Automated Learning of Decision Rules for Text Categorization

CHIDANAND APTÉ and FRED DAMERAU
IBM T. J. Watson Research Center

SHOLOM M. WEISS
Rutgers University

We describe the results of extensive experiments using optimized rule-based induction methods on large document collections. The goal of these methods is to discover automatically classification patterns that can be used for general document categorization or personalized filtering of free text. Previous reports indicate that human-engineered rule-based systems, requiring many man-years of developmental efforts, have been successfully built to "read" documents and assign topics to them. We show that machine-generated decision rules appear comparable to human performance, while using the identical rule-based representation. In comparison with other machine-learning techniques, results on a key benchmark from the Reuters collection show a large gain in performance, from a previously reported 67% recall/precision breakeven point to 80.5%. In the context of a very high-dimensional feature space, several methodological alternatives are examined, including universal versus local dictionaries, and binary versus frequency-related features.

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*indexing methods;* I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*representations* (procedural and rule-based); I.2.6 [**Artifical Intelligence**]: Learning—*induction*

General Terms: Experimentation, Measurement, Performance

1. INTRODUCTION

Assigning classifications to documents is essential to the efficient management and retrieval of knowledge. Document classifications are typically assigned by humans who read the documents and are knowledgeable in the subject matter. In many large organizations, huge volumes of textual infor-

Authors' addresses: C. Apté and F. Damerau, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY 10598; email: {apte; damerau} @watson.ibm.com; S. M. Weiss, Rutgers University, Department of Computer Science, New Brunswick, NJ 08903; email: weiss@cs.rutgers.edu.

mation are both created and examined, and some form of categorization of this textual information flow is required. The major problem in document retrieval is one of determining whether a document is relevant to the query. This determination is inherently imprecise, since experienced people can differ on their judgments with respect to the same document and query pair, even with the whole document available and a considerable range of background information on which to draw. The document and query representations available to computer programs are much less rich, and the results may be less precise. Nevertheless, the number of documents of potential interest to a human searcher far exceeds what one could hope to read. One way that has been used to limit a search to relevant topics is to assign one or more subject codes from a predetermined list to each document added to the storage system. There are a large number of such classification systems in common use for document collections.

Assigning subject classification codes manually to documents is time consuming and expensive. Human-engineered rule-based models for assigning subject codes, while relatively effective, are also very expensive in the time and effort required for their development and continued support. This article presents results on experiments to derive the assignment rules automatically from samples of the text to be classified. In many carefully organized text storage and retrieval systems, texts are classified with one or more codes chosen from a classification system. Examples include the NTIS (National Technical Information Service) documents from the U.S. government, news services like UPI and Reuters, and publications like the *ACM Computing Reviews*. Recent work has shown that in certain environments, knowledge-based systems can do code assignment quickly and accurately [Hayes and Weinstein 1991; Hayes et al. 1990]. Machine-learning methods provide an interesting alternative for automating the rule construction process.

Effective machine-generated solutions would obviously increase efficiency and productivity. A computer can process information much faster than humans. With the explosion of electronically stored text, efficiency is of increasing importance. Beyond the immediate efficiency gains, however, is the great promise of machines that appear to "read," machines that examine free text and make correct decisions. These same techniques that make correct general decisions for text categorization can then be adapted to individual tastes, examining great volumes of text and filtering these documents to suit personal interests [Sheth and Maes 1993]. In this article, we claim that such techniques are currently feasible, that they are capable of processing huge numbers of documents in reasonable times, and that high performance is achieveable when high-quality sample data are available.

A well-known example of an expert system for this task is the CONSTRUE system [Hayes et al. 1990] used by the Reuters news service. This is a rule-based expert system using manually constructed rules to assign subject categories to news stories, with a reported recall and precision of over 90% on 750 test cases [Hayes and Weinstein 1991]. While these are exceptionally good results, the test set seems to have been relatively sparse when compared to the number of possible topics. An example of a machine-learning system for the same task is a system based on Memory-Based Reasoning [Masand et al. 1992], which employs nearest-neighbor-style classification and has a reported accuracy in the range of 70–80% on Dow Jones news stories.

In considering the problem of categorizing documents, the rule-based approach has considerable appeal. While weighted solutions such as the linear probabilistic models used in Lewis [1992b] or nearest-neighbor methods may also prove reasonable, the methods they employ are not explicitly interpretable. Since human-engineered systems have been successfully constructed using rule-based solutions, it would be most useful to continue with a model that is compatible with human-expressed knowledge. Because of the parsimonious and interpretable nature of decision rules, we can readily augment our knowledge or verify the rules by examining precategorized documents. In the remainder of this article, we describe our approach to automating the task of generating text categorization models.

2. INDUCING RULE-BASED CATEGORIZATION MODELS

Machine-learning systems solve problems by examining samples described in terms of measurements or features. For the application of machine-learning methods, the samples of documents must be transformed into this type of representation. For text categorization, an adaptation of a machine-learning method must consider the following main processes:

—A preprocessing step for determining the values of the features or attributes that will be used for representing the individual documents within a collection. This is essentially the *dictionary* creation process.

—A representation step for mapping each individual document into a *training sample* using the above dictionary, and associating it with a *label* that identifies its category.

—A induction step for finding patterns that distinguish categories from one another.

—An evaluation step for choosing the *best* solution, based on minimizing the classification error or cost.

The first step is to produce a list of attributes from samples of text of labeled documents, the dictionary. The attributes are single words or word phrases. Given an attribute list, sample cases can be described in terms of the words or phrases found in the documents. Each case consists of the values of the attributes for a single article, where the values could be either Boolean, e.g., indicating whether the attribute appears in the text or does not, or numerical, e.g., frequency of occurrence in the text being processed. Additionally, each case is labeled to indicate the classification(s) or topic(s) of the article it represents.

For rule induction, the objective is to find sets of decision rules that distinguish one category of text from the others. The best rule set is selected, where "best" is a rule set that is both accurate and not excessively complex. Accuracy of rule sets can be effectively measured on large numbers of independent test cases. Complexity can be measured in terms of numbers of

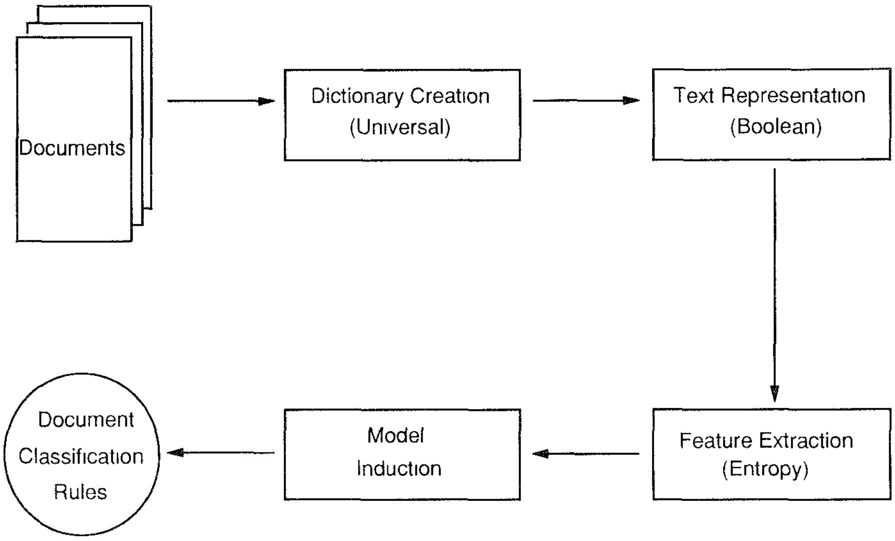
Documents
Dictionary Creation (Universal)
Text Representation (Boolean)
Feature Extraction (Entropy)
Model Induction
Document Classification Rules

Fig. 1. A typical machine-learning organization for document classification.

rules or rule components, where smaller rule sets that are reasonably close to the best accuracy are sometimes preferred to more complex rules sets with slightly greater accuracy. A typical architecture for machine learning and text categorization is illustrated in Figure 1. We will discuss now some of these issues in greater detail.

2.1 Text Representation

Document retrieval systems are supposed to choose documents that are *about* some concept of interest to the retriever. However, documents do not have concepts, but rather *words*. Words clearly do not correspond directly to concepts. Some words are used for more than one concept, e.g., "bank" as a financial instituion and "bank" as part of a river. Some concepts require more than one word for their designation, e.g., the football player "running back," and most concepts can be referenced by more than one word or phrase, e.g., "doctor" and "physician." Humans are relatively good at inferring concepts from the words of a document. To do this, they bring to bear vast knowledge of the grammar of the language and of the world at large. Very little of this knowledge is available to a computer system, in large part because we have only sketchy and incomplete methods for organizing or inferring such information automatically. Programs for parsing sentences and representing their semantic content in some formal language, e.g., first-order logic, often fail. On even simpler tasks, like deciding whether a particular use of the word "bear" is to be taken as a noun or verb, or which "bank" is being referred to, sophisticated parsing systems are far from error-free.

Despite these conceptual weaknesses, current research on text categorization supports the efficacy of the simpler schemes for text representation

[Lewis 1992b]. The conventional approach is a relatively simple selection method for creating the dictionary vocabulary. The text portion of the documents is scanned to produce a list of single words, and a list of word pairs in which neither word belongs to a defined list of stop words or is a number or part of a proper name. Any word or pair that occurs less than five times is eliminated. Words or pairs recurring only a few times are not statistically reliable indicators. Choosing the cutoff at five is arbitrary but has been used in other statistical natural language preprocessing [Church and Hanks 1989]. These two lists are similar to those identified by Lewis [1992b] for words and phrases.

Our experiments confirm reports in the literature that using only pairs as attributes gives poor results in general. In Lewis [1992a], a more sophisticated phrase selection method was used, and the same conclusion was reached. While single words alone are relatively successful, there are instances where including pairs in the dictionary can give better results. For our experiments with dictionaries derived from the full collection of documents on all topics, i.e., *universal dictionaries*, both single words and pairs were entered in the dictionaries. The single word and the pair lists were merged, sorted by frequency, and those terms in the most frequent 10,000 retained. The list was further reduced by eliminating the bottom-ranking terms if not all of the terms of that frequency were in the set of the 10,000 most frequent and by eliminating all function words. As a result, the attribute list started with approximately 10,000 attributes. While experiments could be run with this full-attribute set, most attributes are irrelevant to a given topic, and a widely used approach to prune down the attribute size is to use statistical feature selection methods. For a given categorization problem, statistical feature selection techniques, such as entropy-based techniques, are used to select those words or word pairs that are related to a given topic. In turn, this allows for the processing of greater number of sample cases.

Choosing the right attribute set to represent a document is critical to the successful induction of classification models. The attribute selection process we just described creates a dictionary of terms for a document family. Each individual document in the family can be characterized then by a set of features that are Boolean indicators denoting whether a term from the dictionary is present or absent in the document. An example of some UPI text, dated 12/01/88, and the attributes of single words and pairs that might be generated from this text are illustrated in Table I.

The range of fitting methods, classifier forms, and approaches to feature selection for the categorization problem has been extensive and varied [Biebricher et al. 1988; Fung et al, 1990; Fuhr and Pfeifer 1991; Flower and Jennings 1992; Lewis 1992a]. Since it is difficult to condense all these approaches into one formalism, we will instead utilize Figure 1 to represent a typical architecture for text categorization using a machine-learning approach. A *universal* dictionary is created for all topics, and feature selection is used to select words and phrases from this dictionary to solve a specific text categorization problem. The text of a document is represented as a set of Boolean true or false attributes. In this article, we demonstrate that a

Table I. Example of a Text Fragment and Corresponding Attribute List

.. Tokheim Corp. has announced the formation of a wholly owned environmental subsidiary in response to new U.S. Environmental Protection Agency regulations on underground storage tanks ..

announce
formation
wholly
own
wholly own
environmental
subsidiary
response
regulation
underground
storage
tank
storage tank

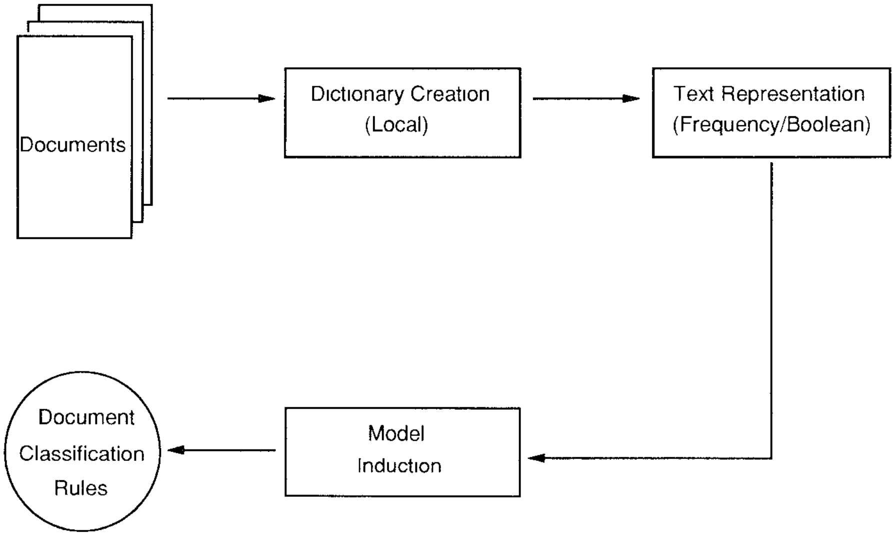
Documents
Dictionary Creation (Local)
Text Representation (Frequency/Boolean)
Model Induction
Document Classification Rules

Fig. 2. Modified machine-learning architecture for document classification

significant departure from this approach yields somewhat better results. Figure 2 illustrates the alternative strategy. Here the universal dictionary is replaced by *local* dictionaries for each classification topic. Only single words found in documents on the given topic are entered in the local dictionary. The complicated statistical feature selection step is completely eliminated, at the slight expense of generating a new dictionary for each topic, a relatively simple task when only single words and simple word-matching strategies are used. For each local dictionary, the n most frequently occurring words are used as features; where the optimal value of n is chosen based on empirical

Table II. Example of an Induced Rule Set from UPI Text for Classifying Football Stories

Rule	Class
running back	football article
kicker	football article
injure reserve	football article
award & player	football article

observations. While it can be argued that this in fact constitutes a feature selection process, we can easily observe that it is a computation-free process, as opposed to most classical feature selection methods. Additionally, instead of Boolean or more complicated frequency-related features, simple counts of the occurrence of words in a story can be used as the feature values.

Using dictionaries of single words does *not* mean that the best solution ignores phrases and combinations of words. Clearly these combinations are important to understand text. Rather, the burden is shifted from a preprocessing program that composes a dictionary to a learning program that finds the solution. Thus, these research results mostly suggest that it is very difficult to find the right combinations of words independent of the ultimate decision model. The implication of this analysis is that performance can be increased by improved learning methods. These are methods that can find higher-order relationships in the feature space, i.e., the dictionary words.

One of the main distinguishing characteristics of our approach is that we will use a rule induction model for our representation. An example of these types of rules is illustrated in Table II. This example is from one of our experiments using universal dictionaries comprised of single words and phrases. Here the problem is posed as a two-class problem, where a decision is made for classifying football stories. When none of these rules is satisfied, the decision reverts to the other default class of a nonfootball circle. Most applications of text classification involve classes that are not exclusive, and one or more of the categories can occur simultaneously. Thus we handle most problems as multiple two-class problems.

2.2 Rule Induction by Swap-1

Rule and tree induction methods have been extensively described in published works [Breiman et al. 1984; Weiss and Kulikowski 1991; Quinlan 1993]. For our document-indexing apparatus, we have used a rule induction technique called Swap-1 [Weiss and Indurkhya 1993]. Rule induction methods attempt to find a compact "covering" rule set that completely partitions the examples into their correct classes [Michalski et al. 1986; Clark and Niblett 1989]. The covering set is found by searching heuristically for a single best rule that covers cases for only one class. Having found a best conjunctive rule for a class C, the rule is added to the rule set, and the cases satisfying it are removed from further consideration. The process is repeated until no cases remain to be covered. Unlike decision tree induction programs and other rule induction methods, Swap-1 has an advantage in that it uses local optimization techniques to revise and improve dynamically its covering set. Once a covering set is found that separates the classes, the induced set of rules is further refined by either pruning or statistical techniques. Using train and test evaluation methods, the initial covering rule set is then scaled back to the most statistically accurate subset of rules.

We briefly discuss Swap-1's problem-solving approach here. Given a set of sample cases, S, where each case is composed of observed features and the correct classification, the problem is to find the best rule set, RS_{best}, such that the error rate on new cases, $Err_{true}(RS_{best})$, is minimum. Swap-1 derives solutions posed in disjunctive normal form (DNF), where each class is classified by a set of disjunctive production rules. Each term is a conjunction of tests, p_i, where p_i is a proposition formed by evaluating the truth of a binary-valued feature or by comparing a threshold to any of the values a numerical feature assumes in the samples. One such model is the decision tree, where all the implicit productions are mutually exclusive. However, a general DNF model does not require mutual exclusivity of rules. With productions that are not mutually exclusive, rules for two classes can potentially be satisfied simultaneously. Such conflicts can be resolved by inducing rules for each class according to a class priority ordering, with the last class considered a default class.

Many tree or rule inductions look ahead one attribute and try to specialize the tree or rule. To this end, a heuristic mathematical function is used, such as an entropy or gini function [Breiman et al. 1984], that evaluates and orders the relevance of attributes for making the best classification decision in a specific context (such as at the node of a decision tree). These heuristics tend to work well on many problems, and the combinatorics of finding an optimal solution make most alternative search procedures impractical.

Unlike those methods, Swap-1 looks back constantly to see whether any improvement can be made before adding a new test. The following steps are taken to form the single best rule: (a) Make the *single* best swap from among all possible rule component swaps, including deleting a component; (b) If no swap is found, add the single best component to the rule. As in Weiss et al. [1990], "best" is evaluated as predictive values, i.e., percentage of correct decisions by the rule. For equal predictive values, maximum case coverage is a secondary criterion. Swapping and component addition terminate when 100% predictive value is reached.

The process of generating the single best rule can be seen in Table III, where an example rule is generated in 7 steps. Swap-1 tries to maximize the *predictive value* of a rule, i.e., the fraction of examples classified correctly by that rule, ideally 100%. The initial rule is randomly assigned a test component, p3, which gets swapped out in favor of the single best test component, p6. Then in step 3, p1 is the single best component that can be added to the rule. However, in step 4, p6 is swapped out for p4, which is found by refining previously selected rule components. In the final step, we see that p3, which was swapped out in the first step, gets swapped in again. Thus, it can be seen that if a test is swapped out, it does not necessarily stay out, but can be

Table III. Example of Swapping Rule Components during Swap-1 Rule Construction Process

Step	Predictive Value	Rule
1	31%	p3
2	36%	p6
3	48%	p6 & pl
4	49%	p4 & pl
5	69%	p4 & pl & p2
6	80%	p4 & pl & p2 & p5
7	100%	p3 & pl & p2 & p5

added back later on if doing so improves the predictive accuracy of the current rule. The completed rule is selected as the single best rule, and the method proceeds as usual with the removal of the covered cases, and the reapplication of the single-best-rule construction procedure to the remaining cases.

Finding the optimal combination of attributes and values for even a single fixed-size rule is a complex task. However, there are other optimization problems, such as the traveling salesman problem [Lin and Kernighan 1973], where local swapping finds excellent approximate solutions.

Given a set of sample S, and a covering set RS, we can progressively weaken RS so that it becomes increasingly less complex, though decreasing in accuracy. The objective is to select rule set RS_{best} from $\{RS_1, \ldots RS_i, \ldots RS_n\}$, a collection of rule sets in decreasing order of complexity, such that RS_{best} will make the fewest errors on new cases T. In practice, the optimal solution can usually not be found because of incomplete samples and limitations on search time. It is not possible to search over all possible rule sets of complexity Cx(RS_i), where Cx is some appropriate complexity fit measure, such as the number of components in the rule set.

Several thousand independent test cases are sufficient to give highly accurate estimates of the error rate of a classifier [Highleyman 1962]. if the set $\{RS_1, \ldots RS_i, \ldots RS_n\}$ is ordered by some complexity measure Cx(RS_i), then the best one is selected by min[Err(RS_i)]. Thus, to solve this problem in practice, a method must induce and order $\{RS_i\}$ by Cx(RS_i) and estimate each rule set's error rate, Err(RS_i). A rule set's error is defined as the fraction of misclassified cases to the total classified cases as a result of applying the rule. Pruning methods adapted to rule induction can be used to prune a set and form $\{RS_i\}$. Let the rule set RS_1 be the covering rule set. Each subsequent RS_{i+1} can be found by pruning RS_i at its weakest link. As in Quinlan [1987], a rule set can be pruned by deleting single rules or single components. The application of a form of pruning known as weakest-link pruning results in an ordered series of decreasing-complexity rule sets $\{RS_i\}$, as illustrated in Table IV. The complexity of RS_i can be measured in terms of Size(RS_i).

Table IV. Example of Swap-1 Rule Induction Process Summary Table

$RuleSet_i$	# Rules	# Components	$Error_{apparent}$	$Error_{test}$
1	11	18	.0000	.1074
2	10	15	.0083	.0909
3	9	13	.0165	.0909
4*	6	7	.0661	.0744
5	6	6	.0826	.1322
6	4	4	.1322	.1322
7	3	3	.2975	.2975
8	2	2	.5372	.5620
9	1	1	.6529	.6529

The net result of this process is an error rate estimate for varying complexity rule sets. A typical result is illustrated in Table IV. For each rule set RS_i, Table IV lists the number of rules, the number of rule components, the apparent error rate on the training cases, and the error rate on independent test cases. In this example, the best solution is rule set 4, with 6 rules and 7 components, having an observed true error rate of 0.0744.

Although Swap-1 uses a criterion of minimum error for selecting the best rule, the computation of the error measure can be adjusted to force Swap-1 to select rule sets that may cover a higher number of correct cases (true positives), at the expense of covering some incorrect cases (false positives). This is done using the standard [Breiman et al. 1984] approach of substituting costs for errors to vary the true positives and false positives. For a cost of one, each false negative (the correct cases missed by a rule set) is counted as one error, but for a cost of two, each false negative is counted as two errors. A cost of one is equivalent to the usual minimum-error criterion. The effect of increasing the cost of false negatives is to increase the true positives, at the expense of increased false positives.

For the document classification application, Swap-1 induces rules that represent patterns, i.e., combinations of attributes, that determine the most likely class for an article. Applying Swap-1 to a training set of cases results in a set of rules and in the associated error rates on the training as well as test samples. The results for applying the rule set of Table II are illustrated in Table V. More in-depth discussions of the Swap-1 algorithm appear in Weiss and Indurkhya [1993].

The rule induction search space is along three major dimensions: (a) the number of documents in a document database, (b) the size of the dictionary, and (c) the number of classes for which classification models have to be learned. For some applications it may be possible to have access to hundreds of thousands of documents for training purposes. Random sampling will be effective in extracting a representative subset for the training cycle. Because

Table V. Example of Observed Error Rates for UPI Football Rule Set

	TRAINING CASES	
	Football	Not Football
Football	151	10
Not Football	0	1081

	TEST CASES	
	Football	Not Football
Football	135	26
Not Football	12	1069

the classes are not mutually exclusive, we formulate the training problem as a series of dichotomous classification induction problems.[1]

The more serious dimensionality problem lies with the dictionary size, which can be in the tens of thousands. Clearly, very large numbers of features pose a computational problem to any learning system. Conventional feature selection algorithms based on the information entropy metric, analogous to those used in decision tree construction [Breiman et al. 1984; Weiss and Kulikowski 1991; Quinlan 1993], can be used to prune down the search space. Typically, using such an approach can reduce the feature set to a small subset of the original universal dictionary. The local dictionary approach adopted by us is substantially faster, eliminating a major step from the overall classification process. More important, it reduces dimensionality severely. As the number of topics grows, a universal dictionary with even 10,000 words will be inadequate to handle low-prevalence topics. Increasing the size of the universal dictionary will increase dimensionality problems. We have observed that the local dictionary approach is both faster and more accurate when compared to using classical feature selection from a universal dictionary.

3. RESULTS WITH REUTERS NEWSWIRES

To develop our text categorization methods, we have run experiments on a number of very large document collections, including scientific abstracts originating from the National Technical Information Service, library catalog records representing the holdings of the IBM libraries, a 1988 sample of the UPI newswire, and a 1987 sample of the Reuters newswire, properly identified as Reuters-22173, but hereafter referred to as "Reuters."[2]

[1] Methods that can handle nonmutually exclusive classes simultaneously, such as neural nets, are likely to continue to use the dichotomous representation. Otherwise, the problems of dictionary dimensionality would be quite severe because the effectiveness of feature selection would be substantially diminished with large numbers of classes.

[2] The latter was obtained by anonymous ftp from /pub/doc/reuters1 on ftp.cs.umass.edu. Free distribution for research purposes has been granted by Reuters and Carnegie Group. Arrangements for access were made by David Lewis.

To provide an objective basis for comparison of our results with others, particularly Lewis [1992a; 1992b], we made a detailed number of runs using the Reuters data. There are 21,450 news stories from the year 1987. All stories beyond April 7 are used as independent test cases, and the remaining data were the training cases. The data consist of 14,704 training cases and 6746 test cases. There are 135 topics of interest, with 93 of these topics occurring more than once in the training data. We chose to experiment with these 93 topics. Our error measures, however, take into account the remaining topics (with one or fewer occurrences in the training data) since the cases associated with these topics are always present in the test data.

Of the original newswires, there are 7133 stories with "empty" topic assignments. We chose to ignore these stories, since we can neither learn from them nor test on them. As a result, the raw data that we worked with had 10,645 training cases and 3672 test cases. We derived our own dictionaries and attributes from the raw document training data and applied rule induction machine-learning methods (Swap-1). For each experiment for a given topic, a random subset, corresponding to 33% of the training data, was reserved for error estimation. Each of the recursively pruned rule sets was evaluated on these randomly selected cases to help select the best rule set. Estimates on these cases were generally within 2% of the performance of the selected rule sets on the 3672 independent test cases from after April 7.

Dictionaries were created two different ways. First, the simpler approach used the local-dictionary process, where the 150 most frequent words for the given topic were generated. We experimented with the cutoff point, evaluating cutoffs both below (50) and above (200) this threshold. The results suggested to us that 150 corresponded approximately to a local minimum, in terms of the accuracy and the performance of the induced rule sets. A brief universal list of 427 stop words was maintained, and these words were removed from the most frequent 150 words. Thus the actual number of features that were used for learning the categorization models varied for each of the 93 topics, in the range of 80–100. The local dictionaries were created using a fast algorithm that used a simple submatch strategy (without a stemmer) to pick up all the unique single words encountered in documents belonging to a topic.

The second approach was to create a universal dictionary by examining all documents in the training set. Depending on the topic, a variable number of features were derived by an entropy-based feature selection model, as in Breiman et al. [1984]. From a universal dictionary of approximately 10,000 features, the number of features selected for each category ranged between 30 and 200. The universal dictionary was created using a match strategy that employed a stemmer to pick up all the unique stems encountered in the entire training set across all topics. The same stop list that was used for the local dictionary was used here, although here it was a one-time application to filter out the stop words from the universal dictionary prior to its application.

For the text representation, we experimented with both frequency and Boolean features. The Boolean features indicate merely whether an entry in the dictionary is present in a document or not, while the frequency feature

Table VI. Induced Rule Set and Performance on Test Data for Reuters "Wheat" Category

wheat & farm ⟶ wheat
wheat & commodity ⟶ wheat
bushels & export ⟶ wheat
wheat & agriculture ⟶ wheat
wheat & tonnes ⟶ wheat
wheat & winter & ¬soft ⟶ wheat

	Test Cases	
	wheat	not wheat
wheat	73	8
not wheat	14	3577

indicates the number of occurrences of a dictionary entry in a given document. No experiments were performed with more complicated frequency-related measures.

Performance is measured by *recall* and *precision*. "Recall" is the percentage of total documents for the given topic that are correctly classified. "Precision" is the percentage of predicted documents for the given topic that are correctly classified. Because the document topics are not mutually exclusive, document classification problems are usually analyzed as a series of dichotomous classification problems, i.e., the given topic vs. not that topic. For example, Table VI illustrates the rule set that was induced for the *wheat* category for a local dictionary with a Boolean representation for the text.[3] Also included in the figure is the performance table of this rule set on the Reuters post-April 7, 1987, test data. Given the rule evaluation table as in Table VI, one can measure performance using a wide variety of metrics, based on error rates or costs. For the purpose of this article, we have chosen the *microaverage* measure, as used in Lewis and Ringuette [1994]. To evaluate overall performance across the entire set of topics, the results are microaveraged, i.e., the performance tables for each of the topics, such as in Table VI, are added, and the overall recall and precision are computed. The point at which recall equals precision is the *breakeven* point; it can be used as a single summarizing measure for comparison of results.

The breakeven point for each of the four combinations of dictionaries and features is illustrated in Table VII. Additionally, the previously reported breakeven points of 67% for decision tress [Lewis and Ringuette 1994] and 65% for a probabilistic method [Lewis 1992a] are listed. If all text is treated uniformly, the breakeven point for the local dictionary with frequency features is 78.9%. However, the newswire stories contain a one-line headline that can provide additional clues to the topic. If the words occurring in the

[3] In this example, the cost of false negatives was set as three times false positive.

Table VII. Recall/Precision Breakeven Points for Various Classification Methods on Reuters Data

Learning Method	Dictionary	Text Representation	Performance Breakeven (%)
Optimized Rule Induction	Local	Frequency + Headlines	80.5
		Frequency	78.9
		Boolean	78.5
	Universal	Frequency	78.0
		Boolean	75.5
Decision Tree			67.0
Probabilistic Bayes			65.0

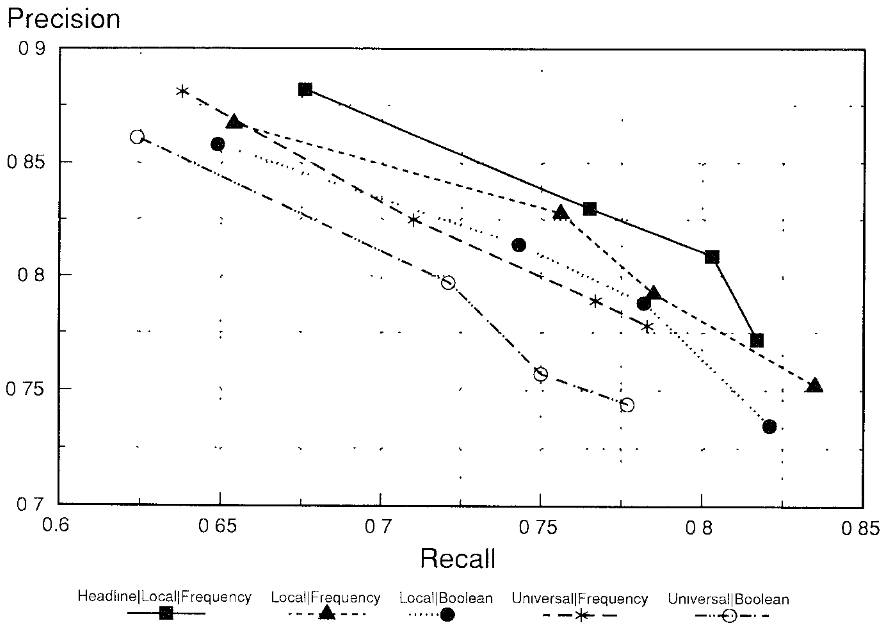

Fig. 3. Decision rule learning results for recall/precision tradeoff for Reuters data for varying text representations.

headline are given additional emphasis, by counting them twice instead of using a uniform count for words in either the headline or body of an article, then performance for the local dictionary with frequency features is increased by almost 2 percentage points, to a breakeven point of 80.5%.

A breakeven point is a combined summary measure, but for text categorization both recall and precision may be of interest. Figure 3 illustrates the overall performance of the rule induction variations. Figure 4 compares our results with previously reported results. To determine a breakeven point (1) several learning experiments must be performed and (2) some parameter must be varied to elicit the tradeoff of recall and precision. The appropriate technique may vary with the learning method. For rule induction, the tradi-

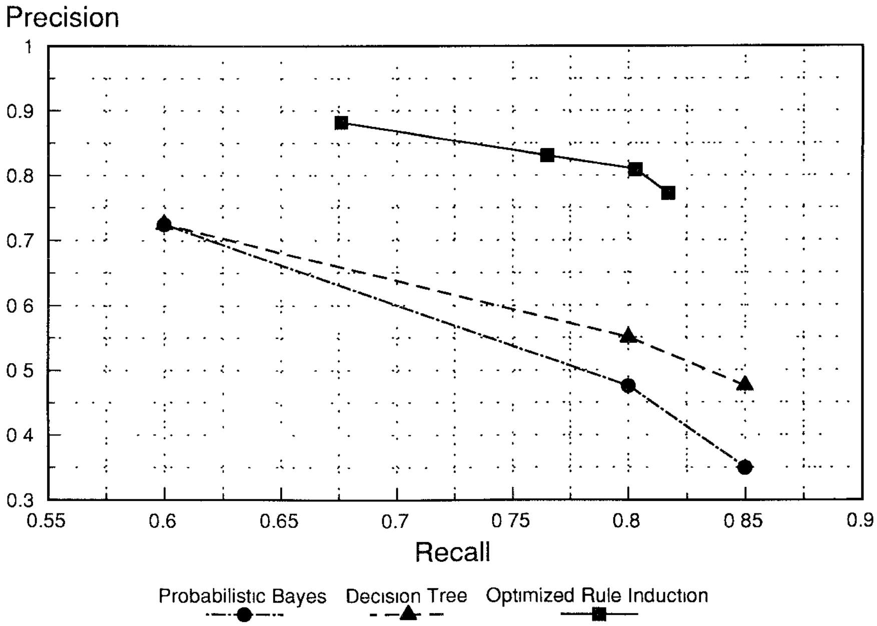

Fig. 4. Comparison of other reported results for Reuters data with best decision rule learning result.

tional goal is to minimize the number of errors, which may not be the breakeven point. We varied the cost setting in Swap-1 to experiment with recall/precision tradeoffs. In our experiments with the Reuters data, the breakeven point was achieved near a cost setting of three.

4. DISCUSSION

When compared to previous results on the Reuters data, the new results appear to be significantly better. A single breakeven measure is used for comparison, but this measure summarizes the results of dozens of relatively independent experiments on tens of thousands of test cases. Thus, we can be assured that the results are a highly significant improvement over previously reported results for the same data.[4]

Figure 3 suggests that the use of local dictionaries and frequency information was effective and improved the results of our rule induction methods.

[4] It has been pointed out to us that there is a difference in our experimental procedures from those reported in [Lewis and Ringuette 1994]. While we chose to ignore the unlabeled wires in the Reuters collection, they were included in Lewis and Ringuette's study. This variance tends to strengthen our results somewhat, while weakening those of the other study. Further investigation will be needed to compare results under exactly identical procedures for training and test data creation.

While previous experience has shown that the optimization techniques of the Swap-1 rule induction method can often improve results substantially over competitive methods, such as decision tress, text classification has a number of characteristics that make optimized rule induction particularly suitable. The optimization techniques that are employed are quite strong in finding feature dependencies. In terms of text classification, this means that given single-word dictionaries, Swap-1 can find the key word combinations that separate topics. Unlike many applications, here the class label that we consider as "truth" is humanly assigned by a reader or the author of the document. Those methods that emphasize models that are most compatible with human reasoning should have a distinct advantage. We already know that human-engineered systems, using the identical representation of production rules, can be successful in text classification. We have demonstrated that these same rule-based systems for text classification can be automatically generated from samples with very comparable performance measures.

Is it possible that we can hope for results even better than the 80.5% breakeven that we obtained from the current set of experiments? A number of possibilities remain to be explored. While we have used the obvious frequency measures, other measures can readily be proposed. Overall, the local dictionary did better and was faster, but we have yet to examine whether there were situations where the universal dictionary performed consistently better. There are hints that this is the case for high-prevalence topics. There are also potential improvements that could be made to the feature selection process of the universal dictionary. We relied on the very simplest of dictionaries and text-matching strategies. It is possible that a more sophisticated matching strategy may yield an improved margin of performance.

A limiting factor in the evaluation of results is that one does not know the true upper bound on performance. The natural expectation is that 100% correct performance can be achieved. From a machine-learning perspective in this application, we know that such performance is not achievable because many labels are not correct. With over 500 possible topics that can be assigned to stories, it is quite likely that a reader will miss the assignment of some topics or will be inconsistent in the assignment of topics. We observed a few topic assignment mistakes in the Reuters collection.[5] Even with a careful reading, human language is not precise enough to allow full agreement by readers of all stories. Machine-learning programs can operate in this uncertain environment and find patterns that separate the populations with some degree of error. It may very well be the case that a blinded comparison by independent observers of the automated approach versus the human-assigned approach will demonstrate that the machine does as well or better.[6]

[5] Although the collection we examined had human-assigned topics, they are now assıgned wıth the aid of a knowledge-based system.

[6] A computer can evaluate thousands of cases in mere seconds. A serious large-scale study would requırc a huge expendıture of human time to valıdate this hypothesıs.

We can look at the recall figures as a measure of overlap or consistency with the human indexers of the documents. There have been studies of the consistency of human indexers with each other, although not in this context. In a survey of this work, Saracevic [1991] reported consistency values ranging from 10% to 80%. In studies comparing indexing of inadvertently duplicated documents in *Information Science Abstracts* and *MEDLINE*, consistency for central concepts or main headings, which are roughly analogous to our subject codes, was 52% to 61%. Even at high levels of precision, our recall figures exceed these percentages. While not definitive, these results suggest that machine-learning methods may be comparable to human performance.

We have also examined a variety of other document collections, including UPI newswires, NTIS technical abstracts, and Library of Congress card catalogs. The strong results that we obtained with the Reuters newswires are consistent with the result obtained with the UPI data. We also got very favorable results with the Library of Congress data, although our experiments were strictly done with holdings of the IBM library system. Given that this collection is inherently skewed toward a technical content, we need to experiment with a more general collection of card catalog information before any conclusions can be drawn. With the NTIS data, we obtained results that did not hold up as favorably as we had expected; detailed postinduction analyses suggest that the NTIS abstracts are frequently prone to erroneous classifications by humans [Apté et al. 1993].

From these experiments, it appears that optimized rule induction is more than competitive with other machine-learning techniques [Masand et al 1992; Lewis and Ringuette 1994; Lewis 1992a] for document classification, and very close behind human-engineered systems [Hayes and Weinstein 1991]. Such conclusions can be supported only by rigorous and exacting comparisons. Given the very large volumes of data, and the sometime proprietary nature of documents, it is not surprising that few, if any, comparisons have been reported in the literature. The 1987 Reuters stories have been widely circulated recently and should prove to be an important benchmark for objective comparisons.

Machine-induced rule-based models permit efficient analytical investigations, since rule sets can be inspected and modified easily, either by human or machine. This process has been found to be useful when attempting to understand why documents get misclassified, and allows experiments with fine tuning of the induced models. Often, this inspection detects erroneous classifications in the existing document database. For example, the NTIS document family was discovered to be widely populated with documents that had incorrect human assignments of topics.

The explosive growth of electronic documents has been accompanied by an expansion in availability of computing. It is unlikely that such information can be managed without extensive assistance by machine. Some processes once thought of as requiring comprehension and understanding may prove to be weaker than a machine's compute-intensive methods for discovering classification patterns. Such machine-learning and discovery systems may be combined with human-developed systems for document classification. These, in turn, could be conceivably coupled as knowledge filters for tools like newswire alerts and information feeds to provide superior information retrieval services to the end user.

REFERENCES

Apté, C., Damerau, F., and Weiss, S. 1993. Knowledge discovery for document classification. In *Working Notes of the AAAI 1993 Workshop on Knowledge Discovery in Databases (KDD-93)*. AAAI, Menlo Park, Calif., 326–336.

Biebricher, P. Fuhr, N., and Lustig, G. 1988. The automatic indexing system (AIR/PHYS)—From research to application In *ACM SIGIR' 88* ACM, New York, 333–342.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. 1984. *Classification and Regression Trees*. Wadsworth, Monterey, Calif

Church, K. W. and Hanks, P. 1989. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*. ACL, 76–83.

Clark, P. and Niblett, T. 1989. The CN2 induction algorithm *Mach Learn. 3*, 261–283

Flower, M. and Jennings, A. 1992. Domain classification of language using neural networks. In *3rd Australian Conference on Neural Networks.*

Fuhr, N. and Pfeifer, U. 1991 Combining model-oriented and description-oriented approaches for probabilistic reasoning. In *ACM SIGIR' 91*. ACM, New York, 46–56.

Fung, R., Crawford, S., and Appelbaum, L. 1990. An architecture for probabilistic concept-based information retrieval. In *ACM SIGIR' 90*. ACM, New York, 455–467.

Hayes, P and Weinstein, S 1991. Adding value to financial news by computer. In *Proceedings of the 1st International Conference on Artificial Intelligence Applications on Wall Street.* 2–8.

Hayes, P. J., Andersen, P. M., Nirenburg, I B., and Schmandt, L. M 1990. TCS: A shell for content-based text categorization. In *Proceedings of the 6th IEEE CAIA*. IEEE, Piscataway, N.J , 320–326.

Highleyman, W. 1962. The design and analysis of pattern recognition experiments. *Bell Syst Tech. J. 41*, 723–744.

Lewis, D 1992a. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, New York, 37–50.

Lewis, D. 1992b. Feature selection and feature extraction for text categorization In *Proceedings of the Speech and Natural Language Workshop.* Defense Advanced Research Projects Agency, Washington, D.C., 212–217.

Lewis, D. and Ringuette, M. 1994. A comparison of two learning algorithms for text categorization. In *Symposium on Document Analysis and Information Retrieval.* ISRI, Univ. of Nevada, Las Vegas. To be published.

Lin, S. and Kernighan, B. 1973 An efficient heuristic for the traveling salesman problem *Oper. Res. 21*, 2, 498–516.

Masand, B., Linoff, G., and Waltz, D. 1992. Classifying news stories using memory based reasoning. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, New York, 59–65.

Michalski, R., Mozetic, I., Hong, J., and Lavrac, N. 1986. The multi-purpose incremental learning system AQ15 and its testing application to three medical domains In *Proceedings of the AAAI-86.* AAAI, Menlo Park, Calif., 1041–1045.

Quinlan, J. R. 1993 *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, Calif.

Quinlan, J. 1987. Simplifying decision trees. *Int. J. Man-Mach Stud. 27*, 221–234

Saracevic, T. 1991. Individual differences in organizing, searching and retrieving information. In *Proceedings of the 54th Annual Meeting of the Society for Information Science,* Jose-Marie Griffiths, Ed. Soc for Information Science, 82–86

SHETH, B. AND MAES, P. 1993. Evolving agents for personalized information filtering. In *Proceedings of the IEEE CAIA-93*. IEEE, New York, 345–352.

WEISS, S. AND INDURKHYA, N. 1993. Optimized rule induction. *IEEE Exp. 8*, 6, 61–69.

WEISS, S. M. AND KULIKOWSKI, C. A. 1991. *Computer Systems That Learn.* Morgan Kaufmann, San Mateo, Calif.

WEISS, S., GALEN, R., AND TADEPALLI, P. 1990. Maximizing the predictive value of production rules. *Art. Intell. 45*, 47–71.

Received June 1993; revised February 1994; accepted March 1994