A Comparative Study on Feature Selection in Text Categorization

Yiming Yang
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3702, USA
yiming@cs.cmu.edu

Jan O. Pedersen
Verity, Inc.
894 Ross Dr.
Sunnyvale, CA 94089, USA
jpederse@verity.com

Abstract

This paper is a comparative study of feature selection methods in statistical learning of text categorization. The focus is on aggressive dimensionality reduction. Five methods were evaluated, including term selection based on document frequency (DF), information gain (IG), mutual information (MI), a χ^2-test (CHI), and term strength (TS). We found IG and CHI most effective in our experiments. Using IG thresholding with a k-nearest neighbor classifier on the Reuters corpus, removal of up to 98% removal of unique terms actually yielded an improved classification accuracy (measured by average precision). DF thresholding performed similarly. Indeed we found strong correlations between the DF, IG and CHI values of a term. This suggests that DF thresholding, the simplest method with the lowest cost in computation, can be reliably used instead of IG or CHI when the computation of these measures are too expensive. TS compares favorably with the other methods with up to 50% vocabulary reduction but is not competitive at higher vocabulary reduction levels. In contrast, MI had relatively poor performance due to its bias towards favoring rare terms, and its sensitivity to probability estimation errors.

1 Introduction

Text categorization is the problem of automatically assigning predefined categories to free text documents. While more and more textual information is available online, effective retrieval is difficult without good indexing and summarization of document content. Documents categorization is one solution to this problem. A growing number of statistical classification methods and machine learning techniques have been applied to text categorization in recent years, including multivariate regression models[8, 27], nearest neighbor classification[4, 23], Bayes probabilistic approaches[20, 13], decision trees[13], neural networks[21], symbolic rule learning[1, 16, 3] and inductive learning algorithms[3, 12].

A major characteristic, or difficulty, of text categorization problems is the high dimensionality of the feature space. The native feature space consists of the unique terms (words or phrases) that occur in documents, which can be tens or hundreds of thousands of terms for even a moderate-sized text collection. This is prohibitively high for many learning algorithms. Few neural networks, for example, can handle such a large number of input nodes. Bayes belief models, as another example, will be computationally intractable unless an independence assumption (often not true) among features is imposed. It is highly desirable to reduce the native space without sacrificing categorization accuracy. It is also desirable to achieve such a goal automatically, i.e., no manual definition or construction of features is required.

Automatic feature selection methods include the removal of non-informative terms according to corpus statistics, and the construction of new features which combine lower level features (i.e., terms) into higher-level orthogonal dimensions. Lewis &Ringuette[13] used an information gain measure to aggressively reduce the document vocabulary in a naive Bayes model and a decision-tree approach to binary classification. Wiener et al.[21, 19] used mutual information and a χ^2 statistic to select features for input to neural networks. Yang [24] and Schutze et al. [19, 21, 19] used principal component analysis to find orthogonal dimensions in the vector space of documents. Yang & Wilbur [28] used document clustering techniques to estimate probabilistic "term strength", and used it to reduce the variables in linear regression and nearest neighbor classification. Moulinier et al. [16] used an inductive learning algorithm to obtain features in disjunc-

tive normal form for news story categorization. Lang [11] used a minimum description length principle to select terms for Netnews categorization.

While many feature selection techniques have been tried, thorough evaluations are rarely carried out for large text categorization problems. This is due in part to the fact that many learning algorithms do not scale to a high-dimensional feature space. That is, if a classifier can only be tested on a small subset of the native space, one cannot use it to evaluate the full range of potential of feature selection methods. A recent theoretical comparison, for example, was based on the performance of decision tree algorithms in solving problems with 6 to 180 features in the native space[10]. An analysis on this scale is distant from the realities of text categorization.

The focus in this paper is the evaluation and comparison of feature selection methods in the reduction of a high dimensional feature space in text categorization problems. We use two classifiers which have already scaled to a target space with thousands or tens of thousands of categories. We seek answers to the following questions with empirical evidence:

- What are the strengths and weaknesses of existing feature selection methods applied to text categorization?
- To what extend can feature selection improve the accuracy of a classifier? How much of the document vocabulary can be reduced without losing useful information in category prediction?

Section 2 describes the term selection methods. Due to space limitations, we will not include phrase selection (e.g.[3]) and approaches based on principal component analysis[5, 24, 21, 19]. Section 3 describes the classifiers and the document corpus chosen for empirical validation. Section 4 presents the experiments and the results. Section 5 discusses the major findings. Section 6 summarizes the conclusions.

2 Feature Selection Methods

Five methods are included in this study, each of which uses a term-goodness criterion thresholded to achieve a desired degree of term elimination from the full vocabulary of a document corpus. These criteria are: document frequency (DF), information gain (IG), mutual information (MI), a χ^2 statistic (CHI), and term strength (TS).

2.1 Document frequency thresholding (DF)

Document frequency is the number of documents in which a term occurs. We computed the document frequency for each unique term in the training corpus and removed from the feature space those terms whose document frequency was less than some predetermined threshold. The basic assumption is that rare terms are either non-informative for category prediction, or not influential in global performance. In either case removal of rare terms reduces the dimensionality of the feature space. Improvement in categorization accuracy is also possible if rare terms happen to be noise terms.

DF thresholding is the simplest technique for vocabulary reduction. It easily scales to very large corpora, with a computational complexity approximately linear in the number of training documents. However, it is usually considered an ad hoc approach to improve efficiency, not a principled criterion for selecting predictive features. Also, DF is typically not used for aggressive term removal because of a widely received assumption in information retrieval. That is, low-DF terms are assumed to be relatively informative and therefore should not be removed aggressively. We will re-examine this assumption with respect to text categorization tasks.

2.2 Information gain (IG)

Information gain is frequently employed as a term-goodness criterion in the field of machine learning[17, 14]. It measures the number of bits of information obtained for category prediction by knowing the presence or absence of a term in a document. Let $\{c_i\}_{i=1}^m$ denote the set of categories in the target space. The information gain of term t is defined to be:

$$\begin{aligned} G(t) = & -\sum_{i=1}^{m} P_r(c_i) \log P_r(c_i) \\ & + P_r(t) \sum_{i=1}^{m} P_r(c_i|t) \log P_r(c_i|t) \\ & + P_r(\bar{t}) \sum_{i=1}^{m} P_r(c_i|\bar{t}) \log P_r(c_i|\bar{t}) \end{aligned}$$

This definition is more general than the one employed in binary classification models[13, 16]. We use the more general form because text categorization problems usually have a m-ary category space (where m may be up to tens of thousands), and we need to measure the goodness of a term globally with respect to all categories on average.

Given a training corpus, for each unique term we computed the information gain, and removed from the feature space those terms whose information gain was less than some predetermined threshold. The computation includes the estimation of the conditional probabilities of a category given a term, and the entropy computations in the definition. The probability estimation has a time complexity of $O(N)$ and a space complexity of $O(VN)$ where N is the number of training documents, and V is the vocabulary size. The entropy computations has a time complexity of $O(Vm)$.

2.3 Mutual information (MI)

Mutual information is a criterion commonly used in statistical language modelling of word associations and related applications [7, 2, 21]. If one considers the two-way contingency table of a term t and a category c, where A is the number of times t and c co-occur, B is the number of time the t occurs without c, C is number of times c occurs without t, and N is the total number of documents, then the mutual information criterion between t and c is defined to be

$$I(t,c) = \log \frac{P_r(t \wedge c)}{P_r(t) \times P_r(c)}$$

and is estimated using

$$I(t,c) \approx \log \frac{A \times N}{(A+C) \times (A+B)}$$

$I(t,c)$ has a natural value of zero if t and c are independent. To measure the goodness of a term in a global feature selection, we combine the category-specific scores of a term into two alternate ways:

$$I_{avg}(t) = \sum_{i=1}^{m} P_r(c_i) I(t,c_i)$$

$$I_{max}(t) = \max_{i=1}^{m} \{I(t,c_i)\}$$

The MI computation has a time complexity of $O(Vm)$, similar to the IG computation.

A weakness of mutual information is that the score is strongly influenced by the marginal probabilities of terms, as can be seen in this equivalent form:

$$I(t,c) = \log P_r(t|c) - \log P_r(t).$$

For terms with an equal conditional probability $P_r(t|c)$, rare terms will have a higher score than common terms. The scores, therefore, are not comparable across terms of widely differing frequency.

2.4 χ^2 statistic (CHI)

The χ^2 statistic measures the lack of independence between t and c and can be compared to the χ^2 distribution with one degree of freedom to judge extremeness. Using the two-way contingency table of a term t and a category c, where A is the number of times t and c co-occur, B is the number of time the t occurs without c, C is the number of times c occurs without t, D is the number of times neither c nor t occurs, and N is the total number of documents, the term-goodness measure is defined to be:

$$\chi^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)}.$$

The χ^2 statistic has a natural value of zero if t and c are independent. We computed for each category the χ^2 statistic between each unique term in a training corpus and that category, and then combined the category-specific scores of each term into two scores:

$$\chi^2_{avg}(t) = \sum_{i=1}^{m} P_r(c_i) \chi^2(t,c_i)$$

$$\chi^2_{max}(t) = \max_{i=1}^{m} \{\chi^2(t,c_i)\}$$

The computation of CHI scores has a quadratic complexity, similar to MI and IG.

A major difference between CHI and MI is that χ^2 is a normalized value, and hence χ^2 values are comparable across terms for the same category. However, this normalization breaks down (can no longer be accurately compared to the χ^2 distribution) if any cell in the contingency table is lightly populated, which is the case for low frequency terms. Hence, the χ^2 statistic is known not to be reliable for low-frequency terms[6].

2.5 Term strength (TS)

Term strength is originally proposed and evaluated by Wilbur and Sirotkin [22] for vocabulary reduction in text retrieval, and later applied by Yang and Wilbur to text categorization [24, 28]. This method estimates term importance based on how commonly a term is likely to appear in "closely-related" documents. It uses a training set of documents to derive document pairs whose similarity (measured using the cosine value of the two document vectors) is above a threshold. "Term Strength" then is computed based on the estimated conditional probability that a term occurs in the second half of a pair of related documents given that it occurs in the first half. Let x and y be an arbitrary pair of distinct but related documents, and t be a term, then the strength of the term is defined to be:

$$s(t) = P_r(t \in y | t \in x).$$

The term strength criterion is radically different from the ones mentioned earlier. It is based on document clustering, assuming that documents with many shared words are related, and that terms in the heavily overlapping area of related documents are relatively informative. This method is not task-specific, i.e., it does not use information about term-category associations. In this sense, it is similar to the DF criterion, but different from the IG, MI and the χ^2 statistic.

A parameter in the TS calculation is the threshold on document similarity values. That is, how close two documents must be to be considered a related pair. We use AREL, the average number of related documents per document in threshold tuning. That is, we

compute the similarity scores of all the documents in a training set, try different thresholds on the similarity values of document pairs, and then choose the threshold which results in a reasonable value of AREL. The value of AREL is chosen experimentally, according to how well it optimized the performance in the task. According to previous evaluations of retrieval and categorization on several document collections[22, 28], the AREL values between 10 to 20 yield satisfactory performance. The computation of TS is quadratic in the number of training documents.

3 Classifiers and Data

3.1 Classifiers

To assess the effectiveness of feature selection methods we used two different m-ary classifiers, a k-nearest-neighbor classifier (kNN)[23] and a regression method named the Linear Least Squares Fit mapping (LLSF)[27]. The input to both systems is a document which is represented as a sparse vector of word weights, and the output of these systems is a ranked list of categories with a confidence score for each category.

Category ranking in kNN is based on the categories assigned to the k nearest training documents to the input. The categories of these neighbors are weighted using the similarity of each neighbor to the input, where the similarity is measured by the cosine between the two document vectors. If one category belongs to multiple neighbors, then the sum of the similarity scores of these neighbors is the the weight of the category in the output. Category ranking in the LLSF method is based on a regression model using words in a document to predict weights of categories. The regression coefficients are determined by solving a least squares fit of the mapping from training documents to training categories.

Several properties of kNN and LLSF make them suitable for our experiments:

1) Both systems are top-performing, state-of-the-art classifiers. In a recent evaluation of classification methods[26] on the Reuters newswire collection (next section), the break-even point values were 85% for both kNN and LLSF, outperforming all the other systems evaluated on the same collection, including symbolic rule learning by RIPPER (80%)[3], SWAP-1 (79%)[1] and CHARADE (78%)[16], a decision approach using C4.5 (79%)[15], inductive learning by Sleeping Experts (76%)[3], and a typical information retrieval approach named Rocchio (75%)[3]. On another variation of the Reuters collection where the training set and the test set are partitioned differently, kNN has a break-even point of 82% which is the same as the result of neural networks[21], and LLSF has a break-even point of 81%.

2) Both systems scale to large classification problems. By "large" we mean that both the input and the output of a classifier can have thousands of dimensions or higher[25, 24]. We want to examine all the degrees of feature selection, from no reduction (except removing standard stop words) to extremely aggressive reduction, and observe the effects on the accuracy of a classifier over the entire target space. For this examination, we need a scalable system.

3) Both kNN and LLSF are a m-ary classifier providing a global ranking of categories given a document. This allows a straight-forward global evaluation of per document categorization performance, i.e., measuring the goodness of category ranking given a document, rather than per category performance as is standard when applying binary classifiers to the problem.

4) Both classifiers are *context sensitive* in the sense that no independence is assumed between either input variables (terms) or output variables (categories). LLSF, for example, optimizes the mapping from a document to categories, and hence does not treat words separately. Similarly, kNN treats a document as an single point in a vector space. The context sensitivity is in distinction to *context-free* methods based on explicit independence assumptions such as naive Bayes classifiers[13] and some other regression methods[8]). A context-sensitive classifier makes better use of the information provided by features than a context-free classifier do, thus enabling a better observation on feature selection.

5) The two classifiers differ statistically. LLSF is based on a linear parametric model; kNN is a non-parametric and non-linear classifier, that makes few assumptions about the input data. Hence a evaluation using both classifiers should reduce the possibility of classifier bias in the results.

3.2 Data collections

We use two corpora for this study: the Reuters-22173 collection and the OHSUMED collection.

The Reuters news story collection is commonly used corpora in text categorization research [13, 1, 21, 16, 3] [1]. There are 21,450 documents in the full collection; less than half of the documents have human assigned topic labels. We used only those documents that had at least one topic, divided randomly into a training set of 9,610 and a test set of 3,662 documents. This partition is similar to that employed in [1], but differs from [13] who use the full collection including unla-

[1] A newly revised version, namely Reuters-21578, is available through http://www.research.att.com/~lewis.

belled documents [2]. The stories have a mean length of 90.6 words with standard deviation 91.6. We considered the 92 categories that appear at least once in the training set. These categories cover topics such as commodities, interest rates, and foreign exchange. While some documents have up to fourteen assigned categories, the mean is only 1.24 categories per document. The frequency of occurrence varies greatly from category to category; *earnings*, for example, appears in roughly 30% of the documents, while *platinum* is assigned to only five training documents. There are 16,039 unique terms in the collection (after performing inflectional stemming, stop word removal, and conversion to lower case).

OHSUMED is a bibliographical document collection[3], developed by William Hersh and colleagues at the Oregon Health Sciences University. It is a subset of the MEDLINE database[9], consisting of 348,566 references from 270 medical journals from the years 1987 to 1991. All of the references have titles, but only 233,445 of them have abstracts. We refer to the title plus abstract as a document. The documents were manually indexed using subject categories (Medical Subject Headings, or MeSH) in the National Library of Medicine. There are about 18,000 categories defined in MeSH, and 14,321 categories present in the OHSUMED document collection. We used the 1990 documents as a training set and the 1991 documents as the test set in this study. There are 72,076 unique terms in the training set. The average length of a document is 167 words. On average 12 categories are assigned to each document. In some sense the OHSUMED corpus is more difficult than Reuters because the data are more "noisy". That is, the word/category correspondences are more "fuzzy" in OHSUMED. Consequently, the categorization is more difficult to learn for a classifier.

4 Empirical Validation

4.1 Performance measures

We apply feature selection to documents in the preprocessing of kNN and LLSF. The effectiveness of a feature selection method is evaluated using the performance of kNN and LLSF on the preprocessed documents. Since both kNN and LLSF score categories on a per-document basis, we use the standard definition of recall and precision as performance measures:

$$\text{recall} = \frac{\text{categories found and correct}}{\text{total categories correct}}$$

$$\text{precision} = \frac{\text{categories found and correct}}{\text{total categories found}}$$

where "categories found" means that the categories are above a given score threshold. Given a document, for recall thresholds of 0%, 10%, 20%, ... 100%, the system assigns in decreasing score order as many categories as needed until a given recall is achieved, and computes the precision value at that point[18]. The resulting 11 point precision values are then averaged to obtain a single-number measure of system performance on that document. For a test set of documents, the average precision values of individual documents are further averaged to obtain a global measure of system performance over the entire set. In the following, unless otherwise specified, we will use "precision" or "AVGP" to refer to the 11-point average precision over a set of test documents.

4.2 Experimental settings

Before applying feature selection to documents, we removed the words in a standard stop word list[18]. Then each of the five feature selection methods was evaluated with a number of different term-removal thresholds. At a high threshold, it is possible that all the terms in a document are below the threshold. To avoid removing all the terms from a document, we added a meta rule to the process. That is, apply a threshold to a document only if it results in a non-empty document; otherwise, apply the closest threshold which results in a non-empty document.

We also used the SMART system [18] for unified preprocessing followed feature selection, which includes word stemming and weighting. We tried several term weighting options ("ltc", "atc", "lnc" , "bnn" etc. in SMART's notation) which combine the term frequency (TF) measure and the Inverted Document Frequency (IDF) measure in a variety of ways. The best results (with using "ltc") are reported in the next section.

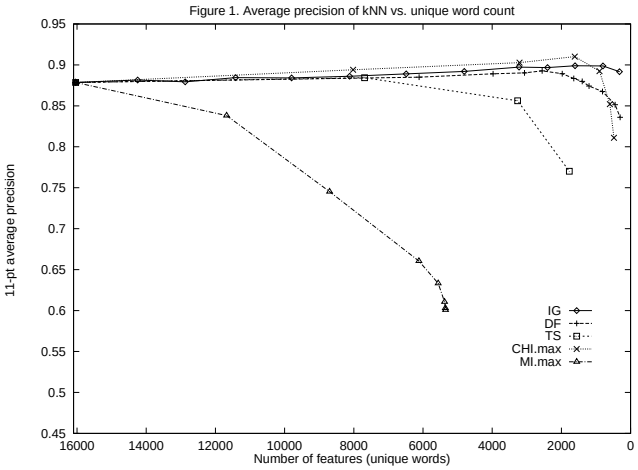
4.3 Primary results

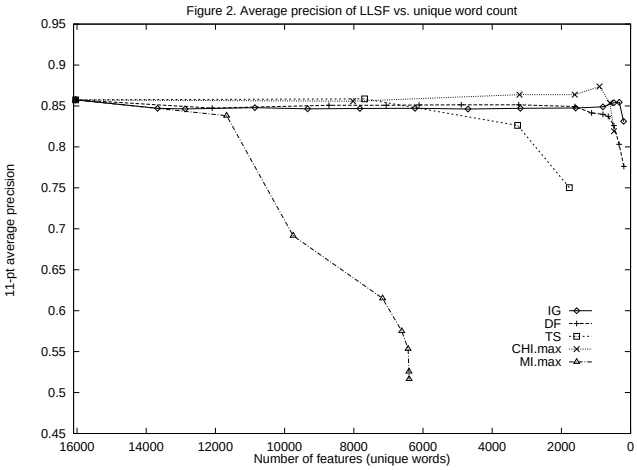
Figure 1 displays the performance curves for kNN on Reuters (9,610 training documents, and 3,662 test documents) after term selection using IG, DF, TS, MI and CHI thresholding, respectively. We tested the two options, *avg* and *max* in MI and CHI, and the better results are represented in the figure.

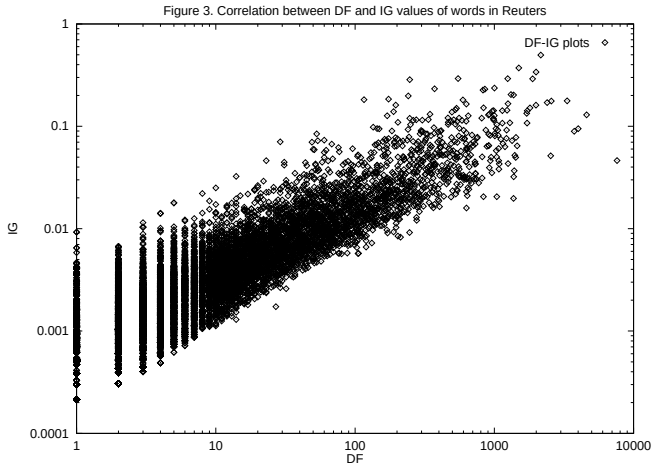
Figure 2 displays the performance curves of LLSF on Reuters. Since the training part of LLSF is rather resource consuming, we used an approximation of LLSF

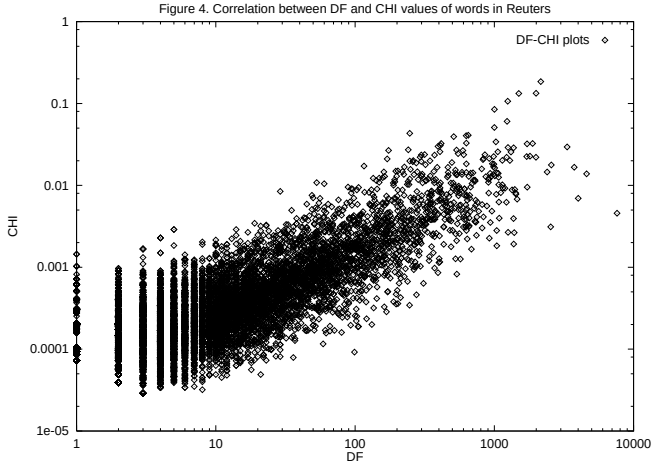
[2]There has been a serious problem in using this collection for text categorization evaluation. That is, a large proportion of the documents in the test set are incorrectly unlabelled. This makes the evaluation results highly questionable or non-interpretable unless these unlabelled documents are discarded, as analyzed in [26].

[3]OHSUMED is available via anonymous ftp from medir.ohsu.edu in the directory /pub/ohsumed.









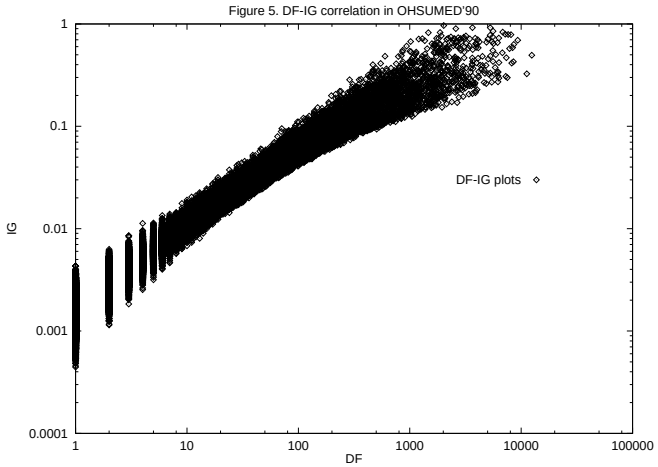
instead of the complete solution to save some computation resources in the experiments. That is, we computed only 200 largest singular values in solving LLSF, although the best results (which is similar to the performance of kNN) appeared with using 1000 singular values[24]. Nevertheless, this simplification of LLSF should not invalidate the examination of feature selection which is the focus of the experiments.

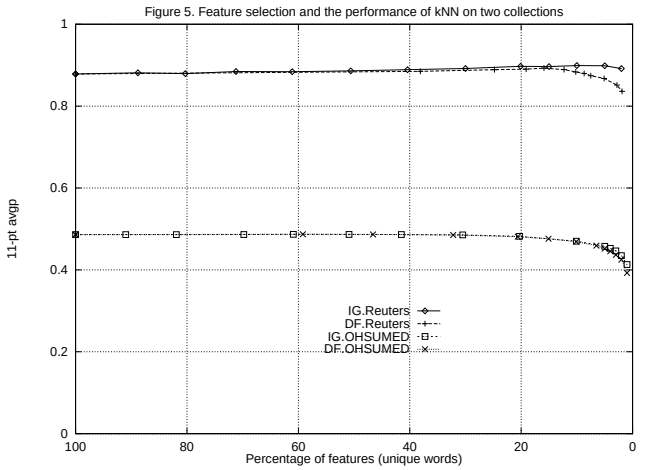
An observation merges from the categorization results of kNN and LLSF on Reuters. That is, IG, DF and CHI thresholding have similar effects on the performance of the classifiers. All of them can eliminate up to 90% or more of the unique terms with either an improvement or no loss in categorization accuracy (as measured by average precision). Using IG thresholding, for example, the vocabulary is reduced from 16,039 terms to 321 (a 98% reduction), and the AVGP of kNN is improved from 87.9% to 89.2%. CHI has even better categorization results except that at extremely aggressive thresholds IG is better. TS has a comparable performance with up-to 50% term removal in kNN, and about 60% term removal in LLSF. With more aggressive thresholds, its performance declines much faster than IG, CHI and DF. MI does not have comparable performance to any of the other methods.

4.4 Correlations between DF, IG and CHI

The similar performance of IG, DF and CHI in term selection is rather surprising because no such an observation has previously been reported. A natural question therefore is whether these three corpus statistics are correlated.

Figure 3 plots the values of DF and IG given a term in the Reuters collection. Figure 4 plots the values of DF and CHI_{avg} correspondingly. Clearly there are indeed very strong correlations between the DF, IG and CHI values of a term. Figures 5 and 6 shows the results of a cross-collection examination. A strong correlation between DF and IG is also observed in the OHSUMED collection. The performance curves of kNN with DF versus IG are identical on this collection. The observations on Reuters and on OHSUMED are highly consistent. Given the very different application domains





of the two corpus, we are convinced that the observed effects of DF and IG thresholding are general rather than corpus-dependent.

5 Discussion

What are the reasons for good or bad performance of feature selection in text categorization tasks? Table 1 compares the five criteria from several angles:

- favoring common terms or rare terms,
- task-sensitive (using category information) or task-free,
- using term absence to predict the category probability, i.e., using the D-cell of the contingency table, and
- the performance of kNN and LLSF on Reuters and OHSUMED.

From Table 1, the methods with an "excellent" performance share the same bias, i.e., scoring in favor of common terms over rare terms. This bias is obviously in the DF criterion. It is not necessarily true in IG or CHI by definition: in theory, a common term can have a zero-valued information gain or χ_2 score. However, it is statistically true based on the strong correlations between the DF, IG and CHI values. The MI method has an opposite bias, as can be seen in the formula $I(t,c) = \log P_r(t|c) - \log P_r(t)$. This bias becomes extreme when $P_r(t)$ is near zero. TS does not have a clear bias in this sense, i.e., both common and rare terms can have high strength.

The excellent performance of DF, IG and CHI indicates that common terms are indeed informative for text categorization tasks. If significant amounts of information were lost at high levels (e.g. 98%) of vocabulary reduction it would not be possible for kNN or LLSF to have improved categorization preformance. To be more precise, in theory, IG measures the number of bits of information obtained by knowing the presence or absence of a term in a document. The strong DF-IF correlations means that common terms are often informative, and vice versa (this statement of course does not extend to stop words). This is contrary to a widely held belief in information retrieval that common terms are non-informative. Our experiments show that this assumption may not apply to text categorization.

Another interesting point in Table 1 is that using category information for feature selection does not seem to be crucial for excellent performance. DF is task-free, i.e., it does use category information present in the training set, but has a performance similar to IG and CHI which are task-sensitive. MI is task-sensitive, but significantly under-performs both TS and DF which are task-free.

The poor performance of MI is also informative. Its bias towards low frequency terms is known (Section 2), but whether or not this theoretical weakness will cause significant accuracy loss in text categorization has not been empirically examined. Our experiments quantitatively address this issue using a cross-method comparison and a cross-classifier validation. Beyond this bias, MI seems to have a more serious problem in its sensitivity to probability estimation errors. That is, the second term in the formula $I(t,c) = \log P_r(t|c) - \log P_r(t)$ makes the score extremely sensitive to estimation errors when $P_r(t)$ is near zero.

For theoretical interest, it is worth analyzing the difference between information gain and mutual information. IG can be proven equivalent to:

$$G(t) = \sum_{X \in \{t,\bar{t}\}} \sum_{Y \in \{c_i\}} P_r(X,Y) \log \frac{P_r(X,Y)}{P_r(X)P_r(Y)}$$

Table 1. Criteria and performance of feature selection methods in kNN & LLSF

Method	DF	IG	CHI	MI	TS
favoring common terms	Y	Y	Y	N	Y/N
using categories	N	Y	Y	Y	N
using term absence	N	Y	Y	N	N
performance in kNN/LLSF	excellent	excellent	excellent	poor	ok

$$= \sum_{i=1}^{n} P_r(t, c_i) I(t, c_i) + \sum_{i=1}^{n} P_r(\bar{t}, c_i) I(\bar{t}, c_i)$$

These formulas show that information gain is the weighted average of the mutual information $I(t, c)$ and $I(\bar{t}, c)$, where the weights are the joint probabilities $P_r(t, c)$ and $P_r(\bar{t}, c)$, respectively. So information gain is also called *average mutual information*[7]. There are two fundamental differences between IG and MI: 1) IG makes a use of information about term absence in the form of $I(\bar{t}, c)$, while MI ignores such information; and 2) IG normalizes the mutual information scores using the joint probabilities while MI uses the non-normalized scores.

6 Conclusion

This is an evaluation of feature selection methods in dimensionality reduction for text categorization at all the reduction levels of aggressiveness, from using the full vocabulary (except stop words) as the feature space, to removing 98% of the unique terms. We found IG and CHI most effective in aggressive term removal without losing categorization accuracy in our experiments with kNN and LLSF. DF thresholding is found comparable to the performance of IG and CHI with up to 90% term removal, while TS is comparable with up to 50-60% term removal. Mutual information has inferior performance compared to the other methods due to a bias favoring rare terms and a strong sensitivity to probability estimation errors.

We discovered that the DF, IG and CHI scores of a term are strongly correlated, revealing a previously unknown fact about the importance of common terms in text categorization. This suggests that that DF thresholding is not just an ad hoc approach to improve efficiency (as it has been assumed in the literature of text categorization and retrieval), but a reliable measure for seleting informative features. It can be used instead of IG or CHI when the computation (quadratic) of these measures is too expensive. The availability of a simple but effective means for aggressive feature space reduction may significantly ease the application of more powerful and computationally intensive learning methods, such as neural networks, to very large text categorization problems which are otherwise intractable.

Acknowledgement

We would like to thank Jaime Carbonell and Ralf Brown at CMU for pointing out an implementation error in the computation of information gain, and providing the correct code. We also like to thank Tom Mitchell and his machine learning group at CMU for the fruitful discussions that clarify the definitions of information gain and mutual information in the literature.

References

[1] C. Apte, F. Damerau, and S. Weiss. Towards language independent automated learning of text categorization models. In *Proceedings of the 17th Annual ACM/SIGIR conference*, 1994.

[2] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information and lexicography. In *Proceedings of ACL 27*, pages 76–83, Vancouver, Canada, 1989.

[3] William W. Cohen and Yoram Singer. 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*, 1996. 307-315.

[4] R.H. Creecy, B.M. Masand, S.J. Smith, and D.L. Waltz. Trading mips and memory for knowledge engineering: classifying census returns on the connection machine. *Comm. ACM*, 35:48–63, 1992.

[5] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. In *J Amer Soc Inf Sci 1, 6*, pages 391–407, 1990.

[6] T.E. Dunning. Accurate methods for the statistics of surprise and coincidence. In *Computational Linguistics*, volume 19:1, pages 61–74, 1993.

[7] R. Fano. *Transmission of Information.* MIT Press, Cambridge, MA, 1961.

[8] N. Fuhr, S. Hartmanna, G. Lustig, M. Schwantner, and K. Tzeras. Air/x - a rule-based multistage indexing systems for large subject fields. In 606-623, editor, *Proceedings of RIAO'91*, 1991.

[9] W. Hersh, C. Buckley, T.J. Leone, and D. Hickman. Ohsumed: an interactive retrieval evaluation and new large text collection for research. In *17th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 192–201, 1994.

[10] D. Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of the Thirteenth International Conference on Machine Learning*, 1996.

[11] K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, 1995.

[12] David D. Lewis, Robert E. Schapire, James P. Callan, and Ron Papka. Training algorithms for linear text classifiers. In *SIGIR '96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996. 298-306.

[13] D.D. Lewis and M. Ringuette. Comparison of two learning algorithms for text categorization. In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, 1994.

[14] Tom Mitchell. *Machine Learning*. McCraw Hill, 1996.

[15] I. Moulinier. Is learning bias an issue on the text categorization problem? In *Technical report, LAFORIA-LIP6, Universite Paris VI*, page (to appear), 1997.

[16] I. Moulinier, G. Raskinis, and J. Ganascia. Text categorization: a symbolic approach. In *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval*, 1996.

[17] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[18] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, Pennsylvania, 1989.

[19] H. Schütze, D.A. Hull, and J.O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *18th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95)*, pages 229–237, 1995.

[20] K. Tzeras and S. Hartman. Automatic indexing based on bayesian inference networks. In *Proc 16th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*, pages 22–34, 1993.

[21] E. Wiener, J.O. Pedersen, and A.S. Weigend. A neural network approach to topic spotting. In *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, 1995.

[22] J.W. Wilbur and K. Sirotkin. The automatic identification of stop words. *J. Inf. Sci.*, 18:45–55, 1992.

[23] Y. Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *17th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 13–22, 1994.

[24] Y. Yang. Noise reduction in a statistical approach to text categorization. In *Proceedings of the 18th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95)*, pages 256–263, 1995.

[25] Y. Yang. Sampling strategies and learning efficiency in text categorization. In *AAAI Spring Symposium on Machine Learning in Information Access*, pages 88–95, 1996.

[26] Y. Yang. An evaluation of statistical approach to text categorization. In *Technical Report CMU-CS-97-127, Computer Science Department, Carnegie Mellon University*, 1997.

[27] Y. Yang and C.G. Chute. An example-based mapping method for text categorization and retrieval. *ACM Transaction on Information Systems (TOIS)*, pages 253–277, 1994.

[28] Y. Yang and W.J. Wilbur. Using corpus statistics to remove redundant words in text categorization. In *J Amer Soc Inf Sci*, 1996.