# A Comparative Study for Domain Ontology Guided Feature Extraction

**Bill B. Wang, *R* I. *(Bob)* M*c*K*a*y, Hussein A. Abbass, Michael Barlow**

School of Computer Science, University College, ADFA, University of New South Wales
Canberra, ACT 2600

{biaowang, rim, abbass, spike}@cs.adfa.edu.au

## Abstract

We introduced a novel method employing a hierarchical domain ontology structure to extract features representing documents in our previous publication (Wang 2002). All raw words in the training documents are mapped to concepts in a concept hierarchy derived from the domain ontology. Based on these concepts, a concept hierarchy is established for the training document space, using is-a relationships defined in the domain ontology. An optimum concept set may be obtained by searching the concept hierarchy with an appropriate heuristic function. This may be used as the feature space to represent the training dataset. The proposed method aims to solve some drawbacks suffered by text classification algorithms and feature selection algorithms. In this paper, we conducted a series of experiments to compare our approach with other comparable feature-selection and feature-extraction methods. The results indicated that our approach has advantages in many aspects.

*Keywords*: text classification, ontology, concept hierarchy, principal component analysis, KNN algorithm, information gain, $\chi^2$ statistics.

## 1. Introduction

With the large amount of papers that exist in organisations, it is becoming vital to automatically import these papers into the Computer. Optical Character Recognition (OCR) software is quite efficient in transforming typed documents. However, after storing millions of documents in databases, the question arises of how to assign these documents to specialised databases. On a more technical level, the question becomes how to classify these documents. Automatic text classification (Salton 1989) is the task of assigning natural language texts to one or more pre-defined categories based on their content.

Compared with common data classification tasks, text classification presents unique challenges due to the large and unfixed number of features present in the dataset, large number of documents, and multi-modality of categories. Existing classification techniques have limited applicability in these datasets because the large numbers of features make most documents undistinguishable in higher dimensional spaces.

Many researchers have shown that similarity based classification algorithms, such as KNN and centroid based classification, are very effective for large document collections (Shankar 2000). A cross-experiment comparison (Yang 1999) between 14 major classification methods, including KNN, decision tree, naive Bayes, linear least squares fit, neural network, SWAP-1, Rocchio, etc., has shown that KNN is one of the top performers, and it performs well in scaling up to very large and noisy classification problems. However, these effective classification algorithms still suffer disadvantages from high dimensionality that greatly limit their practical performance. Empirical and mathematical analysis (Beyer 1999, Hinneburg 2000) has shown that finding the nearest neighbors in high-dimensional space is very difficult because most points in high-dimensional space are almost equi-distant from all the other points.

In fact, in many document data sets, only a relatively small number of the total features may be useful in classifying documents, and using all the features may adversely affect performance. So determining how to reduce the length of document vectors effectively and reasonably is a challenge for classification researchers. Stop words lists (Fox 1992) and word stemming (Frakes 1992) are some of the earliest efforts in this problem. In recent years, many term-weighting and feature-selection algorithms (Lewis 1994, Yang 1997, Shankar 2000, John 1994, Kira 1992) have been developed, to reduce the feature space without sacrificing remarkable classification accuracy. However, the effectiveness of these algorithms heavily depends on the quality of the training dataset. This is a major drawback for text classification methods, as the creation of high quality datasets may be very expensive.

The performances of both the text classification algorithms discussed above, and of feature selection algorithms, depend on the quality of training dataset. The KNN classifier is an instance-based classifier, which means a training dataset of high quality is particularly important. An ideal training document set for each particular category will cover all the important terms, and their possible distribution in this category. With such a training set, a classifier can find the true distribution model of the target domain. Otherwise, a text that uses only some key words out of a training set may be assigned to the wrong category. In practice,

however, establishing such a training set is usually infeasible. In practice, a perfect training set can never be expected.

In our previous work (Wang 2002), we introduced a novel method to effectively and reasonably reduce the dimensionality and improve the performance of a text classifier. By searching the concept hierarchy defined by a domain-specific ontology, a more precise distribution model for a pre-defined classification task can be determined. The experiments indicated that, by using this approach, the size of the feature sets can be effectively reduced and the accuracy of the classifiers can be increased. In this paper, we will compare our approach with some other dimensionality-reduction methods through a series of comparison experiments.

This paper is structured as follows: Section 2 briefly introduces the related work, Section 3 introduces the notion of domain-specific concept ontology and UMLS knowledge resources, Section 4 describes the process of this system, some experimental results and discussions are presented in Section 5, finally the conclusion is given in Section 6.

## 2. Related Work

In this section, we briefly review some background research including unsupervised and supervised dimensionality reduction applied to document datasets, some previous attempts to apply semantic knowledge in unsupervised and supervised learning, and our work.

There are several methods for reducing the dimensionality of high-dimensional data in an unsupervised learning model. Most of these methods reduce the dimensionality by combining multiple variables or features, utilizing the dependencies among the variables detected by statistical tests. Consequently, these techniques can capture synonyms in the document datasets. These methods are also called feature extraction.

Principal Component Analysis (PCA) (Calvo 1998) is a key method. Given an $n \times m$ document-term matrix, PCA uses the first eigenvectors of the $m \times m$ covariance matrix as the axes of the lower $k$-dimensional space. These leading eigenvectors correspond to linear combinations of the original variables that account for the largest amount of term variability (Jackson 1991). Latent Semantic Indexing (LSI) (Deerwester 1990) is a dimensionality reduction technique extensively used in the information retrieval domain and is similar in nature to PCA. In LSI, instead of finding the truncated singular value decomposition of the covariance matrix, the method finds the truncated singular value decomposition of the original $n \times m$ document-term matrix, and uses these singular eigenvectors as the axes of the lower dimensional space. Experiments have shown that LSI substantially improves the retrieval performance on a wide range of datasets (Dumais 1995). However, the reason for LSI's robust performance is not well understood, and is currently an active area of research (Papadimitriou 1998).

In effect, these methods try to find the semantic relationships between features using statistical tests. A key problem is that their performances depend strongly on the sufficiency and the quality of the training dataset. Most importantly, their discovery cannot extend to features that do not occur in the training dataset. In principle, all of the methods developed for unsupervised dimensionality reduction can potentially be used to reduce the dimensionality in a supervised model as well. However, in doing so, they cannot take advantage of the class or category information available in the dataset. Another limitation of these methods in supervised data is that characteristic variables that describe smaller classes tend to be lost as a result of dimensionality reduction. Hence, the classification accuracy on smaller classes in the reduced dimensional space can be quite poor. On the other hand, stratified sampling to avoid this problem can result in poor classification accuracy on the larger classes.

Various feature selection methods have been developed for supervised dimensionality reduction (Kira 1992, Karypis 2000, Yang 1997, Moore 1997, Liu 1998b, Kohavi 1997). A number of researchers have recently addressed the issues of feature subset selection in machine learning. As noted by John, Kohavi and Pfleger (John 1994), this work is often divided along two lines: the filter approach and wrapper approach. In the filter approach, feature selection is performed as a preprocessing step to induction. Thus, the bias of the learning algorithm does not interact with the bias inherent in the feature selection algorithm. The main disadvantage of the filter approach is that it totally ignores the effects of the selected feature subset on the performance of the induction algorithm. In the filter approaches, the different features are ranked using a variety of criteria, and then only the highest-ranked features are kept. A variety of techniques have been developed for ranking the features (*i.e.*, words in the collection) including document frequency (number of documents in which a word occurs), mutual information (Cover 1991, Yang 1997, Joachims 1997), and $\chi^2$ statistics (Yang 1997). Consequently, even though the criteria used for ranking is the measure of the effectiveness of each feature in the classification task, these criteria may not be optimal for the classification algorithm used. Another limitation of this approach is that these criteria measure the effectiveness of a feature independent of other features, and hence features that are effective in classification only in conjunction with other features will not be selected. In contrast to the filter approaches, wrapper approaches find a subset of features using a classification algorithm as a black box (Kohavi 1997, Kohavi 1995, Liu 1998b). In these approaches the features are selected based on how well they improve the accuracy of the classifier. The wrapper approaches have been shown to be more effective than the filter

approaches in many applications (Kohavi 1995, Wettschereck 1997, Langley 1994). However, the major drawback of these approaches is that their computational requirements are very high. This is particularly true for document datasets where there are thousands of features.

To date, there have been few efforts to apply semantic information in text classification.

- Koller and Sahami (Koller 1997) proposed an approach that utilizes the semantic information provided by the hierarchical topic structure to decompose the classification task into a set of simpler problems, one at each node in the classification tree. Their experiments indicate that each of these smaller problems can be solved accurately by focusing only on a very small set of features, those relevant to the task at hand. This set of relevant features varies widely throughout the hierarchy, so that, while the overall relevant feature set may be large, each classifier only examines a small subset.
- Hotho, Staab and Maedche (Hotho 2001) proposed a semantic approach for document clustering. They apply background knowledge during preprocessing in order to improve clustering results and allow for selection between results. In their experiments, all terms occurring in documents were mapped to concepts. They built various views, basing the selection of text features on a hierarchy of concepts. Their results indicate that this approach compares favorably with baselines, such as clustering based on terms by tf/idf measures. The selected concepts may be used to indicate, to the user, which text features were most relevant for the particular clustering results, and to distinguish different views.

## 3 Domain-Specific Concept Ontology and UMLS Knowledge Resources

The term ontology has various meanings when it is used in different ways and in different disciplines. However computer scientists use the term ontology to describe formal descriptions of objects in the world, the properties of those objects, and the relationships among them. In artificial intelligence, according to Gruber (Gruber 1993), an ontology is a specification of a conceptualization. It defines the vocabulary of a domain, and constraints on the use of terms in the vocabulary.

In our research, a *term* is a sequence of alpha-numeric characters which is delimited by white space or punctuation marks. A *domain-specific concept ontology* specifies the concepts that are used to represent documents. A *concept* represents a unit of meaningful information in this domain. A concept may consist of one or more terms. A domain ontology also specifies the categories attached to these concepts, and the relations (ISA in this paper) which exist between concepts and categories (Figure 1). The hierarchical concept structure, which we use for a particular

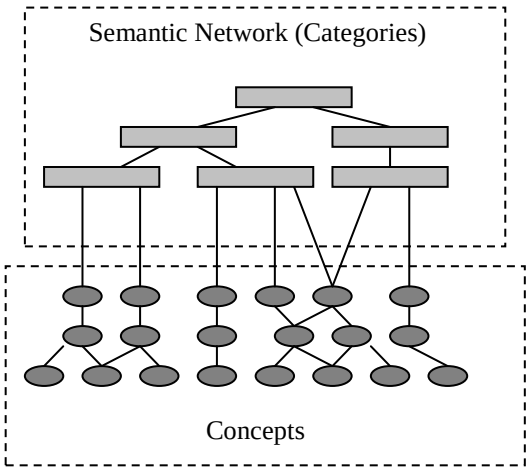


**Figure 1. A Sketch of a Domain-Specific Concept Ontology**

training document set, is a part of a domain-specific concept ontology based on terms used in the training set. The process to establish this structure is introduced in Section 4.

The Unified Medical Language System (UMLS), a set of knowledge sources developed by the US National Library of Medicine, can be viewed as a complete concept ontology for medical domains. It consists of three sections: a metathesaurus, a semantic network and a specialist lexicon; and contains information about medical terms and their inter-relationships. It is organized by concept, and contains over 800,000 concepts and 1.9 million entries. Various types of relationships between concepts are defined in this system. ISA is the primary relationship. We used this relationship to establish the hierarchical concept structure for a particular training set containing documents in the medical domain.

## 4 Establishing Concept Representation

There are four major steps to establishing a concept representation for documents.

1. Map raw terms to concepts based on UMLS
2. Establish a concept hierarchy for the training set
3. Search the concept hierarchy to obtain the optimal concept set
4. Establish a new feature model for both training and test documents

### 4.1 Mapping Raw Terms to Concepts

The most straightforward representation of documents relies on term vectors. The major drawback of this basic approach for document representation is the length of the feature vectors, usually more than 10,000 terms. In the application of text categorisation, however, completely different terms may represent the same concepts. In some cases, terms with different concepts can even be replaced with a higher level concept without negative effect on performance of the

classifier. For example, ANEMIA and LEUKEMIA can be replaced with the higher level concept HEMATOLOGIC DISEASE, in many situations of text categorization, without loss of the classifiers' accuracy. Obviously, mapping terms to concepts is an effective and reasonable method to reduce the dimensionality of the vector space.

The mapping process relies on the API provided by the UMLS system. We use the mapping function provided by the UMLS query interface. We aim to find the 'longest concept units' (LCUs) in documents. An LCU is an independent concept defined by a string of continuous terms such that any other string of continuous terms, which contains this string, does not define an independent concept. For example, consider the sentence
*AIDS is a kind of human immunodeficiency virus.*
According to the mapping algorithm defined below, we will get two concepts: 'AIDS' and 'HIV'. 'Human' and 'virus' are not recorded as independent concepts, even though they do occur as independent concepts in the concept ontology, because they are part of the LCU 'HIV'. We take this approach because an LCU is usually more meaningful for identifying the content of a document.

**Term to Concept Mapping Algorithm:**
**Input:** a sentence consisting of n terms $\Phi = \{t_1, t_2, \ldots, t_n\}$
a concept set C = ∅
**while**($\Phi \neq \emptyset$)
$\Phi_t \leftarrow \Phi$
**while**($\Phi_t \neq \emptyset$)
c ← mapping($\Phi_t$)
**if** c ≠ **Null**
**then** put c in C, remove $\Phi_t$ from $\Phi$, $\Phi_t \leftarrow \emptyset$
**else** remove $t_i$ from $\Phi_t$ ($i = |\Phi_t|$)
**loop**
**loop**
**return** C as the concept set for this sentence

Through this mapping process, we will obtain concept sets for individual documents. Each will include all distinct concepts. The frequency of the concepts is recorded. Combining such data for all training documents, we will also obtain a shared concept set for the training set, which includes all distinct concepts from the whole document set.

## 4.2 Establishing the Concept Hierarchy

The UMLS query interface provides a parent query function for retrieving parents of concepts. The concept hierarchic structure is established by repeatedly querying parent' from shared concepts up to the root of the semantic network. The completed concept hierarchic structure is a fully-connected graph rooted at 'top_type'.
For instance, suppose that there are only five distinct concepts as below occurring in a document set.

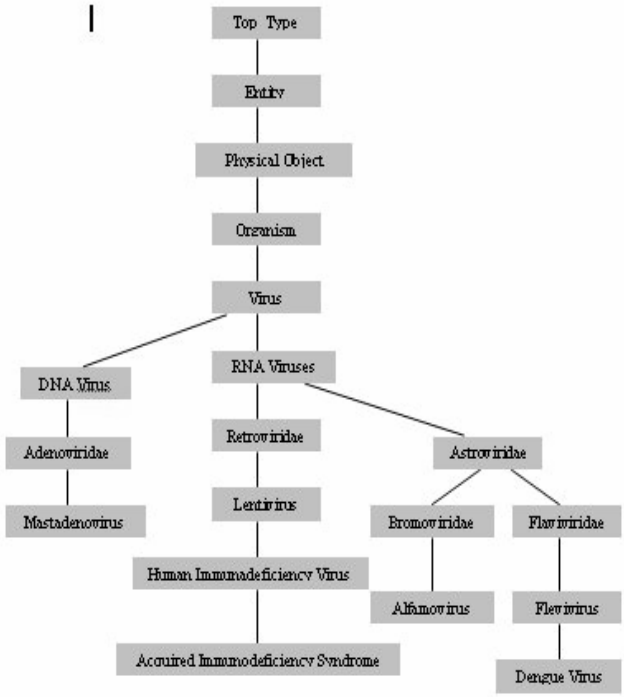


**Figure 2. A Sample Concept Hierarchical Structure**

[*Mastadenovirus, AIDS, Human Immunodeficiency Syndrome, Alfamovirus, Dengue Virus*]
Based on this shared concept set, we will get the concept hierarchic structure in Figure 2. From this structure, we can see that several combinations of different level concepts can be chosen to represent documents for different taxonomic standards, e.g. [Virus], [DNA Virus, RNA Virus] and [DNA Virus, Retroviridae, Astroviridae]. All original concepts can be mapped to these higher level concepts.

Then, when a new concept (e.g. adenoviridae) occurs in the fresh documents, it can be easy mapped to 'DNA Virus' for classification algorithms. This new concept would be ignored in most traditional classifiers because it never occurs in the training data set.

## 4.3 Search Concept Hierarchy

In this paper, we use a hill-climbing algorithm to search the concept hierarchical structure obtained in the previous step to find the optimum representation (a set of concepts) for a particular document set. Our aim is to use a set of concepts to represent training documents which is as high in the concept hierarchy as possible without loss of categorization accuracy.

First, we specify that all concept nodes, except the root node, have an out edge to their parent nodes. Then we establish a copy of the hierarchical structure for each document. We assign the frequency of each concept occurring in the document to the edge leading into the parent concept node. Thus the frequency for the edge leading out of a parent concept node is the sum of the edge frequencies of all child nodes.

The vital problem is to define an appropriate heuristic function for the hill climbing search algorithm. In this model, each document d is considered to be a vector in the concept-space. In its simplest form, each document is represented by the concept-frequency (CF) vector,

$$\vec{d} = (cf_1, cf_2, ..., cf_n)$$

where $cf_i$ is the frequency of the ith concept in the document.

In order to account for documents of different lengths, each frequency is normalized by dividing by the document length.

In the vector-space model, the similarity between document *i* and document *j* is commonly measured using the cosine function, given by

$$s_{ij} = \cos(\vec{d}_i, \vec{d}_j) = \frac{\vec{d}_i \cdot \vec{d}_j}{\| \vec{d}_i \|_2 \quad \| \vec{d}_j \|_2}.$$

Since the document vectors are of unit length, the above formula simplifies to

$$s_{ij} = \cos(\vec{d}_i, \vec{d}_j) = \vec{d}_i \cdot \vec{d}_j .$$

Finally, we define the heuristic function ( $f$ ) for the hill climbing search algorithm as below.

$$f = (1 + \frac{\alpha - n}{\alpha} \beta) \sum_{i \in D} | D_{ci} |, \quad D_{ci} \subseteq D_{KNNi} \quad \beta > 0$$

D is the set of all documents in the training set. $D_{ci}$ is the set of all documents that belong to the same category, which contains document i, in a particular document set. $D_{KNNi}$ is the set of k nearest neighbors for document *i* in the training set. n is the dimensionality of feature space. $\alpha$ is the number of leaf nodes. $\beta$ is a constant. The first part of the right side of the equation is a reward factor, intended to encourage the use of high level concepts in the feature set with despite a limited loss of categorization accuracy. We suggest that $\beta$ is chosen less than 0.05. This means that the effect on heuristic value resulting from the reward is at most 5%. The value of k depends on the size of the document collection. We do not suggest that a large value of k is used. This is because a large k may result in unbalanced performance on different categories. In the future, a further study about the k value sensitivity of the performance of classifiers will be conducted.

We define our bottom-up hill climbing search algorithm as follows.

**Initial status**: current_concept_set $\Phi_{ccs}$ = {all leaf concepts in concept hierarchy}
heuristic $f_{opt} = f(\Phi_{ccs})$
Temporal_ concept_set ($\Phi_{tcs}$) = ∅
**while(**has more unmarked concept c in $\Phi_{ccs}$)
$\Phi_{tcs} \leftarrow \Phi_{ccs}$
take an unmarked concept $c \in \Phi_{tcs}$
find parent concept node ($c_p$) of c in concept hierarchy
use parent $c_p$ to replace c in $\Phi_{tcs}$
remove all child concepts of $c_p$ from $\Phi_{tcs}$
$f = f(\Phi_{tcs})$
**if** $f > f_{opt}$
**then** $\Phi_{ccs} \leftarrow \Phi_{tcs}$, $f_{opt} \leftarrow f$
**else** mark c in $\Phi_{ccs}$
**loop**
**return:** $\Phi_{ccs}$ as optimal concept set

### 4.4 Establish new Feature Model

Based on the optimum concept set obtained from the previous step, we can establish a new feature model for training documents. All training documents are represented by concept vectors, that is, all original concepts are mapped to the optimum concepts.

For new documents, we also map the terms to concepts using the same process. Then we add all the new concepts that did not occur in the training set to the concept hierarchy. We can also represent testing documents as concept vectors based on the optimum concept set.

## 5 Experiment Setup and Results

In this section we experimentally evaluate the effect on the KNN classifier of using the domain-specific concept hierarchy to guide feature selection. In our experiments, we compare the performance of our feature selection method against the performance achieved by a common KNN classifier and other feature-extraction methods. Also, we study the effect of the training set size on the performance of these methods.

The common KNN classification experiments used the RAINBOW system (McCallum 1996), which includes stop word removal, stemming and feature selection. The principal components were calculated by SPSS 11.

### 5.1 Document Collection

We chose documents from 10 journals of the MEDLINE database (PUBMED) to form our training and test document sets as in Table 1. Every document is labeled by the name of the journal that contains the document. The subjects of these documents are obviously independent of each other so they can be viewed as reasonable pre-existing categories. 150 documents were chosen randomly from each journal by people without specialized medical knowledge. 50 of 150 documents from each category were randomly chosen as test set and the remaining 100 formed the training set except where otherwise specified. For one set of experiments comparatively studying the effect of the training set size, we formed specific-size subsets by randomly choosing the same number of documents from each category.
The size of the test set was never varied in the experiments. We used title plus abstract as the text for our experiments.

| Journal Name | Category Name | Covered years |
|---|---|---|
| Addiction (Abingdon, England) | Addiction | 1999, 2000 |
| AIDS Care | AIDS | 1998, 1999, 2000, 2001 |
| American Heart Journal | Heart | 2000, 2001 |
| Cancer Research | Cancer | 1999, 2000 |
| The British Journal of Ophthalmology | Ophthalmology | 2000, 2001 |
| Burns: journal of the International Society for Burn Injuries | Burns | 2000, 2001 |
| Bone | Bone | 2000, 2001 |
| Epilepsy research | Epilepsy | 1999, 2000, 2001 |
| Diabetes | Diabetes | 1999, 2000 |
| Clinical and experimental dermatology | Dermatology | 1999, 2000 |

**Table 1: Details of document collection**

| Training Size | Distinct Terms | Distinct Concepts | Optimum Concepts |
|---|---|---|---|
| 1,000 | 16,163 | 4,645 | 1,634 |
| 750 | 14,046 | 4,034 | 1,540 |
| 500 | 11,454 | 3,182 | 1,327 |
| 300 | 8,948 | 2,436 | 942 |
| 200 | 7,634 | 1,894 | 695 |

**Table 2: Statistical information concerning the training set**

| Category Name | Term-based | Original Concepts | Optimum Concepts |
|---|---|---|---|
| Addiction | 100% | 96% | 94% |
| AIDS | 92% | 92% | 90% |
| Heart | 96% | 92% | 92% |
| Cancer | 96% | 90% | 94% |
| Ophthalmology | 72% | 80% | 84% |
| Burns | 58% | 66% | 70% |
| Bone | 66% | 68% | 74% |
| Epilepsy | 74% | 72% | 78% |
| Diabetes | 80% | 82% | 88% |
| Dermatology | 30% | 52% | 58% |
| Overall | 76.4% | 79.0% | 82.2% |
| STD (Overall) | 0.2162 | 0.1424 | 0.1198 |

**Table 3: Comparison of accuracy in default training set**

### 5.2 Accuracy Measure

To evaluate the trained classifier on test documents for each class, we defined an accuracy measure as follows. It is consistent with that used by RAINBOW system.

$$Accuracy = \frac{correctly \quad assigned \quad documents}{total \quad candidate \quad documents}$$

This accuracy can be used to measure the performance of the classifiers on each particular category. "correctly assigned documents" means all documents which are correctly assigned to the particular category. "total candidate documents" means all test documents which should be assigned to the particular category. The overall performance can be measured in the same way. Since every document in our data set has only one category label, the 'recall' measure is not considered in our experiments.

### 5.3 Document pre-processing

By pre-processing the training documents using the two methods separately, we derived statistical information about our training set as in Table 2.

The number of distinct terms was obtained by using the mapping process and the number of optimum concepts was obtained by using the search algorithm we introduced above. For the heuristic function, $\beta$ of 0.05 and k of 5 were used.

As we see in Table 2, even using the original concept set, compared with the term set, causes a significant reduction in dimensionality of feature space.

### 5.4 Performance comparison between the two approaches

In this section, we used the default training set to compare the effect of different feature sets on performance of KNN classifier. Category ranking in KNN is based on the categories assigned to the k

nearest training documents to the test document. The categories of these neighbors are weighted using the similarity of each neighbor to the cosine between the two document vectors. The reasons for choosing KNN in our experiments are the same as those of Yang and Pedersen (1997): it is one of top-performing classifier; it is a context-sensitive classifier that enables a better observation on feature selection.

Table 3 shows the performance of the classifier on the 10 categories. A desirable classifier should have balanced performance for the pre-defined categories in the training set. Therefore, we computed the standard deviation (STD) for performance of the categories. It is possible that different values of k might be needed to achieve optimal performance for the different methods. For each method, therefore, we tried three values (5, 10, 15) of k, and the best results are reported in the result table.

A number of interesting observations can be made from the results in Table 3. First, compared with the term-based classifier, the overall performances achieved by the original concept model and the optimum concept model increased from 76.4% to 79% and 82.2% respectively, or a 3.4% and a 7.6% increase relatively respectively. Second, we see that our method smoothes the performance of the classifier on different categories. Compared with the term-based classifier, the values of STD for two concept-based classifiers have a 31.4% and a 44.6% relative decrease respectively.

## 5.5 Effect of feature size on performance

In this section, we apply feature selection methods to documents in the pre-processing of a term-based KNN classifier. Through this experiment, we may study the effect of statistics-based feature selection methods on performance of term-based KNN classifier using the default training set. The RAINBOW system provides a feature selection function using the information gain method. A comparative study on feature selection methods (Yang 1997), including document frequency thresholding (DF) (Yang 1997), information gain (IG) (Mitchell 1996), mutual information (MI) (Yang 1997, Wierner 1994), $\chi^2$ statistic (CHI) (Yang 1997), and term strength (TS) (Yang 1995, Wilbur 1992), shows that IG, DF and CHIMAX have similar effects on performance of the classifiers and all are better than the other two. Therefore, the experimental results may, in a sense, provide information on how the statistics-based feature methods affect the performance of a KNN classifier on our dataset.

The influence of the information gain method and CHIMAX method are evaluated using the overall accuracies of the classifier and the STD of accuracies of individual categories. Figure 3 displays the four curves for the term-based KNN classifier on the default training set.

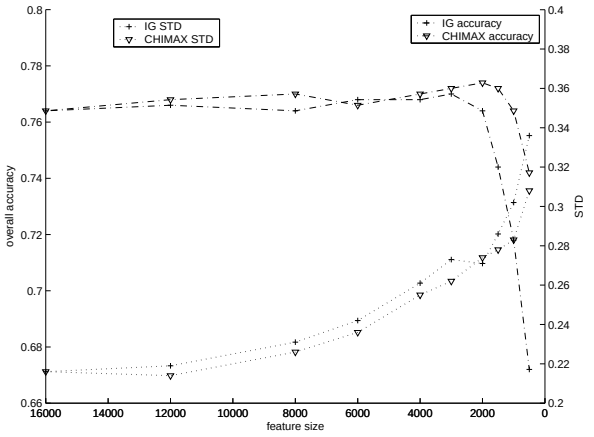


**Figure 3: Overall accuracy and STD vs. feature sizes obtained by feature selection**

An observation emerges from the categorization results of Figure 3. That is, although the performance of the classifier in terms of overall accuracy has not significantly declined until 90% of distinct terms are eliminated, the values of STD starts a clear increase once half the terms are removed for both selection methods. This means that we expect an imbalance of categorization accuracy when the information gain method is employed to reduce the dimensionality of the feature space for a KNN classifier on our dataset.

## 5.6 Effect of training set size on performance

A comparative experiment measuring the performance against the size of training set was conducted using training sets of different sizes listed in table 2. The optimum concept sets were discovered for training sets of different sizes. The feature selection algorithm was not used for this experiment because it does not improve the performance in terms of accuracy. Moreover, it is difficult to define a selection threshold for training sets of different sizes. The experimental results are shown in Figure 4.

When the size of training set increased from 200 to 1000, the accuracy of concept-based classifier increased from 72.2% to 82.2%, or a 13.9% increase relatively, and the accuracy of term-based classifier increased from 66.8% to 76.4%, or a 14.4% increase relatively. In addition to this, another interesting observation can be made from Figure 4. We divide this process into two stages. In the first stage, when the size of training set increased from 200 to 500, the accuracy of the concept-based classifier increased from 72.2% to 79.8%, or a 10.5% increase relatively, and the accuracy of the term-based classifier increased from 66.8% to 72.0%, or a 7.8% increase relatively. In other words, in this stage, the gradient of the accuracy of the concept-based classifier is 34.6% larger than that of the term-based classifier. In contrast, in the second stage, when the size of training set increased from 500 to 1000, the gradient of the accuracy of the term-based classifier is 103.3% larger than that of the concept-based classifier. It seems to indicate that the accuracy of the concept-

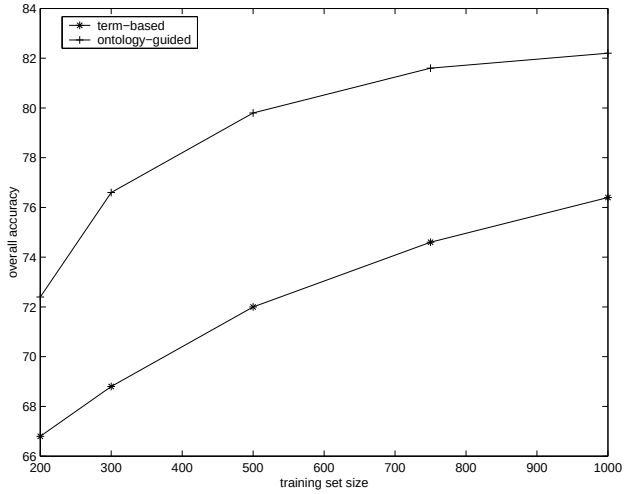


**Figure 4. Overall accuracy of KNN classifiers vs. training set size**

based classifier converges faster than the term-based classifier.

## 5.7 Principal Component Analysis

Comparative experiments (Fuka 2001) indicate that PCA give better classification accuracy than other feature reduction method, such as $\chi^2$ statistics and self organizing map (SOM) clustering (Honkela 1997). As a method of feature extraction, our approach is more similar to PCA. It projects lower level concepts to high level concepts rather than selecting a subset of the original feature set. In order to examine this relationship, we also conducted a series of experiments to compare PCA with our approach.

Due to the high memory requirement of the PCA algorithm, the 500 training set was chosen for comparison experiments. The PCA method was applied on the original concept set (3182 features) and the optimized concept set (1327 features) and their impact on the overall classification accuracy was measured respectively for sets of different sizes. Figure 5 contains experiment results.

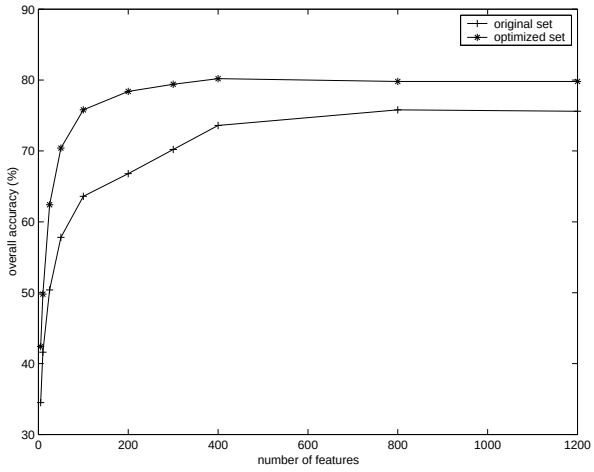


**Figure 5: Overall accuracy vs. number of principal components**

With the increase of the number of principal components, the accuracy of the classifier based on the optimized concept set increased much faster than the one based on the original concept set. When the number of principal components was greater than 200, the classifier based on optimized concept set gave stable results of accuracy which was close to the results from the classifier without using PCA method. The stable point for the classifier based on original concepts was about 400 principal components since which the classifier gave similar accuracy with the classifier without using PCA.

## 6 Discussion and Conclusion

The Experimental results show that seeking the optimum concept set in a concept hierarchical structure is a highly viable method. It enables us to reduce the length of document vectors effectively and reasonably. The experimental results also show that this approach, guided by the domain ontology, effectively improves the accuracy of a KNN classifier. Also, it indicates other advantages compared with currently dominant dimensionality-reduction methods.

From the experimental results, two other impressive characteristics of ontology guided feature selection are

- more balanced performance on individual categories
- faster convergence of classification accuracy against the size of training set

When feature selection is used to reduce the dimensionality of a feature space, the standard deviation of accuracies of individual categories has a faster rate of increase than the rate of decrease of overall accuracy. Our experiments are not sufficient to make a judgment that all statistics-based feature selection methods will lead to unbalanced performance on individual categories in all datasets. However, all statistics-based feature selection methods have a common feature. They involve searching for an optimum subset of features based on term-goodness criteria (e.g. information gain) by discarding uninformative features. Thus their effectiveness depends on the quality of the training set. In a sense, the feature selection problem can be viewed as a special case of the feature weighting problem. No matter which feature selection method is applied, accurately assigning a score to a distinct term depends on whether enough information is provided by the training set. In most situations, the key terms for the individual categories of different natures are assigned scores distributed across a large range. Due to noise or insufficiency of training documents, some non-informative terms may be assigned high weights. Therefore, some terms important to identify the content of some categories are removed earlier than the key terms of other categories and even some non-informative terms. Rather than simply removing terms, the ontology guided feature selection uses the parent

concept to replace child concepts that have the same nature, and is thus less affected by noise.

PCA projects the data on a new space such that the variance in the projected data is maximized. In other words, fewer variables are used to cover variance in the projected data as much as possible. The empirical results showed that PCA is an effective dimensionality-reduction method for both the original concept set and the optimized concept set. However, we can see that PCA could not improve the classification accuracy for both sets. This is attributable to the same reason we mentioned above. In addition, as a post-process of ontology guided feature-extraction, PCA can perform more effectively.

KNN is an instance-based classifier. The performance of instance-based classifiers is more dependent on sufficiency of training set than that of other machine learning classification algorithms. KNN treats documents as individual points in a vector space. Each point is defined by a set of non-zero features of feature space. A smaller training set implies that more terms or term combinations important for content identification may be missing from the training documents. This will negatively affect the performance of a classifier. The domain ontology guided approach can somewhat reduce the negative influence of this problem. We discover the optimum concept set in the concept hierarchy by replacing child concepts with parent concepts when this does not adversely affect performance. Therefore an important term, which resides low in the concept hierarchy, may be mapped to a concept in the optimum concept set, even if this term is not included in the training set.

## 7 Acknowledgments

We would like to thank the U.S. National Library of Medicine for having provided the UMLS knowledge resources.

## 8 References

Beyer, K., Goldstein, J., Ramakrishnan, R. and Shaft, U. (1999): When is 'nearest neighbour' meaningful? *Proc. of ICDT-1999*, Jerusalem, Israel, pages 217-235.

Calvo, R. A. and Partridge, M. (1998): A Comparative Study of Principal Component Analysis Techniques. *Proc. Ninth Australian Conf. on Neural Networks*, Brisbane, QLD.

Cover, T. M. and Thomas, J. A. (1991): *Elements of Information Theory*. John Wiley & Sons.

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

Dumais, S.T. (1995): Using LSI for information filtering: TREC-3 experiments. *Proc. of the Third Text Retrieval Conference (TREC-3)*. National Institute of Standards and Technology.

Fox, C. (1992): Lexical Analysis and Stoplists. In *Information Retrieval: Data Strucure & Algorithms*. 102-130. Frakes, W. B. and Baeza-Yates, R. (eds). P T R Prentice Hall.

Frakes, W. B. (1992): Stemming Algorithms. In *Information Retrieval: Data Strucure & Algorithms.* 131-160. Frakes, W. B. and Baeza-Yates, R. (eds). P T R Prentice Hall.

Fuka, K. and Hanka, R. (2001): Feature Set Reduction for Document Classification Problems. *IJCAI-01 Workshop: Text Learning: Beyond Supervision*. Seattle (August 2001), USA.

Gruber, T. (1993): A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:pp. 199-220.

Heckerman, D. (1995): A tutorial on learning with Bayesian networks. *Technical Report MSR-TR-95-06*, Microsoft Research.

Hinneburg, A., Aggarwal, C. C. and Keim, D.A. (2000): What is the nearest neighbour in high dimensional spaces? *Proc. of the International Conference on Very Large Databases (VLDB)*, pp. 506--515, Cairo, Egypt, Sept. 2000. Morgan Kaufmann.

Honkela, T. (1997): Self-Organizing Maps in Natural Language Processing. *Ph.D. Dissertation*, University of Technology, Helsinki.

Hotho, A., Staab, S. and Maedche, A. (2001): Ontology-based Text Clustering. *IJCAI '01 – Workshop "Text Learning: Beyond Supervision", Seattle*, USA.

Jackson, J. E. (1991): *A User's Guide To Principal Components*. John Wiley & Sons.

Joachims, T., "A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization", In Proc. of the Fourteenth International Conference on Machine Learning, 1997.

John, G., Kohavi, R., and Peleger, K. (1994): Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the Eleventh International Conference Morgan Kaufmann*. pp. 121-129.

Karypis, G. and Han, E.H. (2000): Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval & categorization. *Technical Report TR-00-016*, Department of Computer Science, University of Minnesota, Minneapolis.

Kira, K., and Rendell, L. A. (1992): The feature selection problem: Traditional methods and a new algorithm. *Proceedings of the Tenth National*

*Conference on Artificial Intelligence MIT Press*. pp. 129-134.

Kohavi, R. and John G. (1997): Wrappers for Feature Subset Selection. *Artificial Intelligence*, Vol. 97, No. 1-2, pp. 273-324

Kohavi, R. and Sommerfield, D. (1995): Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. *Proc. of the First Int'l Conference on Knowledge Discovery and Data Mining*, pages 192–197, Montreal, Quebec.

Koller,D. and Sahami,M. (1997): Hierarchically classifying documents using very few words. *Proc. of The Fourteenth International Conference on Machine Learning (ICML'97)*, pages 170-178,.

Langley, P. and Sage, S. (1994): Induction of selective bayesian classifiers. *Proc. of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 399–406, Seattle, WA.

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

Liu, H. and Setiono, R. (1998a): Incremental Feature Selection. Applied Intelligence, 9:217-230.

Liu, H. and Motoda, H. (1998b): *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers.

McCallum, A. K. (1996): Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. Online: http://www-2.cs.cmu.edu/~mccallum/bow/.

Mitchell, T. (1996): *Machine Learning*. McCraw Hill.

Moore, J., Han, E., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V. and Mobasher, B. (1997): Web page categorization and feature selection using association rule and principal component clustering, *In 7th Workshop on Information Technologies and Systems.*

Papadimitriou, C., Raghavan, P., Tamaki, H. and Vempala, S. (1998): Latent sematic indexing: A probabilistic analysis. *Proc. of Symposium on Principles of Database Systems*.

PUBMED, http://www.ncbi.nlm.nih.gov/pubmed/

Salton, G. (1989): *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley.

Shankar, S. and Karypis, G. (2000): Weight adjustment schemes for a centroid based classifier. *Computer Science Technical Report TR00-035*, Department of Computer Science, University of Minnesota, Minneapolis, Minnesota.

UMLS System, http://www.nlm.nih.gov/research/umls/

Wang, B. B., McKay, R I. (Bob), Abbass, H. A. and Barlow, M. (2002): Learning Text Classifier using the Domain Concept Hierarchy. *Proc. of 2002 Inernational Conference on Communication Circuits and Systems and West Sino Expositions (IEEE Press)*, Chengdu, China.

Wettschereck, D., Aha, D.W., and Mohri T. (1997): A review and empirical evaluation of feature-weighting methods for a class of lazy learning algorithms. *AI Review*, 11, 1997.

Wierner, E., Pedersen, J.O. and Weigend, A. S. (1995): A neural network approach to topic spotting. *Proc. of the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95)*.

Wilbur, J. W. and Sirotkin, K. (1992): The automatic identification of stop words. *Journal of Information Science*, 18:45-55.

Yang, Y. (1995): Noise reduction in a statistical approach to text categorization. *Proc. of the Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95)*, pages 256-263.

Yang, Y. (1999): An Evaluation of Statistical Approaches to Text Categorization" Journal of Information Retrieval, Vol 1, No. 1/2, pp 67-88.

Yang, Y. and Pedersen, J. O. (1997): A Comparative Study on Feature Selection in Text Categorization. In *ICML 97*. pp. 412-420.