

Using Corpus Statistics to Remove Redundant Words in Text Categorization

Yiming Yang

Section of Medical Information Resources
Mayo Clinic/Foundation
Rochester, Minnesota 55905, USA

John Wilbur

National Center for Biotechnology Information
National Library of Medicine
Bethesda, MD 20894, USA

Abstract

This paper studies aggressive word removal in text categorization to reduce the noise in free texts and to enhance the computational efficiency of categorization. We use a novel stop word identification method to automatically generate domain specific stoplists which are much larger than a conventional domain-independent stoplist. In our tests with three categorization methods on text collections from different domains/applications, significant numbers of words were removed without sacrificing categorization effectiveness. In the test of the Expert Network method on CACM documents, for example, an 87% removal of unique words reduced the vocabulary of documents from 8002 distinct words to 1045 words, which resulted in a 63% time savings and a 74% memory savings in the computation of category ranking, with a 10% precision improvement on average over not using word removal. It is evident in this study that automated word removal based on corpus statistics has a practical and significant impact on the computational tractability of categorization methods in large databases.

1 Introduction

Removal of non-informative words is a commonly used technique in text indexing and retrieval (van Rijsbergen, 1979; Salton, 1989) to improve the accuracy of the results and to reduce the redundancy of the computation. Non-informative words are often defined by a "stoplist" which typically consists of about 300 or 400 words, including articles, prepositions, conjunctions and some high-frequency words. Most systems apply the same generic stoplist to all text collections without change. The generic stop words are relatively "safe" to remove, that is, their removal rarely causes a significant loss in indexing/retrieval effectiveness; on the other hand, the chance of significant improvement

is also small (Buckley et al, 1993; Yang & Chute, 1994). From a computational efficiency point of view, since a generic stoplist is often much smaller than the vocabularies of real-world text collections, only a limited number of words can be removed from texts, and the computational efficiency cannot be improved by much. Previous experiments in adding collection-specific high-frequency words to a generic stoplist have shown no reliable improvements (Buckley et al, 1993).

In contrast to using generic stop words, Wilbur and Sirotkin developed a novel stop word identification method which allows a far more aggressive removal of words from texts without sacrificing retrieval effectiveness (Wilbur & Sirotkin, 1992). This method uses a collection of training texts to estimate word importance using a score, namely "word strength", in a way that the resulting strength of a word reflects how informative the word is in identifying texts which are related to each other (see Section 2 for the precise definition). The use of relevance information between texts makes a unique feature of word strength, compared to conventional word weighting schemes which are only based on word frequencies in a text collection. In stop word identification, a word is added to a stoplist if its strength is below a certain threshold. Clearly, stop words selected in this manner are collection specific; by changing training text collections, one can obtain different stoplists for various applications in different domains. These lists are usually much larger than a generic stoplist. According to Wilbur's evaluation in a retrieval test on MEDLINE documents, this method reduced 203,040 unique words in 71,311 documents to 50,508 words (a 75% reduction), and the retrieval results were more favorable (based on human judgments) than applying a generic stop-word list of 310 words. In this particular test, queries were randomly selected documents from the database and the task was to find closely related documents in the database. As documents, the queries were longer than most queries a searcher might produce and this may explain why such a high proportion of words could be removed with improved retrieval. In general the proportion of removable words may be less but still vary significant.

In this paper, we apply the Wilbur-Sirotkin stop word identification method to text categorization; we aim for a substantial reduction of the computation cost without sacrificing categorization effectiveness. Text categorization is the assignment of predefined categories to free texts and is tantamount to indexing them. It has wide applications in real-world databases because many of them use a controlled vocabulary, i.e. subject categories, to index their data for retrieval purposes. Manual categorization remains the dominant method in practical databases, which is costly and labor-intensive. MEDLINE, for example, the world's largest on-line bibliographic database, spends over two million

dollars for indexing about 350,000 new entries each year by human indexers. There is thus a strong motivation for automatic or semi-automatic text categorization. Technically speaking, text categorization is highly related to text retrieval, in the sense that many techniques developed for one are applicable to the other. For example, many retrieval systems provide a ranked list of documents to a given query instead of making binary decisions about "related" or "non-related" documents; similarly, many categorization systems also generate a ranked list of candidate categories for a given text instead of making binary decisions about "correct" or "incorrect" categories. A category ranking system can be useful in either computer-assisted human categorization through user interaction, or in automated categorization when a threshold on the ranks or relevance scores of categories is determined. There is a particular problem, however, which makes text categorization very different from text retrieval. That is, often a text and a category are conceptually related, but the concepts in the text happen to be expressed in words which are different from the name of the category. We refer to this as the vocabulary difference problem between free texts and controlled categories of a particular database. Since the vocabulary differences are usually large, a shared-word based matching, which is the basic mechanism in many retrieval systems, is unavoidably ineffective for text categorization.

The importance of using human knowledge to solve the vocabulary difference problem has been recognized, and statistical learning about human knowledge in the form of past relevance judgments between texts and categories has been a major focus in recent research. Statistical categorization methods include the Least Squares Polynomial in the Darmstadt Indexing Approach that uses a regression model to predict categories based on word/category co-occurrences in training texts (Fuhr et al. 1991), the Linear Least Squares Fit (LLSF) mapping that learns a context-sensitive function for category ranking based on text/category-set training pairs (Yang & Chute, 1992, 1993 July, 1994), Bayesian belief networks that use conditional probabilities of categories given words (Tzeras & Hartmann, 1993), and the Memory Based Reasoning (Creedy et al. 1992) (Masand et al. 1992) and Expert Network (Yang, 1994) approaches that predict categories based on a Nearest Neighbor search. All these methods share a common feature, that is, they use manually categorized training texts to predict categories of arbitrary texts. Significant improvements of these statistical approaches have been observed, compared to alternatives such as word-based matching methods which do not use any human knowledge, and thesaurus-based or rule-based methods which are heavily dependent on manually coded human knowledge.

A serious question about a statistical categorization method is its computational tractability when applied to

large databases. MEDLINE, for example, has about 7 million documents with a vocabulary size at a level of a hundred thousand unique words (approximated by counting the number of unique words in the 1994 version of the Metathesaurus of the Unified Medical Language System developed by the National Library of Medicine; refer to Lindberg & Humphreys, 1990), and an indexing language including about 17000 subject categories (Medical Subject Headings, or MeSH, 1993). This means that a regression method which uses the words in a text to predict the categories of this text would need to compute the correlation coefficients between a hundred thousand input variables (words) and 17000 output variables (categories) based on a very large number of training documents. Such a problem is much larger than the problem sizes in traditional statistical analyses, where the total numbers of variables are typically tens or less. However, not all the words in a text are necessarily independent variables, and obviously, some words are more informative than others in terms of identifying the categories of the text. Therefore, it should be possible to substantially reduce the problem sizes in statistical learning without sacrificing categorization effectiveness. The question is how.

One solution for the size reduction problem is to apply the Wilbur-Sirotkin method in the preprocessing stage of a categorization system, i.e. to remove non-informative words aggressively from free texts before applying a categorization method to these texts; other solutions include using the Singular Value Decomposition on training texts to find orthogonal input variables in a regression model (Yang, 1995 SIGIR), and decomposing a learning problem into subproblems (Yang, 1995 IJCAI). In this paper we focus on the first solution. We chose two statistical categorization methods to study the effects of aggressive word removal, the LLSF mapping and the Expert Network (ExpNet). Both methods are developed for learning from past human relevance judgments, and are particularly effective in bridging the vocabulary differences between free texts and controlled categories (Yang & Chute, 1994; Yang, 1994). However, they differ statistically and in computation. LLSF is a regression method which requires intensive computation in training to obtain word-category coefficients. ExpNet, on the other hand, is a Nearest Neighbor (NN) classification method which predicts the categories of a testing text based on the categories of its neighbors in training texts. ExpNet does not need any previous training, but requires an on-line search of the NNs for each testing text, where the neighborhood is determined based on the shared words in this testing text and training texts. A real-time response of the search would be a computational bottleneck when the training sample is very large. We are interested in exploring how these two very different learning methods respond to aggressive word removal, and whether a significant improvement can be obtained in their computations.

In addition to the two statistical categorization methods, we also include a word-based matching method in this study. If there were no prior humanly categorized documents from which to learn, word based matching or some similar surface processing may be the only viable automatic method of text categorization. One is likely to face such a situation any time a new database is constructed or substantial changes are made in the indexing vocabulary of an already established database. For this reason we deem it important to understand the behavior of word matching categorization methods. We chose the SMART system (Salton, 1991) for this part of the study, for its representative performance among word-based matchers and its rich word weighting options for experiments.

In the following parts of the paper, Section 2 outlines the stop word identification method. Section 3 describes the categorization systems used in this study, and the testing conditions with these systems in the experiments. Section 4 describes the data for training and testing, and the measurements for empirical validation. Section 5 analyzes the results of word removal in different categorization systems on various data collections.

For convenience, we use document as a generic word for a free text, which can be the title plus the abstract of an article in a bibliographic database, or a surgical report in a patient record.

2 Stop Words Identification

Word strength measures how informative a word is in identifying related documents, and is computed based on word distribution over related documents. The strength of a word, t , is defined as the probability of finding t in a document which is related to any document in which t occurs,

$$s(t) \stackrel{\text{def}}{=} P_r(\text{word } t \text{ is in document } y | \text{word } t \text{ is in document } x), \quad (1)$$

where x and y denote an arbitrary pair of distinct but related documents. For computing word strength, one needs a training corpus where relevance judgments between documents are available. It would be ideal to use human judgments as the gold standard. Such relevance judgments, however, are often not available in real-world applications. Wilbur and Sirotkin have shown that one can relax the relevance criterion by assuming two documents are related to each other if they have many words in common. That is, one can use the conventional cosine-coefficiency of two vectorized documents to measure their similarity, and identify a pair of documents as related if their cosine-similarity

value is above a threshold. Using the pairs of related documents, one can approximately compute word strength as

$$s(t) \approx \frac{\text{number of document pairs in which word } t \text{ co-occurs in both documents}}{\text{number of document pairs in which word } t \text{ occurs in the first document}} \quad (2)$$

where the "first document" can be any training document. That is, there are no constraints on the first document of a pair; if (x, y) is a pair of related documents, then (y, x) is also a pair of related documents.

Our procedure of stop word identification consists of the following steps:

- (1) use a standard stoplist to eliminate non-informative words from training documents;
- (2) compute the similarity values of all pairs of training documents;
- (3) select the document pairs whose similarity values are above a **document relevance threshold** (chosen experimentally) as related document pairs;
- (4) compute the strength for each word using the related document pairs;
- (5) select the words with strength values equal to or less than a **stop word threshold** (chosen experimentally) as stop words.

Step (5) above is slightly different from the original method by Wilbur and Sirotkin. In the original method, a word strength is compared with the expected word strength of a hypothetical word of the same frequency which is distributed randomly in the database. If the word strength is not at least two standard deviations above the strength of such a hypothetical randomly distributed word it is designated a stop word. In this paper, we use a simplified version, that is, we apply a universal threshold to all the words with different frequencies.

Note that there is a fundamental difference between the word strength introduced here and the commonly used word weight in document retrieval, the Inverse Document Frequency (IDF),

$$IDF_t \stackrel{\text{def}}{=} \log \left(\frac{\text{number of documents in the entire collection}}{\text{number of documents with word } t} \right)$$

IDF is based on the assumption that rare words are more informative than common words, roughly speaking. In an extreme case, a word occurring in only one document has the highest IDF value. In contrast, such a word has the lowest strength of 0, according to formula (2), because it is non-informative in identifying any related documents in a training sample. In another extreme case, suppose a word occurs in two documents only, and the two documents are the only pair of related documents containing this word, then the strength of this word is the highest value of 1.

For words occurring only in non-related documents, they have the lowest strength of 0, regardless of whether they are common words or not. The underlying assumption of word strength is that the shared words among related documents are more informative than others. Therefore, words with high strength values are often common words, or relatively common. However, the inverse assertion does not apply, i.e. a common word does not necessarily have a high strength value, and a rare word may have a high strength.

In summary of the above discussion, word strength and IDF are defined for two different purposes. The former is for removing non-informative words, assuming that the intended information within some words is implied by other words. IDF, on the other hand, is for word weighting, and it ignores the implication among words or the redundancy of words. We apply both methods to our text categorization experiments: we use Wilbur's word strength for obtaining a list of stop words, and IDF (in combination with other kinds of word weights such as within-document term frequency) to weight the remaining words after the word removal.

3 Categorization Methods for Comparison

We assume that the functionality of document words in categorization is dependent on particular methods, and that the effects of aggressive word removal may vary in different methods. To verify these assumptions, we choose two statistical classification methods and one word-based matching method for comparison, namely the LLSF mapping, the Expert Network, and the WBM described below.

(1) **LLSF mapping** is a statistical learning method which can be used for both document retrieval and document categorization (Yang & Chute, 1992, 1994). We use LLSF in this paper as a statistical classifier which uses a training sample of manually categorized documents to establish empirical connections between document words and categories. LLSF mapping does not rely on any shared words or shared tokens in document representation and category representation. That is, we represent documents using their own words, and represent categories using their codes which are different tokens from document words. LLSF establishes document-word/category-code connections based on training documents and their categories, and then uses these connections to estimate the relevance scores of categories of arbitrary documents. This method guarantees the globally-minimized squares error in the mapping from training documents to training categories. LLSF is particularly effective for a mapping from free vocabularies of documents to the controlled vocabulary of a database indexing language, compared to word-based matching methods

and thesaurus-based methods; superior performance in average recall-precision has been observed in our previous studies (Yang & Chute, 1994). This method, however, has a cubic time complexity in the training phase, which is an obstacle in applying LLSF to very large document collections; it is relatively fast in the testing phase (category ranking given a text) once the training is done. It will be beneficial if aggressive word removal can significantly reduce the training time of LLSF without significantly sacrificing categorization effectiveness.

(2) **ExpNet** is another statistical learning method which can be used for both document retrieval and document categorization (Yang, 1994). We use ExpNet in this paper to achieve the same goal of LLSF, i.e. to establish empirical connections between document words and categories. Similar to LLSF, ExpNet also does not rely on any shared words or shared tokens in document representation and category representation, and is as effective as LLSF in mapping from free vocabularies of documents to a controlled vocabulary of categories, according to our previous study (Yang, 1994). ExpNet takes a Nearest Neighbor approach to classification, on the other hand, which is theoretically different from the fitting technique of LLSF. We are interested to see how sensitive these two different methods are in response to aggressive word removal. From a computational point of view, it takes ExpNet little time for "training" (i.e. pre-indexing training documents), but an online search of Nearest Neighbor documents is required each time a testing document is given. A real-time response remains a nontrivial problem when applying ExpNet to very large document collections. It will be beneficial if aggressive word removal can significantly reduce the Nearest Neighbor search in ExpNet without significantly sacrificing categorization effectiveness.

(3) **Word-based matching (WBM)** is a basic search method in practical databases. It is widely used in software tools for users to formulate their queries using indexing terms (subject categories) of particular databases (refer to Evans JT, 1992, about GRATEFUL MED, a user interface to MEDLINE retrieval), and for computer assisted categorization of free documents (Chute et al. 1994). In word based matching, the number of shared words determines the degree of match between documents and category names (descriptors) which are typically given in a controlled-vocabulary taxonomy. Word-based matching is commonly used because it is simpler than statistical learning methods; it does not require any training data, and does not learn from human relevance judgments. Its fundamental weakness, however, is that it cannot match a document and a category which are conceptually related but happen to share few if any words. In our previous papers, we used word-based matching in comparison with LLSF and ExpNet to quantitatively analyze the vocabulary difference between free documents and controlled

categories in real-world applications, and to observe how effective the statistical learning methods were in solving the vocabulary difference problem in text categorization (Yang & Chute, 1992, 1993 November, 1994; Yang, 1994). In this paper, we use word-based matching for a different purpose, that is, to study the effect of aggressive word removal on a basic, non-learning search method. Since word-based matching is heavily dependent on word appearances in documents and category names, it is quite different from LLSF and ExpNet in which shared words are not crucial for document-category matching. Observing how word-based matching reacts to aggressive word removal enables us to study the word removal method from a different perspective. Also, it is useful to know how many words are redundant or non-informative, and how much the computational cost can be reduced, if aggressive word removal is effective in word-based matching.

We use the SMART system as the search engine for word-based matching. SMART, developed by Salton's group (Salton, 1991), is one of the better-known systems in document retrieval. It provides a mechanism for query-document matching based on shared words, and we adapt it to document-category matching. Both a document and a category name are represented using a vector whose dimensions are words in the category vocabulary, and the cosine-similarity between the document vector and the category vector is computed and used as the relevance score of the category with respect to the document. SMART allows the use of statistical word weights such as term frequency (TF), Inverse Document Frequency (IDF), and their combinations ($TF \times IDF$) with various forms of normalization of those weights. We ran the SMART system (version 10) with the default parameter settings, including word weighting options of binary weights (labeled as "bnn" and "bnc" in the SMART nomenclature), TF weights (labeled as "nnn" and "lnc" in SMART), IDF weights (labeled as "btc" in SMART), $TF \times IDF$ word weights (labeled as "atc" and "ltc" in SMART) and other combinations (labeled as "ltc.lnc" and "lnc.ltc" in SMART). We refer to the best result (using the "atc" version of $TF \times IDF$) among these choices in the comparison of SMART with other methods. No claim is made that this result is the best possible for SMART, given that we have not performed an exhaustive comparison with all possible parameter settings for word weights of SMART on our data. It also should be pointed out that SMART is not designed for text categorization, and its performance in text categorization should not be interpreted as an indication of its performance in text retrieval.

We did not use the relevance feedback component of SMART, because it cannot use past relevance judgments to predict categories for new documents. Recall that in document retrieval, relevance feedback is designed for

fixed queries; it requires a human user to feedback related documents for each query, and cannot use this relevance information to predict answers to other queries. When applying relevance feedback to document categorization, documents have to be fixed; it would require feedback of related categories for each document, and cannot use this relevance information for predicting the categories of other documents. It is a rather unrealistic requirement to fix documents, considering that categorization in real-world databases is used to index new documents. It is also an undesirable constraint to not use past relevance judgments for training, because large amounts of previously categorized documents by humans are often available in databases, and significant improvements in categorization effectiveness have been observed in previous studies when such training data were used.

One could still argue the possibility of using on-line relevance feedback for each document, admitting the disadvantage of not being able to use previously categorized documents as training data. One could run an initial search for categories of a given document using word-based matching, and then have the user identify the correct categories among the few top-ranking candidate categories. If any correct categories are found, add the words in the names of these categories to the document, and re-run a word-based search using the expanded document for the categories which were not found in the initial search. While this sounds analogical to the relevance feedback approach to document retrieval, we doubt its effectiveness because of the very different nature of document categorization compared to document retrieval. Relevance feedback works for retrieval because documents relevant to a given query often share important words among themselves. The categories of a document, on the other hand, are often orthogonal concepts, and rarely share words among their names. Thus using a few user-identified categories of a document to expand this document may not be helpful for finding other categories of the document. It will be particularly useless in applications where most documents have only one correct category. For example, 99.8% of the patient-record texts mentioned in Section 4.1 have a uniquely matched category. Relevance feedback would require the user to identify the matched category for most cases; once the category is identified, there is no need for further search. Nevertheless, using relevance feedback in document categorization remains unexplored; to draw firm conclusions about its effectiveness requires empirical study. Since our major focus in this paper is on word removal in categorization methods which have already been evaluated, and there is no published work about using relevance feedback in text categorization, we leave further analysis of this issue to future research.

One could also argue the possibility of using relevance feedback in a modified fashion in that no user feedback is

required. That is, given a new document, run a word-based search for training documents which are lexically similar to the given document. Then use the category names (previously assigned by humans) of these training documents to expand the new document, and re-run a word-based search for categories using the expanded document. Such an approach fundamentally changes relevance feedback to Nearest Neighbor classification which enables the use of past relevance judgments by humans in predicting categories of new documents. In other words, such a use of "relevance feedback" in text categorization is essentially equivalent to ExpNet which is already included in this study.

4 Data Collections and Evaluation Measurements

4.1 Data Collections

To observe the effectiveness of aggressive word removal across different domains or applications, we use four document collections taken from the patient record archive at the Mayo Clinic, the MEDLINE bibliographical database, a molecular biology database at the National Library of Medicine, and the CACM information retrieval test collection in the computer science field. These document collections have already been used for evaluations of retrieval/categorization systems. We name these collections as SURCL, MEDCL, GENCL and CACMCL.

(1) SURCL is a collection of patient-record texts from the Mayo Clinic archive. The patient records at Mayo include diagnoses and operative reports in natural language texts written by physicians. These texts are manually categorized by experts for the purpose of billing and research. About 1.5 million diagnoses and operative reports are manually coded each year. For this experiment, we arbitrarily chose the cardiovascular subdomain of the canonical classification system ICD-9-CM (International Classification of Diseases, 9th Revision, Clinical Modifications), and used the 6134 surgical procedure reports in this subdomain from the 1990 patient records. We sorted the procedure/category pairs by category and arbitrarily split these pairs into odd and even halves. The odd-half was used as the training set which contained 3067 texts with duplicates, or 1461 unique texts; the even-half was used as the testing set which contained 3067 texts with duplicates, or 1492 unique texts. About 58% of the testing texts had an identical text in the training set; about 99% of the words and 97% of the categories in the testing set had occurrences in the training set. About 99.8% of the training and testing texts had a uniquely matched category; the rest had two or three categories. The average length of texts was about 9 words. There were totally

281 categories in the cardiovascular subdomain of ICD-9-CM. The chance of a correct categorization of a procedure text by a random assignment was 1 in 281, or 0.36%.

We used the procedure texts in both the training set and the testing set to compute word strengths. We should clarify that two different kinds of training data are involved here: one for statistical learning about text categorization, and another for word strength estimation. The former are documents with their manually assigned categories; this kind of training data is only needed in LLSF and ExpNet but not in WBM. The latter are documents only, and the categories of these documents are irrelevant to word strength estimation. For the needs of statistical learning methods, we split each document collection into two halves, and use one half for training and another half for testing. In word strength computation, on the other hand, we use all the documents in each collection (or their superset, if available). In the explanations of the document collections, we refer to a "training set" as the training data for statistical categorization methods, if not otherwise specified.

(2) **MEDCL** is a collection of MEDLINE documents. This data set was originally designed for an evaluation of the Boolean search of MEDLINE retrieval (Haynes et al. 1990), and has been used for evaluations of other retrieval and categorization systems (Hersh et al. 1992) (Yang & Chute, 1993 July, 1993 November, 1994). We take the words in the title and abstract of each record without distinction, and call these words together a document; we did not use the records where the abstracts were missing. The resulting collection contains 2344 documents, and each document has categories assigned by MEDLINE indexers. We arbitrarily chose a quarter (586 documents) of these documents for training, and the remaining ones (1758 documents) for testing. There were no duplicate documents in the entire collection, and consequently, none of the testing documents was identical to a training document. There were about 168 words and 17 categories per document on average. The training set contained 7813 unique words and 1832 unique categories. The testing set contained 14,339 unique words and 3430 unique categories; about 42% of the words and 46% of the categories in the testing set had occurrences in the training set. There were 4020 unique categories in total, including the training set and the testing set; these 4020 categories were used as the candidate space of the categorization tests. The chance of a random assignment being correct was 17 in 4020, or 0.42%. We used all 2344 documents to compute word strengths.

(3) **GENCL** is a collection of documents in the domain of molecular biology/genetics. These documents were originally from the MEDLINE database, and were used to augment sequence records in a DNA and protein database

for research purposes at the National Center for Biotechnology Information, National Library of Medicine. This original collection contained 71,311 documents and was used by Wilbur and Sirotkin to evaluate the word removal method in document retrieval (Section 1). This collection was too large for the training of LLSF due to limitations of our current algorithms and computing facilities. We therefore chose a representative subset. We randomly picked 100 documents from the 71,311 documents, and for each document we further chose 50 top-ranking documents based on the cosine-similarity. This resulted in 100 document groups each containing 51 documents. We arbitrarily chose one sixth of the documents from each group for training, and another one sixth of the documents for testing. After removing the duplicate documents, we obtained a training set of 839 documents and a testing set of 838 documents. None of the testing documents was identical to a training document; about 63% of the words and 66% of the categories in the testing set had occurrences in the training set. There were about 46 words and 12 categories per document on average. The 100 document groups contained 4871 unique documents, 12,269 unique words and 2663 unique categories. We used the 2663 categories as the candidate space of the categorization tests. The chance of a random assignment being correct was 12 in 2663, or 0.45%. The 4871 documents were used in computing word strengths.

(4) **CACMCL** is a subset of documents derived from the CACM collection which is one of the standard information retrieval test collections (Fox, 1990). The CACM collection consists of 3703 records each of which contains fields of "title", "abstract", "keys" (keywords), "categories", "author", etc. We take the words in the fields of "title", "abstract" and "keys" without distinction, and call these words together a document. The categories are defined in the Classification System for Computing Reviews (CSCR) and were assigned by humans to documents (ACM Guide to Computing Literature, 1984). We used a subset of the 3703 documents in our experiments. We eliminated documents with an empty category field as obviously unsuitable for the study. We also eliminated documents with empty abstract fields because our primary interest is in documents with the potential for significant word removal. Another consideration in document selection was the type of category codes. Two versions of CSCR category codes had been used in CACM: the original version and an updated replacement version. To avoid potential inconsistencies in the data, we only chose the documents from the larger set that were classified by the original codes. The resulting collection consists of 1121 documents; we refer to it as CACMCL. We further arbitrarily split CACMCL into two halves, and used the first half (561 documents) for training, and the second half (560 documents) for testing. None of the testing documents was identical to a training document; about 60% of the words and 87%

of the categories in the testing set had occurrences in the training set. There were about 120 words and 3 categories per document on average in the CACMCL collection. There was a total of 204 categories in CSCR, and we used the entire set as the candidate space of the categorization tests. The chance of a random assignment being correct was 3 in 204, or 1.5%. Since the total of 1121 documents in the CACMCL collection is rather small for word strength estimation, we used the superset of 3703 CACM documents instead.

Preprocessing was applied to these document collections for the removal of punctuation and numbers, and for changing uppercase letters to lowercase; no stemming was applied.

4.2 Evaluation Measurements

In order to measure the effects of word removal, we define word cut ratios, time saving ratio, memory saving ratio, and recall and precision of text categorization as follows:

$$\text{unique word cut ratio} \stackrel{\text{def}}{=} \frac{\text{number of unique words in document set after word removal}}{\text{number of unique words in document set without word removal}}$$

$$\text{word occurrence cut ratio} \stackrel{\text{def}}{=} \frac{\text{number of word occurrences in document set after word removal}}{\text{number of word occurrences in document set without word removal}}$$

$$\text{time saving ratio } t(x) \stackrel{\text{def}}{=} \frac{\text{saved computation time at word cut ratio } x}{\text{computation time when not using word removal}}$$

$$\text{memory saving ratio } s(x) \stackrel{\text{def}}{=} \frac{\text{saved computation space at word cut ratio } x}{\text{computation space when not using word removal}}$$

For convenience, we use "word cut ratio" to mean unique word cut ratio, if not otherwise specified. To evaluate the effectiveness of a categorization system, we refer to human assigned categories to a document as the correct categories of this document, and use the conventional recall and precision to measure the performance of a system:

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

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

Given a document, for recall thresholds of 10%, 20%, ... 100%, the system assigns in decreasing score order as many categories as needed until a threshold is achieved, and computes the precision value at that point; the resulting 10 point precision values then are averaged for a global measure of the system performance with respect to the document. For a set of documents, the 10-point average precision values of individual documents are further averaged to obtain

the global measure of the system over the entire set. We will refer to precision as the 10-point average precision over a testing set of documents in later discussions, if not otherwise specified.

Average precision is the most commonly used measure in evaluations of categorization systems, because many systems provide a ranked list of categories for a given text instead of binary decisions over categories (Fuhr et al. 1991; Tzeras & Hartmann, 1993; Creecy et al. 1992; Yang, 1994; Yang & Chute, 1994; Apte et al, 1994; Rilloff & Lehnert, 1994). A ranked list, of course, can be used to obtain binary decisions by setting a threshold. There are discussions about whether binary decisions at certain thresholds should be used instead of or in addition to k -point average precision in evaluating categorization effectiveness, about the application-dependent aspects in threshold setting, and about other possible measures for evaluation (Lewis, 1991; Yang & Chute, 1994). These issues, however, are open research questions, and are not the focus of this paper.

5 Results

5.1 Effects of Word Removal on Categorization Effectiveness

In Section 2, we mentioned that a stoplist can be obtained using a word strength threshold. This means that we can test word removal at different degrees of aggressiveness using different values of the threshold. The lowest degree of word removal is "the standard cut" which uses a stoplist derived from the SMART system. The SMART stoplist has 571 words. We applied our preprocessing to this list for consistency with our preprocessing of documents, and obtained a list of 538 stop words. The original stoplist was shortened because we removed letters after apostrophe, e.g. the stop words "they'd", "they'll", "they're" and "they've" were reduced to word "they", and "it's" and "let's" were reduced to "it" and "let", respectively. We call this shortened list "the standard stoplist" for convenience. The more aggressive stoplists were obtained by setting the threshold at word strength values of 0, 0.1, ..., 0.9, selecting the words whose strength value is less than or equal to the threshold, and adding these selected words to the standard stoplist. So we obtained 10 increasingly larger and therefore more aggressive stoplists in addition to the standard stoplist. Figures 1, 2 and 3 show respectively the precision curves of LLSF, ExpNet and WBM on different document collections. Figure 4 compares the precision curves of these three methods on the CACMCL collection. For each system on a particular collection, we applied the 11 stoplists mentioned above to study their

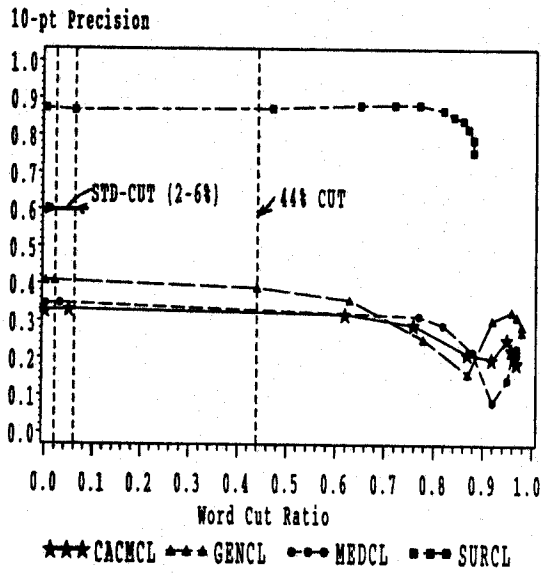


Figure 1. Word cutting in LLSF on different collections.

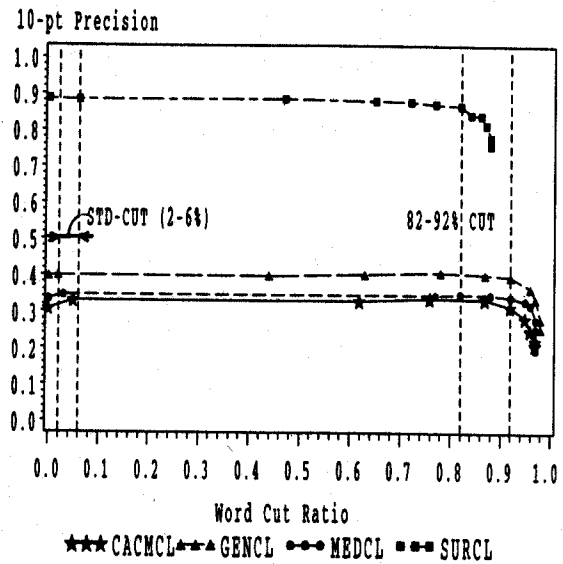


Figure 2. Word cutting in ExpNet on different collections.

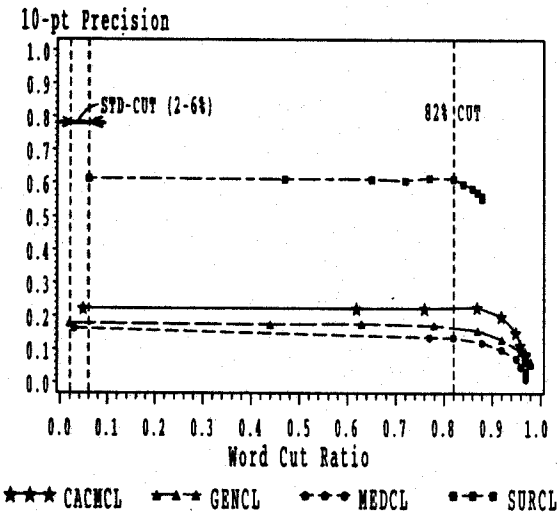


Figure 3. Word cutting in WBM on different collections.

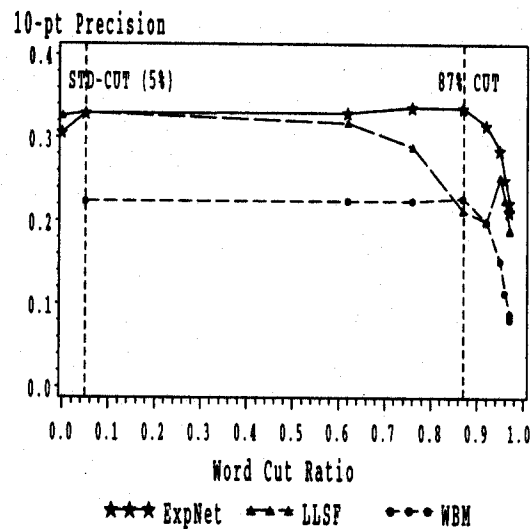


Figure 4. Word cutting in different methods on CACMCL.

effects in categorization precision. When applying such an aggressive stoplist to a document, it is possible that all the words in the document are on the stoplist. To avoid a 100% word removal from a document, we add the following constraint to our method:

- (1) apply a stoplist only if the resulting document contains at least one word;
- (2) otherwise, try the closest and less aggressive stoplist, if applicable.

Comparing Figures 1-4, we can see the reactions of the different methods to aggressive word removal, including similarities and differences. A common observation is that all of these categorization systems have a relatively flat

precision curve at word cut ratios which are far more aggressive than the standard cuts. ExpNet and WBM, for example, have a flat precision curve with up to 82-92% unique word cuts on all the data collections, while the standard word cut only removed 2-6% of the words in these collections. The precision curves of LLSF on three data collections except SURCL are less flat than the curves of ExpNet and WBM; nevertheless, in the worst case of LLSF, its precision curve remains relatively flat with up to 44% word cut on the GENCL collection, which is still a much more aggressive cutoff than the 2-6% standard cuts. These testing results clearly indicate that the amounts of noisy or redundant words in free texts largely exceed the amounts of words in a generic stoplist, and that these noisy or redundant words can be effectively identified and removed using domain-specific word strengths.

The second common observation in the testing results is that all the methods had a better performance on SURCL than on other data collections, although the reason for the better performance of one method may be different than the reason for another method. For ExpNet and LLSF, the better performance came from better training data. That is, about 99% of the words and 97% of the categories in the testing documents of SURCL have occurrences in the training set, while these percentages in the other collections are much lower. Also, the average length of a document in SURCL is 9 words, and 99.8% of those documents have one correct category only. The documents in the other data collections, on the other hand, have an average length of 120-168 words, and the number of correct categories of a document ranges from a few to more than one hundred; there is no information available to specify the correspondence between a shorter piece of a document and a category. So, SURCL provided much more precise relevance judgments for training than the other collections. As for WBM, its better performance on SURCL than on other data sets came from a different reason, i.e. establishing the match between a short surgical report and category names is a much easier or less ambiguous problem than matching a long document and category names.

The third common observation in the testing results is that both LLSF and ExpNet outperformed WBM significantly when aggressive word removal is not used; with aggressive word cuts, ExpNet significantly outperformed WBM at all the word cut thresholds, while LLSF significantly outperformed WBM at most of the thresholds except a few high-middle range thresholds (refer to the discussion later in this Section). It is evident that the statistical learning in LLSF and ExpNet successfully solved the vocabulary difference problem between words in free documents and controlled categories, while word-based matching is fundamentally deficient for such a problem.

As for the different parts of these categorization methods in their response to word removal, ExpNet appeared

to have the best reaction. With a 82% word cut (reducing the vocabulary from 1244 words to 230 words) on SURCL, the precision loss was only 2% compared to no word removal; with a 92% word cut (reducing 16187 unique words to 1271) on MEDCL, the precision improved by 2%; with a 92% word cut on GENCL (reducing 7654 unique words to 622), there was no precision loss; with a 87% word cut (reducing 8002 words to 1045) on CACMCL, the precision improvement was 10%. These results indicate that ExpNet is only sensitive to a small portion of the words in documents, and that these words can be identified using word strengths. In other words, the neighborhood among training documents either did not change by much with up to 82-92% word cuts, or changed into better clusters based on categories of documents. This led to the equally good or even better results of ExpNet because its category ranking is based on the categories of the NNs of a given document.

WBM has a pattern similar to ExpNet in its reaction to aggressive word cuts, that is, its curves on the four data collections are relatively flat. This means that word-based matching is also only sensitive to a small portion of the words in documents, and that redundant words can be well detected using word strengths. On the other hand, the range of aggressive word cuts without significant precision loss in WBM is narrower than the range in ExpNet: with about 82% word removal, it had no precision loss on SURCL and CACMCL, but had a 8-17% precision loss on MEDCL and GENCL, compared to the results when using standard cuts (by the default setting of WBM, the standard stop words were always removed). This is not surprising because an aggressive word cut may reduce lexical variants of concepts in documents, and this may reduce the chance to match documents and categories based on shared words.

LLSF displayed a rather different aspect in its reaction to aggressive word cuts, compared to ExpNet and WBM. There is a "valley" in its precision curves when the middle-high range stoplists were applied to MEDCL, GENCL and CACMCL, although no such valley is in its curve on SURCL. These valleys indicate a potential problem of word removal in context-sensitive categorization methods. By context-sensitive we mean the category ranking is sensitive to word co-occurrences in documents, i.e. the "contexts" under which words are used. We have shown the context-sensitivity of LLSF in a previous paper (Yang & Chute, 1994). According to observed results, it seems that the LLSF method is sensitive to both the words with high strength values and the words with middle range values. Aggressive word removal at certain degrees may severely disrupt the original contexts, and impose inappropriate context constraints instead in the training of LLSF. As a result, a significant precision loss of LLSF appears in

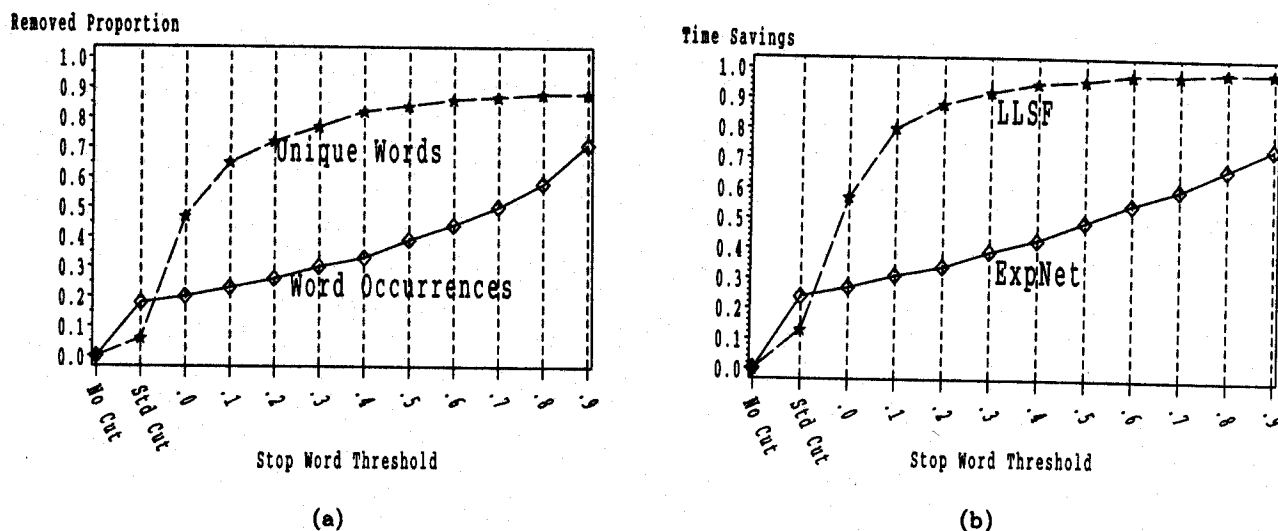


Figure 5. Word cut ratios and time savings on SURCL.

the middle-high range of word cut ratios. The inappropriate contexts may be eliminated when words are further removed, and the precision may then return to a higher level. This is our interpretation of the valleys in the curves of LLSF on MEDCL, GENCL and CACMCL. As for the absence of the valley-shape of LLSF on SURCL, the likely reason is the application-dependent feature of this data collection. The degree of word independence may be different in this application than in the others. It is possible that the words used by physicians in surgical reports are less ambiguous than the words used by article authors, and the word meanings in surgical reports are often less context-dependent than the word meanings in articles. Consequently, aggressive word cuts, which presumably change the original contexts, would have less influence in the categorization precision of LLSF on SURCL than on the other document collections.

5.2 Effects of Word Removal on Computational Efficiency

We have demonstrated aggressive word removal from documents without significant precision loss for different categorization methods. The question is how much improvement in computational efficiency can be gained by doing so, and what is the trade-off in categorization precision, if any? The answers to these questions are dependent on the algorithms implementing the categorization methods.

In LLSF, the most intensive part of the computation is the Singular Value Decomposition (SVD) for which we currently use the conventional LINPACK algorithm (Dongarra et al. 1979). For a training sample of m documents

and n unique words (assuming $m \geq n$), the SVD algorithm has a time complexity of $O(m^2n)$, and a space complexity of $O(mn)$ (Golub and Van Loan, 1989). Suppose a word cut reduces the vocabulary size from n to n' , then the complexities are reduced to $O(m^2n')$ in time and $O(mn')$ in space. The time saving ratio and the memory saving ratio then are both proportional to $\frac{n-n'}{n}$ which is the ratio of unique word cut (defined in Section 5.1). In ExpNet, on the other hand, a major part of the computation is to find the NNs of a given testing document, where the NNs are training documents which are closest to the testing document by the measurement of cosine-similarity. The algorithm for the similarity computation (SIM) has a time and space complexity proportional to the number of word occurrences in documents (Yang, 1994). So the time saving ratio and the memory saving ratio then are both proportional to the word occurrence cut ratio. Figure 5(a) shows the two kinds of word cut ratios when applying different stop word thresholds on the SURCL document collection. Figure 5(b) shows the observed time savings in the SVD part of LLSF and in the SIM part of ExpNet. The unique word cut ratios are much higher than the word occurrence cut ratios at all thresholds, except the standard cut, because the method we use for stop word identification tends to choose relatively uncommon words in the stoplist. Consequently, the time savings in SVD are higher than the time savings in SIM (except with the standard cut). Nevertheless, the computational efficiency improvement is significant in both cases. As shown in Table 1, for example, at the stop word threshold of 0.4, the unique word cut ratio was 82% and the word occurrence cut ratio was 33%; the observed time savings was 95% in SVD of LLSF (3.5 CPU minutes for the 82% unique word cutoff versus 71 CPU minutes for not using word cut), and 43% in SIM of ExpNet (1.3 msec per document for the 33% word occurrence cutoff versus 2.3 msec for not using word cut).

Table 1 shows the results of LLSF, ExpNet and WBM on different data sets at selected word cut thresholds. We select the most aggressive word cuts with some precision gain, or at worst a relatively insignificant precision loss, compared to not using word removal. For comparison, we also show in this table the results when using the standard word cuts (STD-CUT). It is clear in these results that significant efficiency improvements can be obtained by using aggressive word cuts instead of the standard cuts. In LLSF on SURCL, for example, the results in categorization precision are almost identical for a 82% word cut and STD-CUT; the time savings in SVD, on the other hand, was 95% for the former, and 14% for the latter. In ExpNet on CACMCL collection, as another example, a 87% word cut reduced the vocabulary of documents from 8002 distinct words to 1045 words, which resulted in a 63% time savings and a 74% memory savings in the SIM computation, with a 10% precision improvement over not using word cut.

Table 1. Results of categorization methods at different word cutting thresholds.

DATASET	METHOD	THRESHOLD	UNIQUE WORD CUT	ACCURACY (*)	CPU TIME**
S U R C L	LLSF	no-cut	-	.8723	71 min
		std-cut	6%	.8673 (-1%)	61 min (-14%)
		strength=0.4	82%	.8695 (-0%)	3.5 min (-95%)
	ExpNet	no-cut	-	.8859	2.3 msec
		std-cut	6%	.8830 (-0%)	1.7 msec (-24%)
		strength=0.4	82% (reduced 1244 words to 230)	.8679 (-2%)	1.3 msec (-43%)
WBM	std-cut	6%	.6135	12 msec	
	strength=0.4	82%	.6134	11 msec	
M E D I C L	LLSF	no-cut	-	.3466	82 min
		std-cut	3%	.3483 (+0%)	79 min (-5%)
		strength=0.0	77%	.3139 (-9%)	30 min (-63%)
	ExpNet	no-cut	-	.3352	216 msec
		std-cut	3%	.3454 (+3%)	121 msec (-44%)
		strength=0.3	92% (reduced 16187 words to 1271)	.3421 (+2%)	57 msec (-74%)
WBM	std-cut	3%	.1629	373 msec	
	strength=0.1	82%	.1356	343 msec	
G E N C L	LLSF	no-cut	-	.4081	123 min
		std-cut	2%	.4079 (-0%)	117 min (-5%)
		strength=0.0	44%	.3889 (-5%)	79 min (-36%)
	ExpNet	no-cut	-	.3985	168 msec
		std-cut	2%	.3989 (+0%)	158 msec (-5%)
		strength=0.4	92% (reduced 7654 words to 622)	.3977 (-0%)	70 msec (-58%)
WBM	std-cut	2%	.1791	261 msec	
	strength=0.1	63%	.1763	249 msec	
C A C M C L	LLSF	no-cut	-	.3265	55 min
		std-cut	5%	.3302 (+1%)	52 min (-6%)
		strength=0.0	54%	.3180 (-3%)	26 min (-53%)
	ExpNet	no-cut	-	.3065	245 msec
		std-cut	5%	.3295 (+8%)	167 msec (-32%)
		strength=0.2	87% (reduced 8002 words to 1045)	.3358 (+10%)	90 msec (-63%)
WBM	std-cut	5%	.2236	116 msec	
	strength=0.2	87%	.2274	90 msec	

* : the accuracy improvements compared to not using word cut are shown in parentheses;

** : the SVD time (in the training phase) is shown for LLSF; the SIM time per testing document is shown for ExpNet; the total time divided by the number of testing documents is shown for WBM.

Using STD-CUT, on the other hand, the precision improvement was less (8%), and the time savings and memory savings were 32% and 43%, respectively.

The significant efficiency improvement in the LLSF computation implies the potential use of much larger training sets without requiring faster or larger computers, and this will presumably improve the statistical reliability of the LLSF solutions. The significant efficiency improvement in ExpNet means a much faster on-line response of this system each time a free text is given, and such an improvement can be practically substantial when ExpNet is used as an interactive categorization tool for humans. Both LLSF and ExpNet are used at the Mayo Clinic, for example, as interactive tools for human coders to categorize diagnoses and surgeries described in patient records. We can only use about 50,000 training texts in LLSF due to the computational limitations of our current system (a SPARCstation 10), while we use a much larger training set of about 235,000 texts in ExpNet because its computation is much less intensive than LLSF. As a result, ExpNet has a much higher categorization precision on average than LLSF (recall

that LLSF and ExpNet were almost equally effective in our experiments when using the same training data), and the high precision leads to fewer errors in human coding when using ExpNet for category ranking (Chute, 1994). However, the on-line response of ExpNet is about 5 seconds per text, while LLSF is about 1 second per text. This makes many coders prefer to use LLSF rather than ExpNet for a better productivity in terms of the number of coded diagnoses per day, with a cost of a higher error rate in their results. This study indicates that the performance of both systems can be improved when applying aggressive word removal to free texts. That is, we can have much larger training sets and therefore improved precision in LLSF, and we can obtain a much faster real-time response in ExpNet and therefore improved user acceptability.

6 Summary

It is evident from this study that words in natural language texts are highly redundant from a categorization point of view. The redundancy of words can be identified using corpus statistics about related documents. Domain specific stoplists obtained in this manner allow us to remove much larger numbers of words than using a conventional collection-independent list of stop words, and thereby to significantly improve the computational efficiency without sacrificing categorization effectiveness. This has a practical impact on automatic or semi-automatic text categorization in large databases, especially for statistical categorization methods.

Acknowledgement

We would like to thank the Section of Medical Information Resources, Mayo Clinic, for supporting the research and for providing testing resources for the study. This work is supported in part by NIH Research Grant LM-05416 to Mayo Clinic, and National Library of Medicine Training Grant LM-07041 in Medical Informatics to the University of Minnesota.

References

- ACM Guide to Computing Literature* (1984) Baltimore, MD: Association for Computing Machinery, 657-658.
- Apte C, Damerau F, Weiss SM. (1994) Automated Learning of Decision Rules for Text Categorization. *ACM Transactions on Information Systems: Special Issue on Text Categorization*, 278-295.
- Buckley C, Salton G, Allan J. (1993) Automatic Retrieval With Locality Information Using SMART. In: DK Harman, Ed. *The First Text REtrieval Conference (TREC-1)* 59-65.
- Chute C, Yang Y, Buntrock J. (1994) An evaluation of computer-assisted clinical classification algorithms. *18th Ann Symp Comp Applic Med Care (SCAMC 94)* JAMIA 1994;18(Symp.Suppl):162-6.
- Creedy RH, Masand BM, Smith SJ, Waltz DL (1992). Trading MIPS and memory for knowledge engineering: classifying census returns on the Connection Machine. *Comm. ACM*, 35, 48-63.
- Dongarra JJ, Moler CB, Bunch JR, Stewart GW. (1979) *LINPACK Users' Guide*. , Philadelphia, PA, SIAM.
- Evans JT (1992), Bibliographic retrieval - an essential skill for surgeons. *Current Surgery*, 49(1):46-7.
- Fox EA. (Ed.) (1990) *Virginia Disc One*. Virginia Polytechnic Institute and State University, Nimbus Records.
- Fuhr N, Hartmann S, Lustig G, et al. (1991) AIR/X - a rule-based multistage indexing systems for large subject fields. *Proceedings of the RIAO'91*, 606-623.
- Golub GH, Van Loan CE. (1989) *Matrix Computations, 2nd Edition*. Baltimore, MD, The Johns Hopkins University Press.
- Haynes R, McKibbin K, Walker C, Ryan N, Fitzgerald D, Ramsden M. (1990) Online access to MEDLINE in clinical settings. *Ann. Int. Med.* 112, 78-84.
- Hersh WR, Hickam DH, Leone TJ. (1992) Words, concepts, or both: optimal indexing units for automated information retrieval. *Proc 16th Ann Symp Comp Applic Med Care (SCAMC 92)*, 16, 644-648
- Lindberg D, Humphreys B. (1990) The UMLS knowledge sources: tools for building better user interfaces. *Proc 14th Ann Symp Comp Applic Med Care (SCAMC 90)* 14:121-125.
- Masand B., Linoff G., Waltz D. (1992) Classifying News Stories using Memory Based Reasoning. *15th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval*, 59-64.
- Medical Subject Headings (MeSH)*. (1993) Bethesda, MD: National Library of Medicine.

- Riloff E and Lehnert W. (1994) Information Extraction as a Basis for High-Precision Text Classification *ACM Transactions on Information Systems: Special Issue on Text Categorization*, 296-333.
- Salton G. (1989) *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, Pennsylvania.
- Salton G. (1991) Developments in Automatic Text Retrieval. *Science*, 253, 974-980.
- Tzeras K, Hartmann S. (1993) Automatic indexing based on Bayesian inference networks. *Proc 16th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval*, 22-34.
- van Rijsbergen CJ. (1979) *Information Retrieval, 2nd ed*. Butterworths, London, England.
- Wilbur WJ, Sirotkin K. (1992) The automatic identification of stop words. *J Information Science*, 18, 45-55.
- Yang Y, Chute CG. (1992) A Linear Least Squares Fit mapping method for information retrieval from natural language texts. *Proc 14th International Conference on Computational Linguistics (COLING 92)*, 447-453.
- Yang Y, Chute CG. (1993, July) An application of Least Squares Fit Mapping to text information retrieval. *Proc 16th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 93)*, 281-290.
- Yang Y, Chute CG. (1993, November) Words or Concepts: the Features of Indexing Units and their Optimal Use in Information Retrieval. *Proc 17th Ann Symp Comp Applic Med Care (SCAMC 93)*, 685-689.
- Yang Y, Chute CG. (1994) An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems: Special Issue on Text Categorization*, 252-277.
- Yang Y. (1994) Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval. *17th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 94)*, 11-21.
- Yang Y. (1995, SIGIR) Noise Reduction in a Statistical Approach to Text Categorization. Submitted to *18th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 95)*.
- Yang Y. (1995, IJCAI) Problem Decomposition in a Statistical Approach to Text Categorization. Submitted to *International Joint Conference on Artificial Intelligence (IJCAI 95)*.