrieval (Rocchio 1971), although originally proposed for relevance feedback, can be re-interpreted as a density estimation method. The Rocchio method treats the average of the feature vectors that represent the documents assigned to a specific topic as a prototype of that topic. Distances from the prototype express the likelihood that a new document belongs to the topic. If a Euclidean metric is chosen, the underlying data generating process models the noise with spherical Gaussians. This method is inappropriate when the distribution of documents for a given topic is very different from the assumed Gaussian. For example, a term with two distinct meanings is not represented well by an average of these two distinct samples.

On the Reuters-22173 corpus, Lewis and Ringuette (1994) contrasted "naive Bayes" classification with "Classification and Regression Trees" (CART) (Breiman et al. 1984). Wiener et al. (1995) compared logistic regression with nonlinear neural networks. On a

different corpus, Schuetze et al. (1995) compared several classification methods (including neural networks and logistic regression) on several reasonable document representations that included the two representations used here (described in Section 4.4). The bottom line of these experiments on document routing is: a neural network implementation of logistic regression, as used here, outperforms the Rocchio approach.

A few text categorization methods directly minimize the joint loss given by Eq. (1). For example, Yang and Chute (1992) propose a least squares approach to solve the simultaneous regression problem $E(\mathbf{t} \mid \mathbf{x})$. This idea is developed further in Yang and Chute (1994). Taking also another approach, Yang (1994) frames the problem as k nearest neighbor classification and simultaneously estimates $E(t_k \mid \mathbf{x})$ for all k. This approach does not include constraints on the output variables $\mathbf{t}$ and is primarily a computationally efficient way for solving the K independent problems.

It has been noted that category labels are typically not flat but exhibit hierarchical structure. The hierarchy used here was first proposed in Wiener et al. (1995). In other work, Dagan et al. (1996) use the latent hierarchical structure to improve cross-classification performance, and Koller and Sahami (1997) focus on word selection for classifiers based on binary inputs (presence or absence of each word).

A hierarchical classifier for documents that contain exactly one topic has been developed by D'Alessio et al. (1998). It requires hard assignments at each branch. To end up at the right leaf, every decisions in the hierarchy has to be accurate. Wiener (1995) discusses a method for improving classification by exploiting hierarchical structure. We here provide a probabilistic framework that allows several topics to be present in a given document.

3. Data Set

As a test bed for our categorization experiments we used the Reuters-22173 corpus of financial news stories from 1987, a standard evaluation data set in the text categorization literature.[1] While there are 22,173 full news stories total in the corpus, different sets of researchers have used somewhat different subsets for both training and evaluation. In particular, Lewis' original experiments with the corpus (Lewis 1992) used all of the documents, while later work by other authors used subsets consisting of a little less than half of the documents (Apte et al. 1994, Wiener et al. 1995, Cohen and Singer 1996). The subset chosen here (identical to Wiener et al. (1995)) has 9610 documents in the training set and 3662 in the test set. This is the result of selecting those topics that occur at least twice in the training set (and any number of times including zero in the test set), and all documents that have at least one of these topics assigned.

In more detail, there are 92 topics that occur at least twice in the training set. The frequency of the topics varies greatly. The most frequent topic is assigned to about a third of the documents. Documents may be assigned multiple topics. The maximum number of topics assigned to a document in the corpus was twelve. The average number of topics assigned is 1.24. This small value of this mean can be traced back to the fact that the two highest frequent topics (**earnings** and **acquisitions**) tend to occur by themselves. [To clarify whether a word denotes an input term, a topic, or a meta-topic, we use the following typographical convention: `A term or word from the document is typeset in teletype.` **A**

topic, generically called class and corresponding to the leaf of a tree is typeset in bold. A META-TOPIC, GENERICALLY CALLED GROUP AND CORRESPONDING TO A NODE IS TYPESET IN SMALL CAPS.]

While no explicit hierarchical structure was provided with the distribution of the data, an exploratory cluster analysis suggested an implicit hierarchical structure. In particular, the clustering revealed that when topics do co-occur, they tend to co-occur with other topics falling under the same meta-topic. Based on the cluster analysis, we manually grouped the topics into the following meta-topics: AGRICULTURE, ENERGY, FOREIGN EXCHANGE, METALS, and a fifth group for the remaining topics. For brevity, we call it GOVERNMENT, but do not report any results on this meta-topic it here. The Appendix lists all the topics and their assignments.[2]

4. The Model

Performance on a learning task can often be improved by decomposing the task into a set of smaller subtasks, each one easier than the whole, see, e.g., Russell and Norvig (1995), and Nilsson (1998). Our analysis of earlier categorization experiments suggested that such a decomposition could potentially be advantageous for topic spotting. This section first motivates and describes our hierarchical model, then discusses the specific implementation in terms of a neural network, and ends by describing the document representation.

4.1. Motivation

An analysis of the types of errors typically made by flat (non-hierarchical) models showed that a large portion of the high scoring *non*-relevant documents were on topics semantically related to the actually assigned topic. These false-positive stories used vocabulary related to the topic, but only incidentally. In order to predict the Reuters topics correctly, fine-grained distinctions between incidental and actual topics are essential.

For example, while the term `gold` is a good predictor for the topic **gold**, `gold` also tends to appear incidentally in documents about other precious metals, leading to false positive predictions within the group. In contrast, `gold` appears much less frequently in other groups, such as agricultural or energy stories. A hierarchical structure provides the high resolution within a group where it is needed, but drops to a more coarse resolution between groups where confusion is unlikely (and too high a resolution might lead to overfitting).

Using a hierarchical category structure allows the decomposition of the problem: first, determine the general topic group, then within that group, distinguish among topics. In the experiments with the Reuters data reported here we use a single level of hierarchy only. However, this type of decomposition can be generalized to any number of levels.

4.2. Hierarchical Model

The key quantity in Eq. (2) is $P(t_k \mid \mathbf{x})$, the probability that topic k is present given the input $\mathbf{x}$. When constructing the hierarchy, we assume that each t_k appears exactly once, i.e., each topic is assigned to one and only one meta-topic. This mutually exclusive and

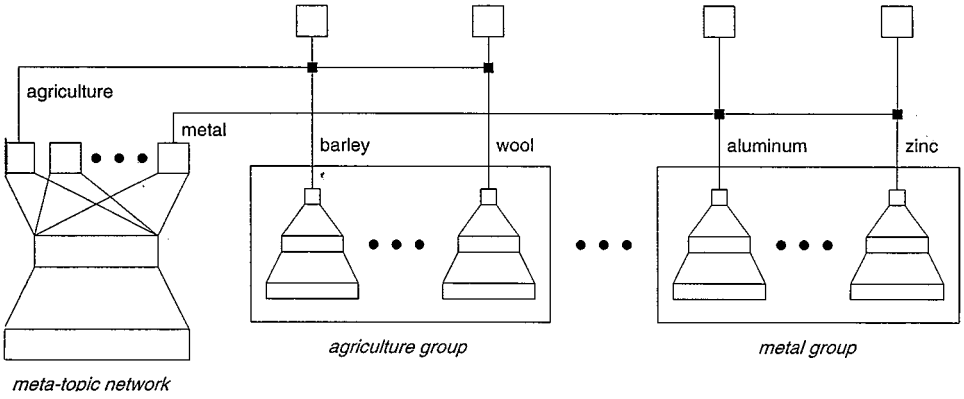

Figure 1. Architecture of the hierarchical architecture. Inputs are at the bottom, outputs on the top. Rectangular boxes with inputs and outputs indicate sets of hidden units. The meta-topic network on the left learns $P(m_i \mid \mathbf{x})$ as function of its input $\mathbf{x}$. The networks on the right specialize in predicting the topic given the meta-topic. The inputs can be different from the global inputs into the meta-topic networks, as discussion in Section 4.4.

exhaustive assumption allows us to write

$$P(t_k \mid \mathbf{x}) = P(t_k \mid m_i, \mathbf{x}) P(m_i \mid \mathbf{x})$$

where m_i denotes meta-topic i that includes topic k.

We separately estimate $P(t_k \mid m_i, \mathbf{x})$ and $P(m_i \mid \mathbf{x})$ and take the product as an estimate of the probability t_k. Note $P(m_i \mid \mathbf{x})$ acts as a gating quantity which determines the influence of the separately trained local models $P(t_k \mid m_i, \mathbf{x})$ on the overall computation of the class probability. We refer to the model for m_i as the meta-topic model, and the local models as topic models.

We implemented the decomposition outlined above using the architecture shown in figure 1. The component on the left is a classifier estimated on all the training data that learns to predict the probabilities of each of the five meta-topics for a given document that is presented at the input. A meta-topic is defined to be present if one or more of the topics it contains is present. The remainder on the right is a set of five classifier groups, one for each meta-topic. Each group consists of a separate classifier for each topic in that group, trained only on the documents of the corpus that contain the meta-topic for that group. For example, the **wheat** classifier is trained only on documents with at least one topic in the AGRICULTURE group. The wheat-classifiers can thus focus on the subtask of separating wheat documents from other agriculture documents, rather than the entire task of separating wheat documents from all other documents as a flat architecture would require.

To compute topic predictions for a given document using this hierarchical approach, we present the document to each of the topic classifiers as well as to the meta-topic classifier. Note that different representations of the document can be chosen as inputs into each classifier, see Section 4.4. The outputs of the individual topic classifiers are then multiplied by the output of the corresponding meta-topic classifier to produce final topic estimates. For example, the output of the **wheat** classifier is multiplied by the output of the AGRICULTURE

meta-topic classifier. The role of the meta-topic classifier is to "turn on" each classifier group to the degree the incoming document is judged to be relevant for the corresponding meta-topic. If, for instance, an incoming document is about precious metals but not foreign exchange, the meta-topic classifier will "shut off" the foreign exchange classifiers because their predictions are neither needed nor meaningful—the foreign exchange classifiers were not trained on metal documents.

4.3. Implementation

We write the hierarchical approach as a neural network. Section 4.3.1 briefly reviews the framework of neural networks for classification in relation to standard logistic regression. Section 4.3.2 suggests several architectures for text classification, outlines the dimensions of variation, and justifies the architecture we chose. Background references for this section include the excellent book by Bishop (1996), and, emphasizing the statistical perspective on neural networks, Rumelhart et al. (1996) and Cherkassky and Mulier (1998). The code used for the experiemnts was written in MATLAB.

4.3.1. Cost or Objective Function, and Search or Parameter Estimation. A neural network implements a functional mapping from an input or feature vector $\mathbf{x}$ to an output y, $\mathbf{x} \rightarrow y$. In the trivial case of direct connections between input and output, this architecture is identical to *linear regression* if the output is linear, and identical to *logistic regression* if the output maps the real axis onto $(0, 1)$ through

$$y = \frac{1}{2}\left(\tanh(\xi) + 1\right) \tag{3}$$

where ξ is a linear combination of the inputs.[3]

Neural networks are trained by gradient descent on a cost function. This cost function can be interpreted as the negative logarithm of the likelihood of the data given the model.

Classification tasks can be classified into 1-of-K and k-of-K tasks. 1-of-K tasks can be viewed as competition. This is not the case here, since a document can have $k = 1, 2, \ldots$ meta-topics assigned to it. Similarly, the task for each meta-topic classifier is also k-of-K learning, since given more than one topic from the subset of topics corresponding to the meta-topic may be present a document.

k-of-K classifications are equivalent to K independent 1-of-2 classifications. For each of these 1-of-2 classifications, the appropriate cost function is given by the "cross-entropy" (McCullagh and Nelder 1989, Rumelhart et al., 1996, Bishop 1996)

$$-\sum_d \left\{ t_k^d \log y_k^d + \left(1 - t_k^d\right) \log\left(1 - y_k(\mathbf{x}^d)\right) \right\}.$$

In this double sum, d extends over all documents. For each document, k denotes one specific class, y denotes the prediction, and $t \in \{0, 1\}$ is the target value that indicates membership. Interpreting the formula: when the document is assigned to the class ($t_k^d = 1$), the first term contributes $\log(1/y)$ as an error (which gets smaller the closer y is to unity), and when it is not assigned ($t_k^d = 0$); only the second term contributes $\log\left(1/(1 - y)\right)$ towards the error.

Mathematically identical, consider reversing the order of the summations. Beginning with the first topic, all documents that contain the topics have a target value of 1, and all those that do not contain the topics have a target value of 0. From this perspective it can be seen that the output will give the expected value, corresponding to the probability that the topic is present in response to a certain input. The hard constraint that the probability has to take values in (0, 1) is automatically fulfilled if a sigmoid is taken as output activation function. Note that this cost function is used for both the topics in the individual classifiers, and for the meta-topics in the meta-topic classifier.

This objective function is identical to the one used in logistic regression (apart from a sign change and sometimes a factor of two). Viewed in a maximum likelihood framework, this objective reflects an underlying binomial distribution for the errors and allows the outputs y to be interpreted as conditional probabilities.

Note that in practice, the user typically needs to go beyond a probabilistic assignment and needs to make a decision, often with a very asymmetric loss function (see Section 2). For example, missing a document in a search for prior work for a patent application is much worse than bringing up an irrelevant document. Luckily, this does not change the problem fundamentally—a different decision threshold is taken into account through a linear transformation of the cost function. In the evaluation part of this paper, Section 6, we consider several decision thresholds and average over them in order to obtain single performance measures.

4.3.2. Architecture. We now turn to the "architecture," i.e., the functional form of the classifier, and show how it incorporates assumptions about the problem domain.

Figure 2 shows three examples of non-hierarchical solutions. In all cases, the information flows from the bottom to the top, i.e., the bottom lines represent the inputs, the horizontal

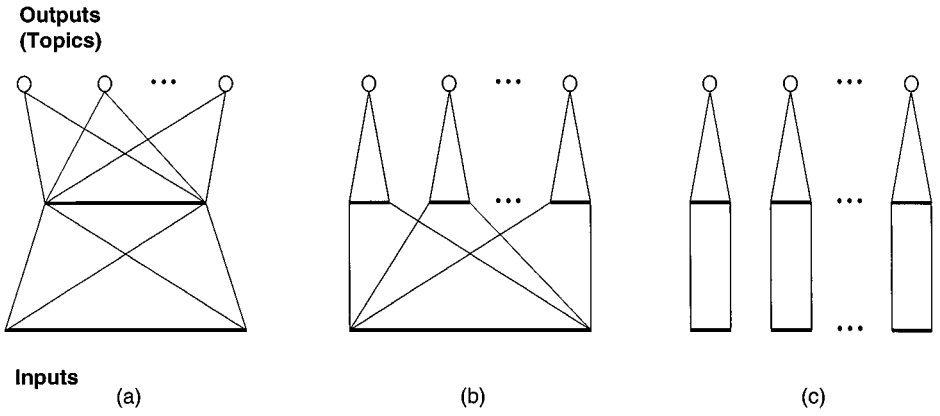

Figure 2. Three flat (non-hierarchical) architectures. Architecture (a) describes the case of shared inputs and shared hidden units, (b) of shared inputs but separate groups of hidden units for each topic, and (c) of individual sets of inputs and hidden units for each topic. This paper uses architecture (a) for the network that predicts the meta-topics (left part of figure 1), and architecture (c) for the networks that predict the individual topics, given the meta-topic (right part of figure 1).

lines in the middle represent the hidden units, and the top circles represent the outputs (the predictions for the topics). Connections between these lines indicate the parameters or weights that are estimated from the training set by minimizing the cost function discussed in Section 4.3.1 above. The hidden units are given tanh activation functions (centered around zero), and the output units are given sigmoid activation functions (Eq. (3)) that incorporate the hard constraint for the range of a probability.

Figure 2(a) uses a common set of inputs that feed into one single set of shared hidden units. This global re-representation of the inputs in the hidden units space is followed by individual logit models on the hidden units.

Figure 2(b) still uses one common set of inputs. The hidden units, however, are broken into non-interacting subsets, each corresponding to one of the output classes. Both these architectures have the same inputs. This allows them to operate on a single set of selected terms, or a representation such as LSI (Latent Semantic Indexing), discussed in Section 4.4.

In contrast, figure 2(c) gives each classifier a different set of inputs, chosen to be particularly useful for that specific class. The overall architecture in (c) is a parallel estimation of all the probabilities—there is no (positive or negative) interaction between the classes. Note that this complete independence allows the easy addition of new classes, since each network is trained separately. Furthermore, the size of the weights between inputs and hidden units in each network indicates the importance of individual terms for the individual tasks. Since sharing hidden units does not make sense in this case, there are no further possibilities for nonhierarchical architectures.

We use both architectures (a) and (c) as "flat" baseline models in this article, enabling us to gauge the relative performance of our hierarchical models for both shared and class-specific representations. We do not use architecture (b) since, as noted above, nonlinearity does not appear to play a major role in this problem. Previous work (Wiener 1995) has shown that architecture (c) performs best on the Reuters data but that architecture (a) is adequate for the coarse-grained distinctions exemplified by the high-frequency topic labels.

The hierarchical architecture we propose in this article is shown in figure 1. Within each group, the sub-models correspond to figure 2(c), i.e., individual networks with relevant terms selected for each individual prediction task. The important difference is the "group network" on the left. It has a single shared set of inputs, a set of shared hidden units, and one output for each *meta-topic*, such as AGRICULTURE and ENERGY. In summary, we use architecture (a) for the meta-topic, and architecture (c) for the individual topics.

We train the networks using a fully supervised approach: Every document has explicit targets for the meta-topics and for the individual topics. As discussed above, each topic had been manually assigned to exactly one meta-topic. This allows for completely separate training of all networks, using cross-entropy errors on all levels.

4.4. Document Representation

To build the classifier, we still need to address the question of how to represent the text, i.e., how do we show each document to the classifier? The total number of terms (order of 50,000, the exact value depending on preprocessing such as stemming and truncation) is too large as input space: in such high-dimensional spaces almost all points (i.e., documents)

are very far apart, making generalization from the training set to the test set very difficult ("curse of dimensionality"). The dimensionality of the large space needs to be reduced for the input into the classifier.

We use two statistical techniques to reduce the number of inputs. The first projects the full term vector linearly onto a hyperplane and only retains the coordinates along that hyperlane. The hyperplane is chosen such that the sum of the squared distances between the original points and their projections is as small as possible. This standard technique, known in statistics and engineering as principal component analysis, as singular value decomposition, and as Karhunen-Loéve transformation, is called in the information retrieval field *Latent Semantic Indexing* (LSI) (Deerwester et al. 1990). It works well to the degree that most of the "signal" can indeed be captured by a hyperplane of about 1 percent of the total number of dimensions, and that the remaining 99 percent of the dimensions mainly capture the "noise."

Within this first technique, the remaining decision is whether to use all of the documents of the corpus ("global"), or only a specific subset ("local"). We use LSI for both. In the global case, the input into the first classifier (which predicts meta-topics) is determined by applying LSI on the document-term matrix of the entire corpus. In the local case, the inputs into the other classifiers (which predict the individual topics given the meta-topic) are obtained by applying LSI on the document-term matrix that contains only the documents in a topic group. This local representation for each topic group (meta-topic) varies across groups.

While LSI captures as much variance as possible for any given number of retained variables, its dimensions are typically linear superpositions of the original terms. This problem is addressed in the second technique we use. The goal of term selection is to choose a small subset of highly discriminating terms from the full set of terms and use these terms as input dimensions. More specifically, we use the chi-squared statistic to estimate the predictive power of the terms with respect to topics, and then select the highest ranked terms. The chi-squared statistic measures the discrepancy between the observed counts in a contingency table of topic-term co-occurrence, and the expected counts under the assumption that the terms are distributed uniformly in all of the documents:

$$\chi^2 = \sum_{\text{all cells}} \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$$

Details and interpretation of the chi-squared measure are given, for example, in Collett (1991).

We use the chi-squared term selection approach for the networks that predict the individual topics. We compute a separate input representation for each topic based upon analyzing the predictive power of each term with respect to the particular topic. These separate representations allow us to keep the number of inputs for each of these networks small.

When determining the chi-squared values for a given topic, the remaining choice is whether to compute the expected values from the entire corpus, or from the meta-topic under which the topic under consideration falls. For example, to select a local set of terms for predicting the topic **crude**, the selection technique can either analyze topic-term co-occurrence over the entire training set or over just the documents in the ENERGY meta-topic. The difference is best illustrated by comparing the top fifty terms selected for **crude** using the chi-squared measure on the full training set versus just on the ENERGY set. Using the

entire corpus, we obtain the following predictors:

```
barrel oil crude bpd opec petroleum
energy exploration refinery bbl gas
distillate gasoline sea drill production
saudi offshore day iraq api kuwait arabia
ecuador tanker petroleos wti exxon cubic
iranian earthquake gulf iran natural
pipeline texaco grisanti hernandez xon
shell crudes field venezuela petrobras
al escort herrington output sour iraqi
```

If we do not use the entire corpus, but only on the documents in the ENERGY set, the best predictors are:

```
oil crude harbor file gallon octane
pound barrel resin unlead bpd
petrochemical last thermoplastic butane
midland barge fall customer effective
opec houston super pel beaumont
ethylene polypropylene widely propane
dow modernization component adhesive
jet cgp pittsburgh convert heat diesel
petrol regular tonne day co sulphur
cts he subsidiary corp cent
```

Note, for example, that the terms `petroleum` and `energy` are considered good predictors for **crude** in the full training set, but are not considered good predictors within the context of ENERGY documents. Overall, local representations tend to give better performance in the hierarchical network because they pick out the relevant discriminating features.[4] This is why we use the highest ranked terms for a given topic from the given local comparison with the meta-topic only, and not with the entire corpus.

5. Results

5.1. Evaluation Measures

To present a single number evaluation of a classifier for a particular topic, one summarizes effectiveness over a range of potential decision thresholds. We do this by computing P_{avg}, the average precision over a fixed set of evenly spaced recall levels. For a given topic, one can rank test documents by the predicted probability that they belong to that topic. The true topic assignments for test documents are known. Two numbers can be computed for any cut point. The first number, *precision*, is given by the following ratio: the number of documents that are correctly assigned to the topic divided by the number of documents retrieved. The second number, *recall*, is given by the following ratio: the number of documents that are correctly assigned to the topic divided by the total number of documents that have that topic.

To achieve one single number for one topic from the entire precision recall curve, it is customary to average precision over a number of recall points. All the numbers reported here are the arithmetic mean of the precision values at eleven evenly spaced recall points of 0%, 10%, 20%, . . . ,90%, 100%. If no exact cutoffs were available, we used linear interpolation to obtain an approximate value. In this computation, the specific decision threshold to achieve a particular recall point can vary by topic. We only report this measure here, it is highly correlated (order of 0.95) with other evaluation measures, such as the so-called F-measure (van Rijsbergen 1979). A detailed discussion of measures for evaluation is given in Yang (1999), and the exhaustive compilation of our experimental results using a variety of evaluation measures is given in Wiener (1995).

The P_{avg} measure characterizes the average performance for each topic. If we want to summarize the performance over a whole set of topics, two ways of averaging are used in the literature: The effectiveness can be computed for each topic separately and then averaged over the topics, or the topic decisions can be computed for each document and then averaged across documents. The following two terms are used to describe these situations:

- *Macroaveraging* takes the expectation where the topics are given even weight. A topic that only appears a couple of times in the test set influences the results as much as a topic that appear thousands of times as often.
- *Microaveraging* takes the expectation where the documents are given even weight. The resulting value is dominated by the frequent topics, since they appear much more often than the rare ones.

Microaveraging essentially measures performance for the easy-to-predict high-frequency topics. This is also a main reason of the similarity between the microaveraged results of most text categorizations methods on the Reuters data. In comparison, macroaveraging emphasizes the medium and low frequency topics. This is harder but bears more information. This paper thus focuses on macroaveraged results.

Some systems report performance only using microaveraging. Lewis (1992) uses proportional assignment rather than fixed recall points for finding decision thresholds. When necessary for a clear comparison, we also microaverage.

5.2. Baseline Performance

The first step in our empirical evaluation establishes baseline figures for non-hierarchical, flat models. We show that the results of our baseline model compare favorably with those of previously published results for the same corpus, as well as with standard alternative text categorization models.

5.2.1. Baseline Model. For the flat network, we used architecture (c) from figure 2, where a separate classifier was trained for each topic. We tried both a global LSI representation (retaining 200 dimensions) and global selected term representations (using the top 20 terms for each topic). Each of the networks had six hidden units.

5.2.2. Comparison to Published Results. Our experimental setting is most similar to the one used by Apte et al. (1994), hence we can directly compare results. While there is a difference in the training/test split of the Reuters data, Yang (1999) provides evidence that this difference in splitting the data has no major effect on the performance. The best experiment reported by Apte et al. (1994) (excluding experiments with extra weight given to headline terms) microaveraged over all 92 topics and resulted in a breakeven point of .789.[5] The breakeven points for our flat models, microaveraged over the same 92 topics, are .801 for the LSI network and .775 for the selected term network. This, combined with the careful cross-system comparison given in Yang (1999), suggests that our flat models are competitive with the best results reported for text categorization over the Reuters data.

5.2.3. Comparison to Other Methods. To establish the baseline performance further, we also compared it to two other standard categorization methods on the same data: k-nearest neighbors (Masand et al. 1992) and prototype-based classification (Hull 1994, Ittner et al. 1995). For these experiments, we macroaveraged over the top 58 most frequent topics. This attempts a compromise between too much emphasis on the high-frequency topics (as micro-averaging would have done) and too much variance from the low-frequency topics (below 16 occurrences in the training set).

The best results we achieved for k-nearest neighbors were 0.644 average precision using LSI and 0.756 average precision using selected terms, compared with 0.765 for the network using LSI and 0.771 using selected terms.

We also established a benchmark performance for a prototype-based approach. Using the Buckley et al. (1994) variant of the Rocchio (1971) algorithm, we first built a prototype for each topic, and subsequently assigned topic scores based on the cosines between documents and prototypes. The best results we achieved were 0.637 average precision for an LSI representation and 0.678 average precision for a term representation, which is below the 0.765 for the LSI and the and 0.771 for the selected term networks, reported above.

These results show that our baseline models are at least competitive with other standard text categorization techniques.

5.3. Hierarchical Performance

For our experiments with the hierarchical model, we decided to exclude the umbrella meta-topic that we had called GOVERNMENT. As can be seen from the detailed list in the Appendix, this meta-topic had little semantic coherence (ranging from t-bond and housing to ship) and would have watered down the results since its breadth is similar to the non-hierarchical case. Applying this cut along with the requirement to have at least 16 positive examples in the training set leaves the final set of 37 topics for the experiments. This set is used throughout the experiments reported here. On this set, we used topic-specific term selection for the topic networks, trying both local and global chi-squared selection (retaining the top 20 terms as inputs). We also tried both local and global LSI representations for the topic networks, in each case using the first 200 dimensions as inputs. For the meta-topic network, we used a global LSI representation, retaining the first 200 dimensions as inputs into the network. In all networks, we used six hidden units, as in the baseline case.

Table 1. Macroaveraged P_{avg} for hierarchical and flat comparison models over 37 topics. The column labeled 'All' gives the performance over all 37 topics. The remaining three columns indicated the performances for top 1/3 by frequency $\geq$76 examples in the training set, the middle third (between 36 and 75 examples in the training set), and the bottom 1/3 (between 16 and 35 examples in the training set).

Model	All	High	Medium	Low
H-GlobalTerm	0.852	0.893	0.833	0.845
H-LocalTerm	0.849	0.892	0.815	0.852
H-GlobalLSI	0.742	0.847	0.782	0.659
H-LocalLSI	0.831	0.875	0.796	0.835
F-GlobalTerm	0.811	0.875	0.787	0.798
F-GlobalLSI	0.764	0.864	0.811	0.678

Table 1 shows precision macroaveraged over four topic frequency ranges: all topics, the most frequent third, the middle, and the lowest frequent third of the 37 topics. (The exact cut-offs are described in the caption.) The following six model classes are compared:

- H-GlobalTerm: a hierarchical model using global topic-specific term selection for the topic models;
- H-LocalTerm: a hierarchical model using local topic-specific term selection for the topic models;
- H-GlobalLSI: a hierarchical model using global LSI representation for the topic models;
- H-LocalLSI: hierarchical model using a local LSI representation for the topic models;
- F-GlobalTerm: a flat model using global topic-specific term selection;
- F-GlobalLSI: a flat model using global LSI representation.

The performance of the last two entries, the flat baseline networks, is computed on the same 37 topics and subgroups as used in the hierarchical networks. For those two entries we only use global methods (both term selection and LSI) since the networks are trained on the full corpus. Note the improvement in performance of the hierarchical models compared to the flat models, and of the local representations compared to the global representations.

Table 2 tests for the significance of the differences between models by using a paired t-test on the average precision values across the individual topics (Hull 1993).

Table 2. Percentage difference in P_{avg} for hierarchical models compared to relevant baselines. Figures in bold were found to be statistically significant using a paired t-test at level .05.

Comparison	All	High	Medium	Low
Hierarchical vs. hierarchical				
H-LocalLSI vs. H-GlobalLSI	**9.4**	1.7	1.0	**21.8**
H-LocalTerm vs. H-GlobalTerm	−0.4	−0.1	−2.2	0.8
Hierarchical vs. flat				
H-GlobalTerm vs. F-GlobalTerm	**5.0**	2.1	5.9	**6.0**
H-GlobalLSI vs. F-GlobalLSI	**−2.9**	−2.0	**−4.5**	−2.1

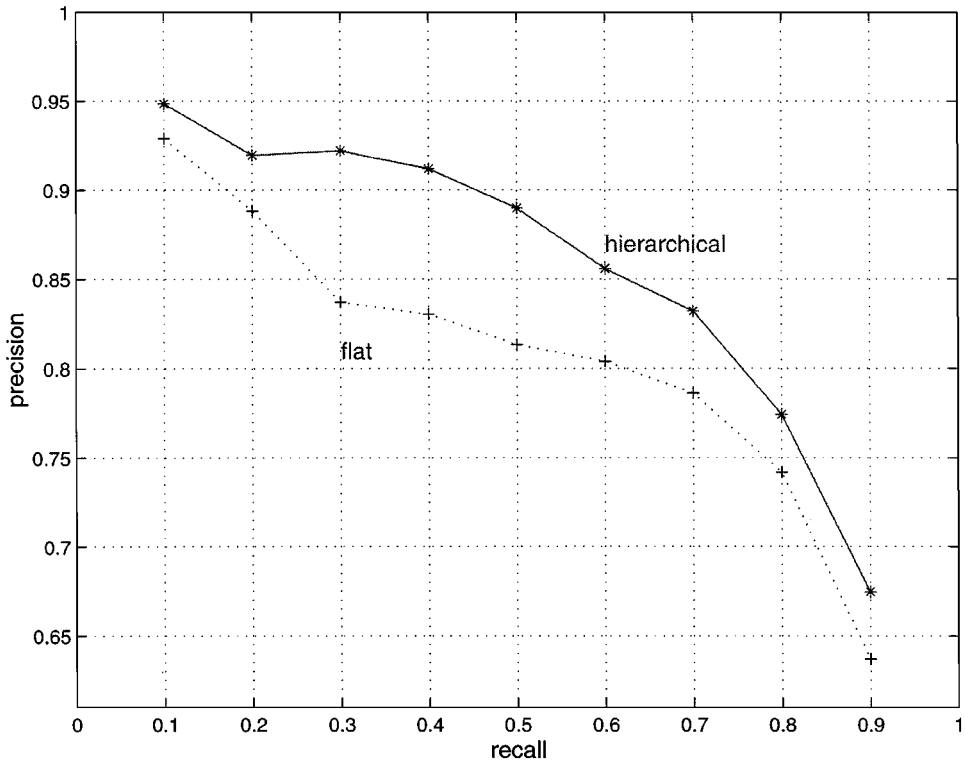

Figure 3. Macroaveraged precision of the hierarchical network compared to the flat network, both using global term representations.

The full precision-recall curves for the hierarchical network in comparison to the flat networks using global term selection are shown in figure 3.

6. Analysis

The main result of all our experiments is that the best performance is obtained by the hierarchical network that uses global *term* selection. The improvement in performance over a flat network is especially noticeable for lower-frequency topics. Furthermore, the hierarchical network with a global *LSI* representation for the topic classifiers performs relatively poorly—significantly worse than a flat network with a global LSI representation. This suggests that global LSI is not a good representation for the topic classifiers of a hierarchical model. However, the local LSI representation performs almost as well as the selected term representation for the hierarchical model.

The performances of the hierarchical model with local and with global selected term representations are not significantly different. Thus, we can attribute all of the improvement over the flat model to the hierarchical model itself rather than to the representation.

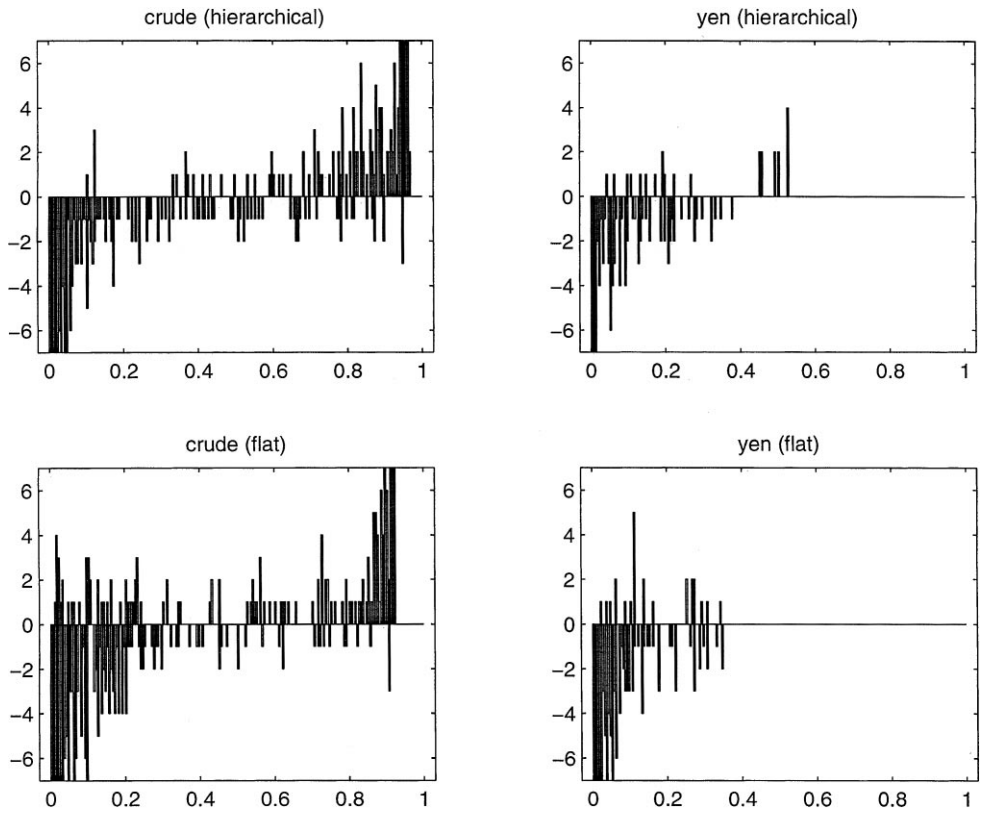

Figure 4. Comparison of output distributions for hierarchical and flat networks.

To better understand the performance difference between the hierarchical and flat models, figure 4 compares the output distributions for two networks predicting the topics **crude** and **yen** using global term selection. The x-axis labels output probability, and the y-axis labels frequency of occurrence of that probability for test documents which both should correctly be assigned the topic (positive y axis) and should not be assigned the topic (negative y axis). Note that the spread between true positive and true negatives is greater for the hierarchical model, and hence the likelihood of misclassification is smaller than in the non-hierarchical case.

There appear to be two reasons for the hierarchical network's improvement over the flat network assuming a selected term representation. First, the meta-topic network, with its shared representation, does a good job shutting off meta-topics that have no bearing on the document. Thus, many of the false positives generated by the flat model are not generated by the hierarchical model. For example, we observed that many of the **earnings** and **acquisitions** documents that were erroneously given high scores by the flat models, were in contrast shut off by the meta-topic component of the hierarchical model.

Table 3. Comparison of weights in a linear flat network and linear hierarchical network predicting the topic **natural gas**. The first 16 terms are listed in the order selected by the chi-squared measure.

Term	Hierarchical	Flat
gas	.829	.491
oil	−.707	−.071
natural	.169	.133
cubic	.223	.237
barrel	−.183	−.022
foot	.143	.054
reserve	.244	.120
production	.227	.214
energy	.055	.186
price	.441	.525
exploration	.184	.254
well	−.157	−.026
petroleum	.049	.156
drill	.087	.112
say	.043	.109
pipeline	.094	.132

Second, models trained locally can be more sensitive to the subtle distinctions between similar topics than globally trained models. As an example, consider the topic **natural gas**, for which the hierarchical model does substantially better than the flat model. Examining the weights indicates the terms that are important for the two models. Table 3 displays the weights for both a linear flat network and the topic component in a linear hierarchical network, both using the same 16 selected terms.[6] The most striking difference is in the weights for the term `oil`. In the flat network, `oil` appears to be essentially ignored, while in the hierarchical network it is a strong negative predictor. Apparently, `oil` is important for discriminating between natural gas and non-natural gas documents within the region of most difficulty, the energy documents. However, this is not picked up on by the flat network. Since `oil` has two very different senses (petroleum vs. vegetable oil), it may be difficult for the flat network, trained on the entire corpus, to make use of `oil` as a predictor.

We found the same phenomenon in examining the **crude** and **yen** topic networks. For example, the term `gas` is picked as a good positive predictor for the topic **crude** by the global term-selection measure, but ends up with a strong negative weight in the hierarchical network. In the flat network, on the other hand, the weight for `gas` is close to zero. For the topic **yen**, `dollar` is selected as the second highest positive predictor by the chi-squared measure, but it is given a negative weight in the hierarchical network and no weight in the flat network. Thus, while `dollar` appears to predict the topic **yen**, it is actually more useful in helping to discriminate **yen** documents from other foreign-exchange documents. This analysis confirms the importance of doing local modeling within the context of topic groups.

7. Conclusion and Outlook

Many problems in information systems and knowledge engineering have some domain structure of hierarchical nature. Examples range from cataloging systems in traditional libraries to hierarchical directories on the Web. Most published algorithms for text categorization do not take any advantage of the inherent hierarchical structure. The task, to obtain estimates of the probabilities of the individual categories, is very difficult to achieve for flat, non-hierarchical architectures. On the input side, a flat architecture requires a large number of predictors to contain sufficient information for the potentially very subtle distinctions in between very close categories. The finer the required distinctions are, the worse the curse of dimensionality becomes. On the output side, many categories have a very low ratio of positive to negative examples, often only one positive for tens of thousands of negative examples.

This paper addresses these problems by using a divide-and-conquer strategy that mimics the hierarchical structure of knowledge that is ignored in flat inference models. On the first (coarse) level of the hierarchy, the task is to assign the probabilities of the meta-topics to each document. We have manually divided all topics into five meta-topics. Each meta-topic contains a reasonable number of positive examples in comparison to the negative examples. Since this level does not require particularly fine distinctions between individual topics, Latent Semantic Indexing (LSI) turns out to be an appropriate input representation. We use a 200-dimensional linear subspace of the original term space.

On the next level of the hierarchy, each model has to learn to differentiate only within the meta-topic it belongs to, and no longer against the other meta-topics. Note that this again avoids too small ratios of positive to negative examples in training. For the inputs of these sub-models, we compared several choices for the inputs, consisting of individual terms as well as of LSI. In both cases, we computed the optimal sets both on a group basis (where every sub-model had the same set of inputs), as well as individual sets for each subset or category.

The improvement is robust: Differences between these 2×2 choices were small compared to the gain obtained through the hierarchical approach in comparison to a non-hierarchical approach. We presented a statistically significant overall improvement of five percent for averaged precision. The strongest gains are on rare categories that otherwise suffer from the lack of relevant inputs and a too small ratio of positive to negative examples. Beside the performance improvements, we gave several insights into the specific solutions the network has found.

This paper used a hierarchy of two levels. The extension to more than two levels is straightforward and can be carried out recursively. From a practical perspective, this makes sense as long as a sufficient number of positive training example per category is available. For the Reuters corpus, two levels were appropriate. For most Web search catalogues or directories, deeper hierarchies can (and should) be used. The main result of this paper is that a hierarchical structure helps to improve predictions for rare classes. Furthermore, the different parts of the hierarchical architecture can be trained independently. This solves the otherwise very serious problem of scaling up to very large and heterogeneous text collections such as the Web.

The approach presented here is based on a solid statistical framework that allows the interpretation of the results as probabilities. This is important since it allows the combination of topic predictions from our model in a principled way with information from other models.

Appendix: The Topics and Their Hierarchy

The Appendix presents all of the topics with additional information in form of a table.

Table 4. The topics are sorted in decreasing frequency on the training set. The columns are: topic rank, topic name, number of documents in the training set, number of documents in the test set, percentage of documents that have at least one additional topic assigned ("%>1"), first letter of topic group. The last five columns give the corresponding meta-topics of additionally assigned topics, when present.

Rank	Topic	Train	Test	%>1	Group	%A	%E	%F	%G	%M
1	earn	2878	1170	2	G	0	27	0	69	11
2	acq	1650	774	4	G	10	26	1	53	16
3	cbond	954	169	24	G	0	0	1	98	1
4	money-fx	541	206	59	F	0	0	50	56	0
5	grain	434	169	91	A	95	1	1	11	1
6	corp-news	420	95	36	G	2	6	0	83	11
7	loan	392	117	18	G	14	12	17	69	3
8	crude	388	196	38	E	2	47	1	63	2
9	trade	369	135	35	G	23	5	24	58	6
10	interest	346	148	56	G	0	0	71	36	0
11	wheat	212	82	100	A	100	1	0	5	1
12	ship	198	92	50	G	34	56	0	8	3
13	corn	177	64	100	A	100	0	0	5	1
14	ebond	173	55	71	G	0	0	1	100	0
15	money-supply	140	37	15	G	0	0	37	81	0
16	dlr	131	54	96	F	1	1	98	14	0
17	sugar	126	43	25	A	69	7	0	38	7
18	oilseed	123	54	94	A	99	1	0	10	0
19	gbond	115	23	20	G	4	0	11	75	18
20	coffee	113	32	21	A	60	13	0	53	10
21	tbill	109	17	24	G	0	0	23	83	0
22	gnp	103	37	46	G	6	2	20	91	0
23	gold	94	33	35	M	2	0	9	44	60
24	veg-oil	87	40	76	A	97	2	0	11	2
25	soybean	79	38	100	A	100	0	0	7	0
26	nat-gas	75	33	70	E	1	91	0	33	1

(*Continued on next page.*)

Table 4. (*Continued*).

Rank	Topic	Train	Test	%>1	Group	%A	%E	%F	%G	%M
27	livestock	75	30	77	A	98	0	1	12	1
28	bop	75	33	71	G	3	1	9	100	0
29	cpi	69	30	27	G	4	7	19	93	0
30	cocoa	56	19	16	A	83	8	0	25	8
31	reserves	55	19	32	F	4	0	42	58	8
32	carcass	50	20	84	A	98	0	0	15	0
33	copper	47	20	40	M	7	0	0	41	52
34	jobs	46	23	26	G	0	0	6	100	0
35	yen	45	24	84	F	0	0	100	16	0
36	tbond	42	7	35	G	0	6	29	100	0
37	ipi	42	12	17	G	11	0	0	100	0
38	iron-steel	40	17	35	M	20	25	0	90	10
39	cotton	39	24	59	A	97	3	3	16	8
40	rubber	37	15	21	A	82	9	0	36	18
41	gas	37	19	70	E	3	85	0	15	5
42	barley	36	16	100	A	100	0	0	4	0
43	rice	35	28	98	A	100	0	0	10	2
44	alum	35	24	19	M	9	0	0	73	36
45	palm-oil	30	12	100	A	100	5	0	14	5
46	meal-feed	30	20	74	A	100	0	0	3	0
47	fbond	27	3	13	G	0	0	25	100	0
48	sorghum	23	13	100	A	100	0	0	0	0
49	retail	23	2	20	G	0	0	0	100	0
50	zinc	21	15	56	M	0	0	0	10	100
51	silver	21	8	86	M	4	0	4	20	96
52	pet-chem	20	13	42	E	14	50	0	57	0
53	wpi	19	10	21	G	0	33	0	67	0
54	tin	18	16	12	M	75	0	0	50	25
55	stg	18	0	94	F	0	0	94	35	0
56	strategic-metal	17	11	46	M	0	0	0	38	69
57	rapeseed	17	8	100	A	100	0	0	8	0
58	orange	16	11	19	A	100	0	0	20	0
59	housing	16	4	15	G	0	0	0	100	0
60	hog	16	7	96	A	95	0	0	14	0
61	lead	15	15	77	M	0	9	0	26	74
62	heat	14	7	52	E	9	82	0	27	0
63	soy-oil	13	13	100	A	100	0	0	4	0

(*Continued on next page.*)

Table 4. (*Continued*).

Rank	Topic	Train	Test	%>1	Group	%A	%E	%F	%G	%M
64	fuel	13	11	54	E	0	92	0	15	0
65	lei	12	3	7	G	0	0	0	100	0
66	sunseed	11	6	100	A	100	0	0	6	0
67	soy-meal	11	11	100	A	100	0	0	0	0
68	dmk	10	4	100	F	0	0	100	14	0
69	tea	9	4	62	A	100	25	0	38	25
70	income	9	8	35	G	0	0	33	100	0
71	nickel	8	2	50	M	0	0	0	20	100
72	lumber	8	7	27	A	50	0	0	50	0
73	oat	7	6	100	A	100	0	0	0	0
74	l-cattle	6	3	89	A	100	0	0	0	0
75	sun-oil	5	2	100	A	100	0	0	0	0
76	rape-oil	5	4	100	A	100	0	0	11	0
77	platinum	5	8	69	M	0	0	0	11	100
78	inventories	5	0	40	G	0	0	0	100	0
79	instal-debt	5	1	0	G	0	0	0	0	0
80	groundnut	5	6	82	A	100	0	0	11	0
81	oil	4	1	100	E	40	40	0	20	0
82	jet	4	1	40	E	0	100	0	0	0
83	coconut-oil	4	3	100	A	100	0	0	14	0
84	coconut	4	2	83	A	100	0	0	0	0
85	austdlr	4	0	100	F	0	0	75	75	0
86	propane	3	3	83	E	0	80	0	40	0
87	potato	3	4	14	A	100	0	0	0	0
88	can	3	1	75	F	0	0	100	0	0
89	wool	2	0	50	A	100	0	100	100	0
90	saudiriyal	2	1	100	F	0	0	67	33	0
91	palmkernel	2	1	100	A	100	0	0	0	0
92	naphtha	2	4	83	E	20	80	0	20	20

Acknowledgments

We would like to thank the two anonymous referees for their valuable comments, as well as the following individuals: Marc Ringuette made the Reuters data available to Andreas Weigend. The preprocessing of the corpus was done by Jan Pedersen at Xerox PARC in 1992. Hinrich Schütze generated the LSI representations. Kitinon Wangpattanamongkol helped explore the plethora of different term selection methods while an undergraduate

at Chulalongkorn and a summer student at Xerox PARC in 1992. Andreas Weigend thanks David E. Rumelhart for sharing his intuitions about neural networks, and John W. Tukey for all the discussions about input features and his guidance in the interpretation of the clustering results that lead to the final hierarchical structure we used here. The results presented here were obtained by Erik Wiener during summer 1994 at Xerox PARC and are presented in his Masters thesis at the University of Colorado (Wiener 1995). A different set of questions on the same corpus was answered at SDAIR in 1995 (Wiener et al. 1995).

Notes

1. The Reuters-22173 collection has been replaced since our experiments by the Reuters-21578 collection, which represents a cleaner, better documented subset of the same underlying data. The Reuters-21578 collection is maintained by David D. Lewis at www.research.att.com/∼lewis/reuters21578.html
2. While we manually picked these groups in order to focus on the effect of a grouping onto the task of text categorization, is also possible to *learn* a hierarchical structure. Pereira et al. (1993) use an annealing approach to clustering proposed by Rose et al. (1990) for this task. Hofmann (1998) shows that a hierarchical mixture model can be used for clustering of documents. Note, however, that we are interested in supervised learning, since we start with a collection of documents where topic classes are provided and already assigned. This allows for rigid evaluation of the performance which is by nature not possible in unsupervised learning (here in the form of clustering and of competitive learning). Furthermore, many documents are not solely about a single topic but tend to be assigned more than one topic. This is in contrast to the assumption made in the data generating process for clustering where each observed vector is assumed to have been generated by exactly one hidden group. An interesting area of research is the development of statistically principled methods that allow for overlapping groups and combine the supervised part (topics given) with an unsupervised part (groupings to be learned).
3. The comparison to logistic regression (i.e., no hidden units) is a healthy check: many standard statistical packages have good routines for logistic regression. The neural network without hidden units should lead that same results as logistic regression. When hidden units are subsequently introduced, the amount of improvement over simple logistic regression indicates the importance of the potential nonlinear structure in the data. Our experiments on the Reuters topic spotting data do not show significant improvement with hidden units over direct connections. We suspect that two factors make it difficult to find nonlinearities: (1) the simple heuristic of "early stopping" has a bias towards linear models (LeBaron and Weigend 1998), (2) the high noise in the data masks potentially more subtle nonlinearities. To focus this article on the issue of hierarchy, we use hidden units in the neural networks throughout this paper and eliminate the extra dimension of nonlinear vs. linear logistic regression. The detailed comparison between linear and nonlinear performances is given in (Wiener 1995).
4. It might be interesting to compare our two sets of terms for the topic **crude** with the top ten terms given by Koller and Sahami (1997) ("KS"). Using the chi-squared measure based on the entire corpus, the top 50 terms include eight of the ten KS terms (missing `ship` and `attack`). In contrast, when computing the chi-squared measure based on the ENERGY set, the top 50 terms include only a single KS term (`barrel`). This indicates that KS's term selection method does not exploit the local structure to the same degree as the chi-squared approach on the ENERGY set does.
5. We thank one of the referees for pointing out that these results have recently been improved. The best published results on the Apte et al. (1994) are now 0.85 using k-nearest neighbors (Yang 1999). This can be interpreted as a further validation of the range of the baseline results.
6. Since we found little difference in the performance of linear and nonlinear networks, we use a linear network for analysis here in order to be able to more easily analyze the network weights. We could have also used sensitivity analysis to discover the important inputs in the nonlinear networks.

References

Apte C, Damerau F and Weiss S (1994) Towards language independent automated learning of text categorization models. In: Proceedings of the 17th Annual ACM/SIGIR Conference, pp. 23–30.

Berger JO (1985) Statistical Decision Theory and Bayesian Analysis. Springer Verlag.

Bishop CM (1996) Neural Networks for Pattern Recognition. Oxford University Press.

Breiman L, Friedman JH, Olshen RA and Stone CJ (1984) Classification and Regression Trees (CART). Wadsworth, Pacific Grove, CA.

Buckley C, Salton G and Allen J (1994) The effect of adding relevance information in a relevance feedback environment. In: Proceedings of the 17th Annual ACM/SIGIR Conference, pp. 292–300.

Cherkassky VS and Mulier FM (1998) Learning from Data: Concepts, Theory, and Methods. Wiley, New York.

Cohen WW and Singer Y (1996) Context-sensitive learning metods for text categorization. In: SIGIR'96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 307–315.

Collett D (1991) Modelling Binary Data. Chapman and Hall, London.

Dagan I, Feldman R and Hirsh H (1996) Keyword-based browsing and analysis of large document sets. In: Fifth Annual Symposium on Document Analysis and Information Retrieval (SDAIR'96), pp. 191–208.

D'Alessio S, Kershenbaum A, Murray K and Schiaffino R (1998) Hierarchical text categorization. Technical Report, Department of Computer Science, Polytechnic University, Brooklyn, NY.

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.

Duda RO, Hart PE and Stork DG (1999) Pattern Classification and Scene Analysis, Part I: Pattern Classification. Wiley, New York.

Haykin SS (1998) Neural Networks: A Comprehensive Foundation. Prentice Hall.

Hertz J, Krogh A and Palmer RG (1991) Introduction to the Theory of Neural Computation. Addison-Wesley, Reading, MA.

Hofmann T (1998) Learning and representing topic: A hierarchical mixture model for word occurrences in document databases. In: Conference for Automated Learning and Discovery, Workshop on Learning from Text and the Web (CMU).

Hull D (1993) Using statistical testing in the evaluation of retrieval experiments. In: Proceedings of the 16th Annual ACM/SIGIR Conference, pp. 329–338.

Hull D (1994) Improving text retrieval for the routing problem using latent semantic indexing. In: Proceedings of the 17th Annual ACM/SIGIR Conference, pp. 282–291.

Ittner DJ, Lewis DD and Ahn DD (1995) Text categorization of low quality images. In: Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval, pp. 301–315.

Kennedy RL, Lee Y, Roy BV and Reed C (1998) Solving Data Mining Problems Through Pattern Recognition. Prentice Hall.

Koller D and Sahami M (1997) Hierarchically classifying documents using very few words. In: Proceedings of the 14th International Conference on Machine Learning (Nashville, Tennessee), pp. 170–178.

LeBaron B and Weigend AS (1998) A bootstrap evaluation of the effect of data splitting on financial time series. IEEE Transactions on Neural Networks, 9(1):213–220.

Lewis DD (1992) Representation and Learning in Information Retrieval. Ph.D. Thesis, Computer Science Department, Univ. of Massachussetts at Amherst.

Lewis DD and Ringuette M (1994) A comparison of two learning algorithms for text categorization. In: Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval, pp. 81–93.

Masand B, Linoff G and Waltz D (1992) Classifying news stories using memory based reasoning. In: Proceedings of the 15th Annual ACM/SIGIR Conference, pp. 59–65.

McCullagh P and Nelder JA (1989) Generalized Linear Models. Chapman and Hall, London.

Nilsson NJ (1998) Artificial Intelligence: A New Synthesis. Morgan Kaufmann.

Pereira F, Tishby N and Lee L (1993) Distributional clustering of English words. In: Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, pp. 183–190.

Rocchio JJ (1971) Relevance feedback in information retrieval. In: The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice Hall, chap. 14, pp. 313–323.

Rose K, Gurewitz E and Fox GC (1990) Statistical mechanics and phase transitions in clustering. Physical Review Letters, 65:945–948.

Rumelhart DE, Durbin R, Golden R and Chauvin Y (1996) Backpropagation: The basic theory. In: Smolensky P, Mozer MC and Rumelhart DE, eds. Mathematical Perspectives on Neural Networks. Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 533–566.

Russell SJ and Norvig P (1995) Artificial Intelligence: A Modern Approach (Prentice Hall Series in Artificial Intelligence). Prentice Hall, Englewood Cliffs, NJ.

Schuetze H, Hull DA and Pedersen JO (1995) A comparison of classifiers and document representations for the routing problem. In: Fox EA, Ingwersen P and Fidel R, eds. Proceedings of the 18th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, pp. 229–237.

van Rijsbergen CJ (1979) Information Retrieval, 2nd ed. Butterworths.

Vapnik VN (1998) Statistical Learning Theory (Adaptive and Learning Systems for Signal Processing, Communications, and Control). Wiley, New York.

Wiener ED (1995) A neural network approach to topic spotting in text. Master's Thesis, Department of Computer Science, University of Colorado at Boulder `www.stern.nyu.edu/~aweigend/Research/Papers/TextCategorization/Wiener_Thesis95.ps`.

Wiener ED, Pedersen, JO and Weigend AS (1995) A neural network approach to topic spotting. In: Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95), pp. 317–332. `www.stern.nyu.edu/~aweigend/Research/Papers/TextCategorization/Wiener.Pedersen.Weigend_SDAIR95.ps`.

Yang Y (1994) Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In: Proceedings of the 17th Annual ACM/SIGIR Conference, pp. 13–22.

Yang Y (1999) An evaluation of statistical approaches to text categorization. Information Retrieval.

Yang Y and Chute CG (1992) A linear least squares fit mapping method for information retrieval from natural language texts. In: Proceedings of the 14th International Conference on Computational Linguistics, pp. 447–453.

Yang Y and Chute CG (1994) An example-based mapping method for text categorization and retrieval. In: ACM Transaction on Information Systems (TOIS), pp. 252–277.