1998 International Computer Symposium
Workshop on Artificial Intelligence
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

# SIMPLE FEATURE SELECTION METHODS THAT MAKE A DIFFERENCE: AN APPLICATION TO TEXT CATEGORIZATION

*Ayşe Pınar Saygın*

Department of Computer Engineering and Information Science,
Bilkent University, Ankara, Turkey
Email: psaygin@cs.bilkent.edu.tr

## ABSTRACT

Feature selection is an important issue in machine learning. Especially in domains that inherently possess a large number of features, feature selection is a good resort to reduce computational costs. Moreover, gains in accuracy are also expected after feature selection since irrelevant features can act as noise. Text categorization is an area that has been getting a lot of attention lately an applying feature selection to this domain can be highly beneficial. We present a very simple approach to feature selection for text categorization and demonstrate that favorable and significant results can be obtained using commonsense, rule-of-thumb methods.

## 1. INTRODUCTION

Feature selection is one of the central problems in machine learning. Attributes that are irrelevant to the target concept are not only 'excess baggage', but can also act as noise. If we reduce the set of features to be used by the induction algorithm, we can not only decrease computational costs considerably, but also improve the accuracy of the resulting model. It is desired that the induction algorithms base their decisions on the features that are relevant to the target concept.

For domains that are characterized by a large number of features, the selection of the attributes to use in predictions becomes even more significant since it becomes more likely that some of these attributes are irrelevant. In many practical machine learning applications, we are faced with domains that contain irrelevant features that are not known *a priori*. A similar situation occurs when the data is too large or complex to be processed by humans and decisions about relevant and irrelevant features must be trusted to the induction algorithm. It may also be the case that the same body of data is used to learn different concepts: Features that are relevant to one concept may be totally irrelevant to another. For these, and similar reasons feature selection is an increasingly prominent issue in machine learning.

A recent and challenging area that attracts the attention of many computer scientists is text categorization. This is a topic of augmenting importance and automating it can be beneficial for many theoretical studies and industrial applications. A natural idea has been that of applying machine learning techniques to this domain. However, the domain poses many problems that require special attention. For one thing, unlike the data that we may be faced with in other domains, text data contain features that are highly context-sensitive. The features[1] may lose or change meaning when considered independently of each other. Most applications of machine learning to text categorization ignore this issue, preferring probabilistic methods that regard each feature as an independent entity, even though there are some approaches to text categorization [Cohen & Singer, 1996] that try to handle context-sensitivity. In addition to context, handling overlapping concepts and irrelevant attributes are also advantageous for obtaining reliable results.

On the other hand, for this domain, the amount of data that needs to be processed is very large. Employing complex techniques turns out to be impractical, to say the least. This is one of the reasons why much of the research done on text categorization centers on probabilistic approaches. We admit that incorporating sophisticated NLP and AI techniques in machine learning applications to obtain more accurate results in text categorization and similar tasks sounds very promising. However, the degrees of sophistication aimed for need not be that high at this point. We should focus on developing *practical* applications that are at least scaleable to, and preferably efficient on text categorization tasks.

A prevalent characteristic of the text categorization domain is the large number of features involved. We are naturally lead to the idea of applying feature selection to determine relevant features for use in the categorization of documents. This paper argues that the text categorization domain can benefit from feature selection to a great extent. In particular we will try to demonstrate that basic, commonsense feature selection methods are easy to apply, feasible and can be noticeably useful in real life text categorization tasks.

The rest of this paper is organized as follows. In Section 2, we will briefly describe the related work in machine learning. Section 3 serves the same purpose and describes some issues and work from the text categoriza-

[1] We are using the term feature to mean individual words, although this is not the only alternative. Surprisingly, using phrases or clusters of words/phrases are shown to be less effective representations of text than single words. See [Lewis, 1992].

tion area. Both of these sections refer the interested reader to appropriate references. Section 4 describes the experiments we have carried out in substantial detail. The results are analyzed and future directions are listed in Section 5.

## 2. RELATED WORK IN FEATURE SELECTION

Large numbers of features often adversely affect the performance of induction algorithms. Moreover reducing the feature set size also reduces computational costs. We would like induction algorithms to scale well to domains with many irrelevant features. However, the most prevalent machine learning algorithms perform poorly in this respect, suggesting that the irrelevant features are detected and reported to them beforehand. Reasons of this sort have encouraged the machine learning community to explore ways of selecting the relevant features. As defined in [John *et. al.*, 1994], most of this work is divided along two lines: Filter and wrapper models.

The filter model performs feature selection as a preprocessing step to induction. The whole feature set is reduced in size, and *then* passed to the induction algorithm. Thus, the bias of the induction algorithm and the bias of the feature selection algorithm remain independent from each other. Filter methods are easy to apply and efficient compared to the wrapper methods. The wrapper model is more sophisticated. This model searches through the feature subset space using the estimated accuracy from the induction algorithm to be used as the measure of 'goodness'. The bias of the induction algorithm is exploited during the feature selection process. The wrapper model provides a preferable approach compared to filter methods. It has one disadvantage, however. While being theoretically powerful, wrapper methods are usually very expensive to run and can be impossible to employ in the presence of a very large number of features.

A number of researchers have recently addressed the issue of feature selection in machine learning. John, Kohavi and Pfleger [John *et. al.*, 1994] provide formal definitions of feature relevance for machine learning. Earlier approaches to feature selection were all filtering methods. Of these, the most well known methods are FOCUS [Almuallim & Dietterich, 1991] and RELIEF [Kira & Rendell, 1992].

More recently, work has concentrated around the wrapper method. This method was first introduced by John, Kohavi and Pfleger [John *et. al.*, 1994]. Although some methods to reduce the time spent by the feature selection algorithm are being developed [Caruana & Freitag, 1994] the high computational cost of the wrapper methods remain their biggest disadvantage. This cost is inherently difficult to lower because the induction algorithm is called each time a feature subset is considered.

For more information on the topic the reader is referred to [Almuallim & Dietterich, 1991; Kira & Rendell, 1992; Koller & Sahami, 1996; Langley, 1994; John *et. al.*, 1994; Liu & Setiono, 1996; Domingos, 1997].

## 3. RELATED WORK IN TEXT CATEGORIZATION

There has been much interest towards text categorization from various areas of computer science in the recent years. As the amount of online information, of which a large part is textual, continues to increase, the demand for text categorization is also bound to augment.

Machine learning techniques are used in text processing tasks such as keyword extraction, document filtering, routing, and summarization. Information retrieval is one of the disciplines that borrow from machine learning [Chen, 1995].

Various different machine learning algorithms have been used in text categorization such as rule learners [Cohen, 1995], nearest neighbor classification [Yang, 1994], multiplicative and additive weighting algorithms [Dagan, *et.al.*, 1997; Lewis, *et. al.*, 1996], statistical classifiers [Lewis & Ringuette, 1994; Lewis & Gale, 1994; Yang, 1997; Mladenic, 1998] and decision trees [Lewis & Ringuette, 1994]. Statistical classifiers and nearest neighbor methods are preferred because they are least affected by the large sizes of the training data.

As was mentioned before, the characteristics of the text categorization domain, such as the presence of irrelevant features, imply that it would be advantageous to apply feature selection to this domain. In fact, simple stop word elimination can be thought of as a filter method for feature selection since it removes certain words from the data before it is processed. Removing words that occur very few or very many times in the data set [Lewis, 1992; Yang & Pedersen,1997] is a very simple, but successful, approach. Other approaches have been using certain scoring methods that are borrowed from statistics, information theory and information retrieval [Yang & Pedersen, 1997; Mladenic, 1998]. These approaches are simple, yet reliable. They order the features according to some measure of relevance and select those that remain above a certain threshold. This is the approach that we are also using in the current study.

Wrapper methods, especially context sensitive feature selection [Domingos, 1997], are very fetching. However, they are not directly applicable to the text categorization domain because of two reasons: The first one has already been mentioned above. Wrapper methods are not feasible when the number of features is high. In the text categorization domain the features are words and many real life applications are bound to contain hundreds, even thousands of features. The second reason is a consequence of this. Wrapper methods that have been proposed usually add or remove a single feature by testing the outcome of this action on the accuracy of the induction algorithm. However when the feature size is large and there are many features that do not contribute much to the classification, the likelihood of observing a difference in accuracy upon adding or removing one feature from the set of those that are to be considered is very low. Combining this fact with the incredibly high cost of applying cross validation, it becomes apparent that wrapper methods are not suited for use in this domain. An alternative hybrid filter-wrapper method can be employed, which will be described briefly in Section 5.

## 4. EXPERIMENT

We have conducted an experiment to see how feature selection effects the classification accuracy of the *k*-nearest-neighbor classifier. Simple feature selection approaches were employed: Stop-word elimination, frequency-based elimination of scarce words, and a statistical measure called *distinctiveness* were used. The text categorization task was to classify newsgroup articles into appropriate categories, i.e. newsgroups.

### 4.1 The Data Set

For the experiments, we have used a data set of 2000 newsgroup articles from 20 different newsgroups. Table 1 lists the newsgroups that these articles were taken from. Before processing, all words were converted to lowercase. Punctuation, numbers, message headers, and their contents (except for the contents of the 'subject' header) were removed. A stop-word list of about 600 words (which mainly consisted of prepositions and common verbs) was used to pre-process the documents.

**Newsgroups for the articles**

alt.atheism
comp.graphics
comp.os.ms-windows.misc
comp.sys.ibm.pc.hardware
comp.sys.mac.hardware
comp.windows.x
misc.forsale
rec.autos
rec.motorcycles
rec.sport.baseball
rec.sport.hockey
sci.crypt
sci.electronics
sci.med
sci.space
soc.religion.christian
talk.politics.guns
talk.politics.mideast
talk.politics.misc
talk.religion.misc

**Table 1.** Newsgroups that the data set was taken from

Also to prevent meaningless strings to enter the data set, words of length greater than 16, words containing '@' and the substring '//:' were removed.[2] It may be argued that this is 'unfair' since we have used knowledge about the domain, i.e. that the strings with the character '@' and the substring '//:' in them should be ignored. However, we strongly believe that such knowledge should be exploited, when available. If what we are looking for are deployable text categorization applications, we need to restrict ourselves to certain domains, at least for the time being. Almost every document in the data set contained e-mail addresses, host names, paths and URLs and we have seen that removing words by the rules-of-thumb described above left us with cleaner, yet still natural, pieces of text.

Much of the research in machine learning for text categorization uses 'neat' data, such as news articles or artificial data sets. However, in real life, texts are much less organized and error-free. If one is concerned with developing functional text categorization applications, this must be taken into consideration. Results obtained on 'neat' data may not always apply to 'messy' data. The articles in the data set we used were real-life newsgroup articles from various groups and thus contained many problems that are characteristic of the writings of 'the-guy-next-door'. Spelling mistakes, words that do not mean what they were intended to mean, jumps from subject to subject were frequent.

In the experiment, the classification task was assigning each article to the newsgroup to which it belonged. None of the articles in the data set belonged to more than one category. This is not necessarily the case for all text categorization tasks. Moreover, for each message, the class that an article belongs to was automatically determined by the newsgroup it was taken from. The possibility of an article submitted to a group being irrelevant to the topic was ignored.

For evaluation, the accuracy measure, i.e. the ratio of the correctly classified documents over the total number of classifications, was used. This accuracy was computed by 5-fold cross-validation.

### 4. 2 The Learning Algorithm

The induction algorithm used was the *k*-Nearest-Neighbor algorithm (KNN). This algorithm was chosen because it suffers from irrelevant attributes largely and because it scales to the large number of features involved in the task. Other algorithms and learning methods can also be used to further validate our results on the usefulness of feature selection on text categorization tasks.

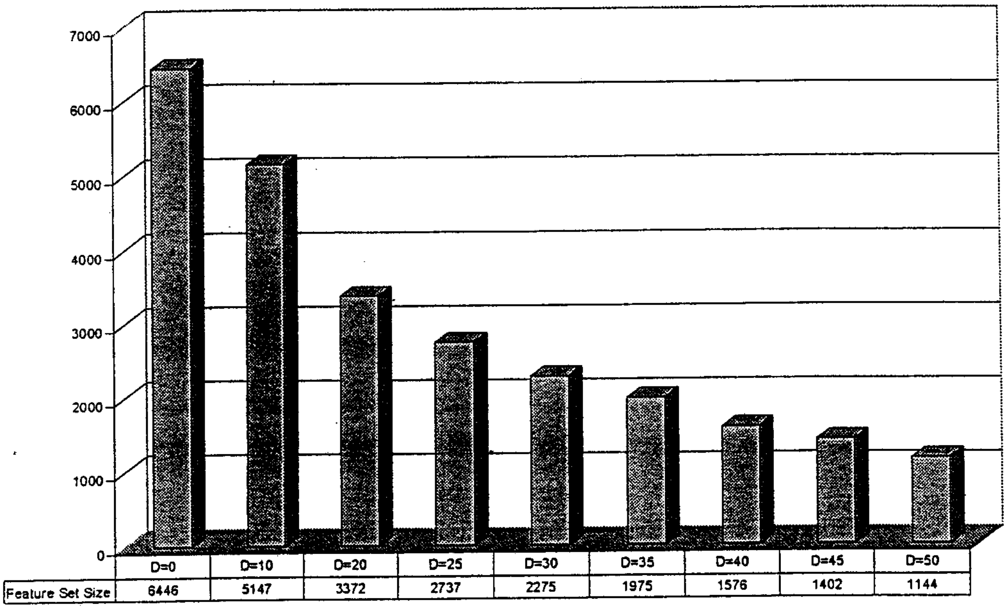
### 4.3 Feature Selection

The feature selection task was done in two steps. First, the words that occurred less than five times in the whole data set and those that occurred in a single document were removed. This roughly corresponds to term selection based on document frequency (DF) as used in [Yang & Pedersen, 1997]. This process reduced the vocabulary size immensely and inspection shows that the majority of the removed words were spelling mistakes; many were

[2] To prevent e-mail addresses and URL's from being used as features.

names, streets and similar highly specific and irrelevant features; only a few were actual, meaningful words.

We resorted to filtering out these words because we believed that eliminating them would make the data smaller and more noise-free while keeping the accuracy levels high. Both inspection and experimental results support this claim. As was mentioned before the data set we used was quite 'messy' and by this simple filtering approach, most spelling and typing mistakes were handled. Also removing words that occur only in a single document seemed to filter out names of people or very specific words related to spurious concepts that became the discussion topic only transiently (for instance, when one is replying to a thread or quoting other people, a name or a word can occur numerous times in a single document, but may not be relevant throughout the instance space) and repeated misspellings of the same words.[3]



| | D=0 | D=10 | D=20 | D=25 | D=30 | D=35 | D=40 | D=45 | D=50 |
|---|---|---|---|---|---|---|---|---|---|
| Feature Set Size | 6446 | 5147 | 3372 | 2737 | 2275 | 1975 | 1576 | 1402 | 1144 |

**Table 1.** The variation of the feature set size with respect to *D*.

The remaining words were further filtered using a statistical weighting approach, which we will call the *distinctiveness* measure. The bias of our feature selection was as follows: The features that are uniformly distributed among the categories are not likely to contribute to the classification, i.e. are not distinctive. More specifically, in this case, the words that have occurred uniformly in documents from all newsgroups are not considered to be typical representatives of a particular category and hence, will be omitted.

$$D = \frac{\sum_{i=1}^{k}(X_i - X)^2}{(k-1)X^2 + (N-X)^2} \qquad (1)$$

To measure how 'good' a feature is, we have tried to see how close it is to being 'ideal'. A commonsense notion was used: The ideal situation is the case when a word is present *only* in the documents from a certain group. That is, the word is characteristic of the corresponding class and its presence in test instances will strongly imply that the instance belongs to the class in question. These are the kinds of words that we are interested in keeping. On the other hand, if a word displays a uniform distribution across categories, removing it from consideration is not likely to have a negative effect on classification accuracy while significant reductions in feature set size can lead to faster programs and less memory consumption and thereby be useful.

The standard deviation (or its square, variance) is the commonly used statistical measure to detect how 'uniform' a distribution is. Our interests are, as it happens, non-uniform distributions so we are after words with frequencies that display a high standard deviation across the categories. However, the variance tends to be higher for words that occur numerous times than those that display a more desirable distribution but occur fewer times. Normalization is thus needed and we chose a very sim-

[3] This situation is surprisingly common. Such words will go undetected only if several people spell the word wrong or if one author has posted multiple messages containing the misspelled word. In these cases it may be argued that even the wrong version of the word is a relevant feature. One such word that went undetected was 'suecide'.

ple method. For each word, we consider the ratio of the actual variance to the most desirable distribution's variance. The simplified general formula for distinctiveness, $D$, is given in (1). $X_i$ denotes the number of occurences of the word in the $i^{th}$ category, $N$ is the total number of times the word occurred in the whole data set, $k$ is the number of classes (in our case, newsgroups) and $X$ is the mean of the frequencies, i.e $N$ / $k$. After constants are canceled out, the denominator corresponds to the variance of the ideal case (where all $N$ occurrences are in a single class) and the nominator corresponds to the actual variance. For simplicity, we multiplied this value by 100 so as to obtain a percentile result. Figure 1 shows how the total number of words decreases with this percentage. In the experiments, we have used different threshold values for this percentage to select the features for use in classification. Words with scores above the threshold value were retained and those with scores below it were discarded from the data set.

Notice that in Figure 1, the value corresponding to $D$=0 is 6446. The initial size of the vocabulary was, however, 29570. This means that the DF-like filtering method removed 23124 words, which constitutes approximately 78% of the initial vocabulary. Moreover, after this reduction, the accuracy of the nearest neighbor classifier increases. This suggests that it should be possible to remove more words from the vocabulary without sacrificing from the classification accuracy [Yang & Pedersen, 1997].

### 4.4 Results

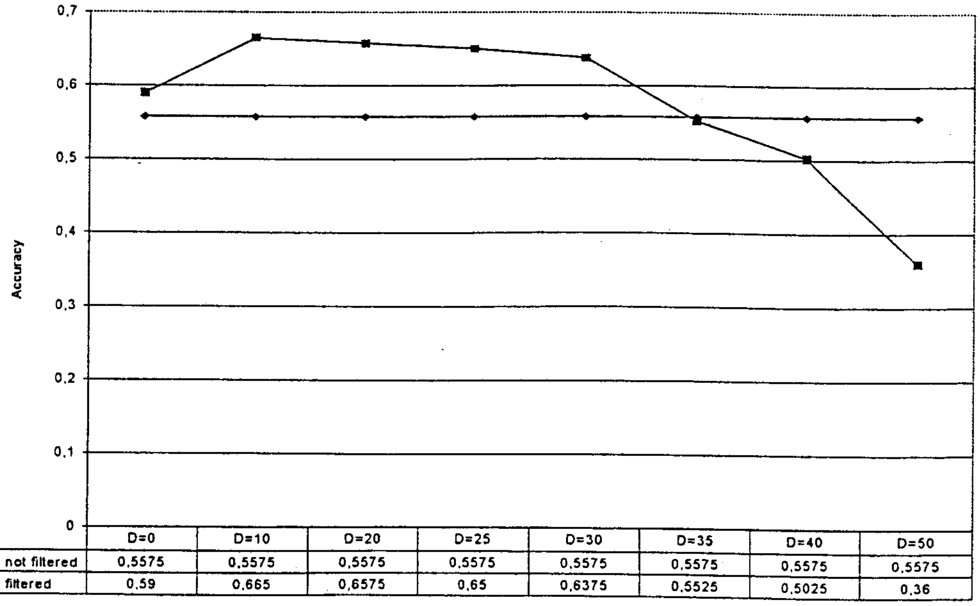
The aim of this study was to show that a small subset of the vocabulary (as opposed to the whole set of words in the data set) could be sufficient for performing text categorization successfully. The experiment results were favorable since they indicate that the feature subset size can be decreased dramatically without experiencing any loss in classification accuracy. Moreover, increases in accuracy were observed for a wide range of threshold values for $D$.

Reducing the vocabulary size will result in significant gains in time and memory consumption. In practical applications, it is desirable to have small amounts of data to process. However, care must be taken so that valuable information is not lost. Often, it is not possible to keep reducing the feature set size without losing some information. The experiment results indicate that the text categorization domain is not faced with this tradeoff point early on in the reduction process. This is due to the large number of irrelevant features present in this domain.

As can be seen in Figure 1, the vocabulary size decreases significantly up until the threshold for $D$ is 30%. After this point, the feature set size continues to decrease, albeit at a slower rate. It must be noted that the substantial decrease in the feature set size occurs after the first filtering step. As was mentioned above, the number of words removed in that step was 23124, which is more than 78% of the total number of words. However, considering the second step performed over the result of the first one, and assuming a threshold of 30% is chosen there is a further 65% reduction in the feature set size. Considering the overall effect, it can be seen that the feature set size was reduced by 93% and appreciable increases in accuracy can be observed.

The effects of feature selection on classification accuracy are shown in Figure 2. KNN increases in accuracy after feature selection for all $k$ values. Besides reducing the computational cost, feature selection was also seen to improve accuracy significantly for all values of $k$ tested whenever the threshold value for $D$ was chosen between 10% and 30%. The gain in accuracy begins to diminish after 30% threshold is passed and a loss is seen after the threshold is increased above 35%. This is because more documents start becoming "empty" after selecting fewer features for use.

An interesting point worth noting is that the significant increase in accuracy can be observed only after filtering



| | D=0 | D=10 | D=20 | D=25 | D=30 | D=35 | D=40 | D=50 |
|---|---|---|---|---|---|---|---|---|
| ◆ KNN, k=10, not filtered | 0,5575 | 0,5575 | 0,5575 | 0,5575 | 0,5575 | 0,5575 | 0,5575 | 0,5575 |
| ■ KNN, k=10, filtered | 0,59 | 0,665 | 0,6575 | 0,65 | 0,6375 | 0,5525 | 0,5025 | 0,36 |

**Figure 2.** KNN on the filtered data compared to KNN on the original data using accuracy as a measure for $k$=10. The graph is given with the percentile threshold for $D$ on the $x$-axis and the accuracy on the $y$-axis.

out the words according to the *distinctiveness* measure. The reason behind this could be that it tries to pick out the meaningful words from text in addition to reducing the feature set size.

The reduction in the size of the feature subset is not so significant after increasing the thresholds above 35% so it seems likely that results that attain a 'nice' balance between gains in computational cost and accuracy can be obtained by selecting a threshold value around 25% for this data set.

## 5. CONCLUSIONS AND FUTURE WORK

This paper dealt with the problem of feature selection in text categorization and attempted to show that simple 'rule-of-thumb' techniques can be of considerable use in this task. In Sections 2 and 3 approaches to feature selection, text categorization and the associated problems are discussed so as to give the reader a general picture. The experimental studies were reported in detail in Section 4. A new and simple feature selection measure, called was introduced.

It was observed that those feature selection methods applied here result in notable reductions in the vocabulary size. Moreover, it was seen that choosing appropriate threshold points for *D*, we could actually increase the classification accuracy of the KNN classifier, usually by more than 10%. The decreases in run time were prominent; while the run time for the unprocessed data was around 6 hours, processed data ran in 5-15 minutes.

A possible future work could be to use the *distinctiveness* measure as feature weights rather than doing selection based on a threshold. In this case, naturally the vocabulary size remains the same and gains in computational cost are not to be expected. However, this can be carried out in order to see whether additional increases in accuracy can be observed.

As can be seen, the success of the feature selection is highly dependent on the threshold value used. The optimal value for the threshold can vary depending on the data and the induction algorithm. Rather than trying to come up with a value to use on every domain and with every learning algorithm, we could apply a wrapper strategy to learn a threshold such that the value used is optimal for that particular data set and induction algorithm. While wrapper methods are not feasible for use in text categorization tasks, the sophistication that they provide can be incorporated in the classification task in this manner.

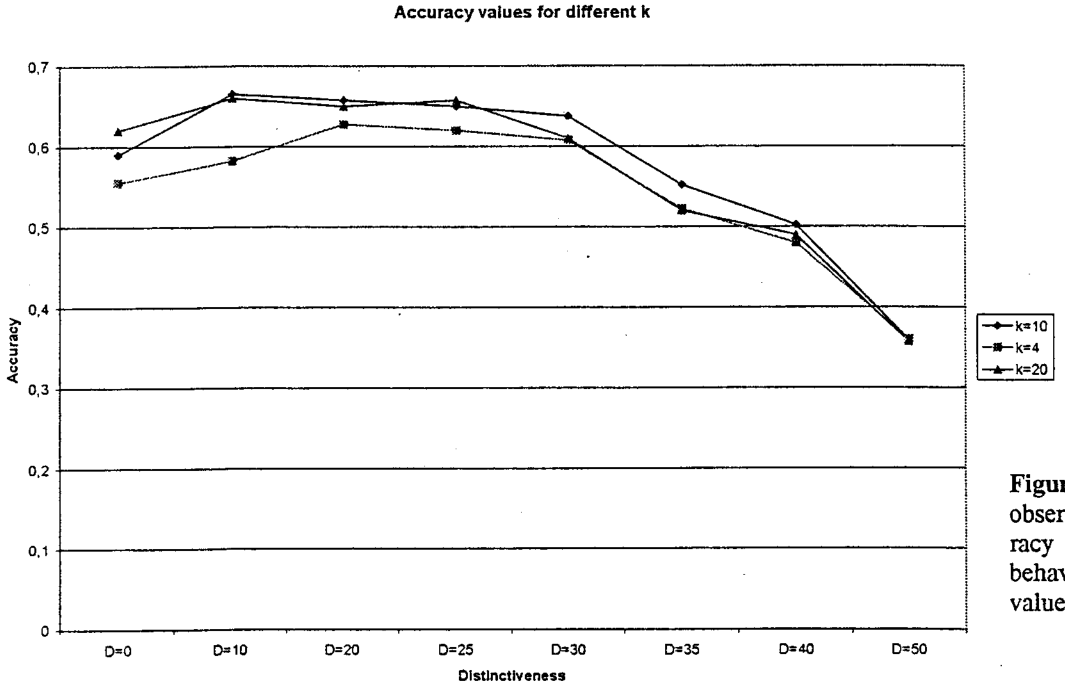


**Figure 3.** It has been observed that the accuracy shows a similar behavior for different values of *k*.

**Acknowledgments**

I would like to express my gratitude for Dr. Halil Altay Güvenir, without whom this work would not have been possible, for his kind help and suggestions. I want to thank my colleague Tuba Yavuz for her help, patience and support while carrying out the experiments.

## References

Almuallim, H., & Dietterich, T.G. (1991). Learning with many irrelevant features. *Proceedings of the Ninth National Conference on Artificial Intelligence* (pp. 547-552). San Jose, CA: AAAI Press.

Almuallim, H., & Dietterich, T.G. (1992). Efficient algorithms for identifying relevant features. *Proceedings of the Ninth Canadian Conference on Artificial Intelligence* (pp. 38-45). Vancouver, BC: Morgan Kaufmann.

Caruana, R.A., & Freitag, D. (1994). Greedy attribute selection. *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 25-32). New Brunswick, NJ: Morgan Kaufmann.

Chen, H. (1995). Machine Learning for information retrieval: Neural networks, symbolic learning and genetic algorithms. *JASIS 46(3),* 194-216.

Cohen, W. (1995). Text Categorization and Relational Learning. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 124-132) Lake Tahoe, CA: Morgan Kaufmann.

Cohen, W. & Singer, Y. (1996). Context-sensitive learning methods for text categorization. *Proceedings SIGIR '96.*

Dagan, I., Karov, Y., and Roth, D. (1997). Mistake-driven learning in text categorization. *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing.*

Devijver, P.A. & Kittler, J. (1982) *Pattern Recognition: A Statistical Approach.* Englewood Cliffs, NJ: Prentice/Hall.

Domingos, P. (1997). Context-sensitive feature selection for lazy learners. *Artificial Intelligence Review, 11,* 227-253.

John, G.H., & Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 121-129). New Brunswick, NJ: Morgan Kaufmann.

Kira, K. & Rendell, L. (1992). A practical approach to feature selection. *Proceedings of the Ninth International Conference on Machine Learning* (pp. 249-256) Aberdeen, Scotland: Morgan Kaufmann.

Kohavi, R., & John, G. (1995). Wrappers for feature subset selection. Technical Report, Computer Science Department, Stanford University.

Koller, D. & Sahami, M. (1996). Toward optimal feature selection. *ICML-96: Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 284-292). Bari, Italy: Morgan Kaufmann.

Lang, K. (1995) NewsWeeder: Learning to filter NetNews. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 331-339) Lake Tahoe, CA, Morgan Kaufmann.

Langley, P. (1994). Selection of irrelevant features in machine learning. *Proceedings of the AAAI Fall Symposium on Relevance.* New Orlans, LA: AAAI Press.

Lewis, D.D., (1992). Feature selection and feature extraction for text categorization. *Proceedings of Speech and Natural Language Workshop* (212-217). San Mateo, CA. Defense Advanced Research Projects Agency: Morgan Kauffmann.

Lewis, D.D. & Gale, W.A. (1994) A sequential algorithm for training text classifiers. *Proceedings SIGIR '94.* (pp. 3-12) Dublin, Ireland: Springer-Verlag.

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

Lewis, D.D. & Schapire, R.E. & Callan, J.P. & Papka, R. (1996). Training algorithms for linear text classifiers. *Proceedings SIGIR '96.*

Mladenic, D. (1998). Feature subset selection in text-learning. *10th European Conference on Machine Learning (ECML-98).*

Yang, Y. (1994) Expert Network: Effective and efficient learning from human decisions in text categorization and retrieval. *Proceedings of SIGIR'94.* (pp. 13-22) Dublin, Ireland: Springer-Verlag.

Yang, Y., Pedersen J.P. (1997) A Comparative Study on Feature Selection in Text Categorization, *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97).*

Yang, Y. (1997) An Evaluation of statistical approaches to text categorization. Technical Report CMU-CS-97-127, Computer Science Department, Carnegie Mellon University.