A Comparison of Two Learning Algorithms for Text Categorization

(Symposium on Document Analysis and IR, ISRI, Las Vegas)

David D. Lewis
Ctr. for Info. and Lang. Studies
University of Chicago
Chicago, IL 60637

Marc Ringuette
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

April, 1994

Abstract

This paper examines the use of inductive learning to categorize natural language documents into predefined content categories. Categorization of text is of increasing importance in information retrieval and natural language processing systems. Previous research on automated text categorization has mixed machine learning and knowledge engineering methods, making it difficult to draw conclusions about the performance of particular methods.

In this paper we present empirical results on the performance of a Bayesian classifier and a decision tree learning algorithm on two text categorization data sets. We find that both algorithms achieve reasonable performance and allow controlled tradeoffs between false positives and false negatives. The stepwise feature selection in the decision tree algorithm is particularly effective in dealing with the large feature sets common in text categorization. However, even this algorithm is aided by an initial prefiltering of features, confirming the results found by Almuallim and Dietterich on artificial data sets. We also demonstrate the impact of the time-varying nature of category definitions.

Function: Learning
Domain: Natural Language Text
Foundation: Statistical

1 Introduction

Text categorization—the automated assigning of natural language texts to predefined categories based on their content—is a task of increasing importance. A primary application of text categorization systems is to assign subject categories to documents to support information retrieval, or to aid human indexers in assigning such categories [BFL+88, HW90]. Text categorization components are also seeing increasing use in natural language processing systems for data extraction. Categorization may be used to filter out documents or parts of documents that are unlikely to contain extractable data, without incurring the cost of more expensive natural language processing [DLW+91, GSM91, Hob91]. They also can be used to route texts to category-specific processing mechanisms [DeJ82, SS89, JR90], and even to generate fillers for some fields [BT91, DGCN91, DR91, Lew91a].

Text categorization systems attempt to reproduce human categorization judgments. One approach to building a text categorization system is to manually assign some set of documents to categories, and then use inductive learning to automatically assign categories to future documents based on, say, the words they contain. Such an approach can save considerable human effort in building a text categorization system, particularly when replacing or aiding human indexers who have already produced a large database of categorized documents.

This paper presents empirical results on the text categorization performance of purely inductive methods. Two inductive learning algorithms are investigated, one based on Bayesian classifiers and the other on decision trees. Each algorithm is studied on two tasks: indexing financial newswire stories for document retrieval, and extracting data on terrorist incidents from highly variable text sources. We find that both algorithms achieve reasonable performance and allow tradeoffs between false positives and false negatives. We also find that problem characteristics found to be important on other inductive learning problems are crucial to text categorization as well, including feature set size and representativeness of training sets.

2 Text Categorization: Nature and Approaches

Text categorization has been applied to technical abstracts, newswire stories, electronic mail, financial telexes, survey data, and many other forms of text. Entire documents or parts of documents, such as paragraphs or sentences, can be categorized. Some representation of a text must be examined in order to make categorization decisions. Many text categorization systems treat a text as a "bag of words", ignoring the original ordering of the text and treating the presence or absence of each word as a binary feature. Other systems take advantage of multi-word phrases, positional or linguistic structure, or other information.

There have been two main approaches to the construction of text categorization systems. First, a number of systems [VS87, Har88a, HW90] have embodied approaches similar to those used in expert systems for classification or diagnosis [Cla85]. Knowledge engineers define one or more layers of intermediate conclusions between the input

```
1-APR-1987 07:01:25.19
TOPICS: gold silver    END-TOPICS

PRECIOUS METALS CLIMATE IMPROVING, SAYS MONTAGU
    LONDON, April 1 - The climate for precious metals is improving
with prices benefiting from renewed inflation fears and the switching
of funds from dollar and stock markets ... Silver prices in March
gained some 15 pct in dlr terms due to a weak dollar and silver is
felt to be fairly cheap relative to gold ... The report said the
firmness in oil prices was likely to continue in the short term ...
REUTER
```

Figure 1: Parts of a categorized story from the Reuters newswire.

evidence (words and other textual features) and the output categories and write rules for mapping from one layer to another, and for confirming or removing conclusions.

The second strategy is to use existing bodies of manually categorized text in constructing categorizers by inductive learning. A wide variety of learning approaches have been used, including Bayesian classification [Mar61], decision trees [CFAT91], factor analysis [BB63], fuzzy sets [COL83], linear regression [BFL^{+}88], and memory-based approaches [CMSW91]. Learning-based systems have been found to be cheaper and faster to build, as well as more accurate in some applications [CMSW91].

Text categorization applications nevertheless provide many challenges for machine learning. Feature sets are huge—on the order of tens of thousands of features when words are used, or even more if multi-word phrases are allowed. Natural language features exhibit a number of properties, including synonymy, ambiguity, and skewed distributions, that interfere with forming classification functions [Lew92]. Proper categorization may depend on subtle distinctions in emphasis. A human indexer assigned the story in Figure 1 to the GOLD and SILVER categories, but not to the DLR or OIL categories, as these concepts were apparently not considered central.

The bulk of text categorization research has been conducted by organizations with a pressing need for an operational text categorization system. This has meant that inductive learning, when used, has almost always been supplemented by knowledge engineering or ad hoc fixes. Little is known about the effect that characteristics of text have on inductive learning algorithms, or about what the performance of purely learning-based methods would be. Our paper attempts to remedy this lack.

3 Our Algorithms

We investigated two inductive learning algorithms. Both operate within the same model: given a training set of pre-categorized documents, and a vector of binary features for each document, they learn to categorize a test set of documents. The first, PropBayes, uses Bayes' rule to estimate the category assignment probabilities, and then assigns to a document those categories with high probabilities. The second algorithm, DT-min10, uses decision tree techniques to recursively subdivide the training examples into interesting subsets, based on an information gain metric.

Before either algorithm was used, a preliminary filtering of the features for each data set was done. All features were ranked for each category using the information gain measure [Qui86a] and the top features were given to each algorithm. This enabled us to explore variations in the size of the feature set given each algorithm, and was also a practical necessity given the size of the full feature sets (see Section 4.1).

3.1 PropBayes

We describe PropBayes only briefly, since full details have been presented elsewhere [Lew92, Lew92b]. It uses Bayes' rule to estimate $P(C_j = 1|D)$, the probability that a category C_j should be assigned to a document, based on the prior probability of a category occurring, and the conditional probabilities of particular words occuring in a document given that it belongs to a category. For tractability, the assumption is made that probabilities of word occurrences are independent of each other, though this is often not the case. A four feature classifier for the Reuters ACQ (corporate acquisitions) category is:

$\log P(C = 1|D) = -2.187 + \log(5.879 \times \mathit{acquire} + 0.840) + \log(5.398 \times \mathit{acquisition} + 0.795) + \log(2.609 \times \mathit{shares} + 0.715) + \log(5.414 \times \mathit{stake} + 0.826)$

Given accurate estimates of $P(C_j = 1|D)$, the optimal categorization strategy, under the assumption of equal costs for all errors, is to set a threshold p and assign to document D all C_j for which $P(C_j = 1|D) >= p$ [DH73]. This strategy is not necessarily optimal when there are errors in the probability estimates, due to limited samples or the violation of our independence assumptions. We have experimented with a number of alternative thresholding strategies, and found the best to be *proportional assignment*. Each category is assigned to its top scoring documents in proportion to the number of times it was assigned on the training corpus. To trade off false positives and false negatives, a proportionality constant k is used. If X% of the training documents are in category C_j, we place the kX% of the test documents with the highest estimates of $P(C_j = 1|D)$ in that category.

3.2 DT-min10

Our second approach was a decision tree learning algorithm implemented using the IND package [Bun90, Bun91]. A decision tree was constructed for each category using the recursive partitioning algorithm with information gain splitting rule. A leaf was forced when fewer than 10 examples fell at a node, and no pruning was done. (Several other tree-building methods were tried on subsets of the data, including the CART-style and Bayes Trees options of IND, but this method as approximately as good as any other, and simpler.) A probability is maintained at each leaf, rather than a binary decision; we vary the willingness of the algorithm to assign categories by changing a

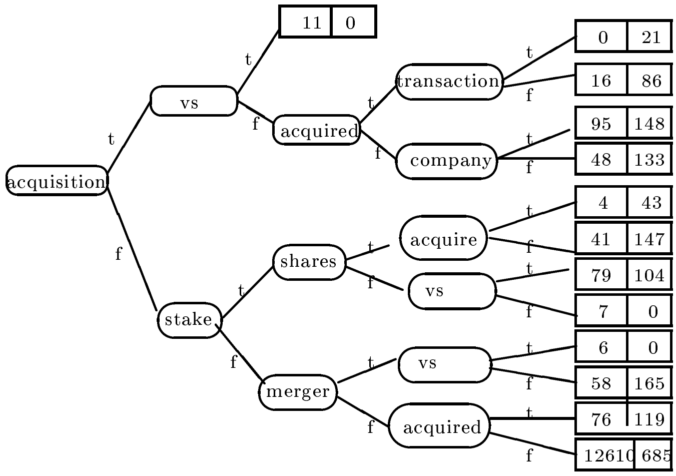


Figure 2: An example decision tree

threshold on leaf probabilities. [1]

An example tree for the ACQ category, truncated to to depth 4, is shown in Figure 2. Shown at the leaves are the number of negative and positive examples at that leaf in the training set. Most actual trees are much larger.

3.3 Comparing the Algorithms

Both PropBayes and DT-min10 use training examples to estimate conditional probabilities of category occurrence given feature occurrences. PropBayes can estimate a conditional probability of category assignment for any conjunction of feature values, but only by (problematically) assuming a product distribution. DT-min10 estimates conditional probabilities separately for each of a selected set of conjunctions of feature values. This lets DT-min10 avoid independence assumptions, but demands larger numbers of training instances. It also means that DT-min10 that can induce any discrete probability distribution over the feature space, and unlike PropBayes will converge to the optimal categorizer (for a given feature set) in the limit of an infinite well-distributed training set.

If we consider the tendencies or *bias* [Utg86] of the algorithms instead of asymptotic behavior, we see that DT-min10 most succintly represents distributions where conditional probabilities are uniform except as specifiable by conjunctions of a small number of features. PropBayes most easily represents distributions which can be decomposed into the product of a small number of probabilities conditional on single features.

[1]The actual commands given to the IND package in constructing and using the tree were `mktree -o "-s10" -p -n` for creating the tree and `tclass -St` for classifying, with the ratio of negative `utilities` for false negatives and positives being varied between `0.00091` and `256`.

4 Evaluation

We evaluated the performance of our algorithms by measuring their ability to reproduce manual category assignment on two data sets. We describe the data sets and the evaluation measures below.

4.1 Data Sets

Two data sets requiring quite different text categorization tasks were used in our experiments. These two sets were selected because results from previous evaluations using them led us to believe that the manual category assignments were particularly consistent. The first was a set of 21,450 Reuters newswire stories from the year 1987. These stories have been manually indexed using 135 financial topic categories, to support document routing and retrieval by Reuters customers. All stories dated April 7, 1987 and earlier went into a set of 14,704 training documents, and all stories from April 8, 1987 or later went into a test set of 6,746 documents.

Of the 135 categories, 112 had one or more occurrences on the training set, 94 had one or more occurrences on the test set, and 90 had one or more occurrences on both. Obviously, performance on categories for which there are no positive training instances will be low for a purely learning-based method. Each document was represented by a set of 22,791 binary features corresponding to English words that occurred in two or more training documents. Capitalization was ignored and a standard list of stop words (mostly grammatical function words) were removed.

The second data set consisted of 1,500 documents from the U.S. Foreign Broadcast Information Service (FBIS) that had previously been used in the MUC-3 evaluation of natural language processing systems. The documents are mostly translated from Spanish, and include newspaper stories, transcripts of broadcasts, communiques, and other material. Documents were represented by a set of 8,876 binary features corresponding to English words occurring in 2 or more training documents. The MUC-3 text was all capitalized. Stop words were not removed.

The MUC-3 task required extracting simulated database records ("templates") describing terrorist incidents from these texts. MUC-3 systems generated one or more templates for each test document, and their performance was measured by comparing the generated templates with manually coded templates. Eight of the template slots had a limited number of possible fillers, so a simplificiation of the MUC-3 task is to view filling these slots as text categorization. There were 88 combinations of these 8 slots and legal fillers for the slots, and each of these combinations was treated as a binary category.

We used for our categorization test set 300 documents (the 200 MUC-3 test documents plus 100 MUC-3 training documents) for which templates were encoded by the MUC-3 organizers. We used the other 1,200 MUC-3 training documents (encoded by 16 different MUC-3 sites) as our categorization training documents. It is likely that category assignments on our training set are less consistent than assignments on our test set.

Details of the Reuters [Lew92, Lew92b] and MUC-3 [Lew91a] datasets are available in other publications.

4.2 Evaluation Measures

The performance measures used were *recall* (number of categories correctly assigned divided by the total number of categories that should be assigned) and *precision* (number of categories correctly assigned divided by total number of categories assigned) [van79]. Recall is the true positive rate from signal detection. Precision is an alternative to *fallout* (the false positive rate), and has been more widely used than fallout in evaluating text categorization systems.

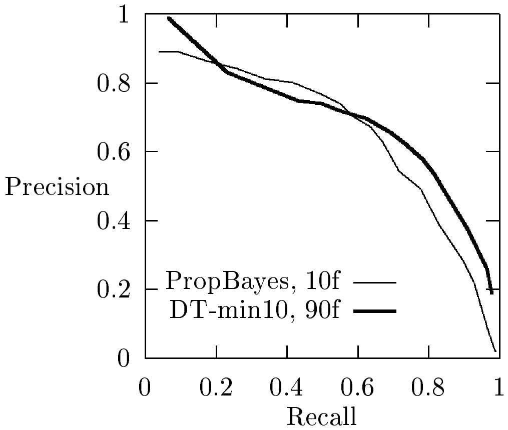
For a set of k categories and d documents a total of $n = kd$ categorization decisions are made. Given those kd decisions, several ways of computing average performance are available [Lew91b]. We used *microaveraging*, which considers all kd decisions as a single group and computes recall and precision for the group as whole.

Both of our categorization algorithms include an adjustable parameter controlling the algorithm's willingness to assign categories to documents. As the algorithm is made more willing to assign categories, recall goes up, and precision usually (though not always) goes down. The performance of the algorithm at several parameter settings can be plotted to show the tradeoffs possible between recall and precision. As a single summary figure for recall precision curves we take the *breakeven* point, i.e. the value (or the highest value) at which recall and precision are equal. Linear interpolation was used between recall precision points to get the breakeven values.

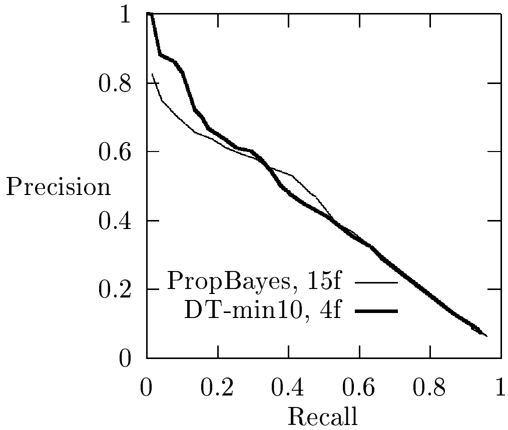
5 Results: Overall Performance

During our experiments we ran each algorithm with various-sized feature sets chosen by information gain. The best-performing variations are shown in the recall/precision curves below.

Graph 1a: performance on Reuters data



Graph 1b: performance on MUC3 data



Both algorithms performed well on the Reuters data and less well on the MUC-3 data. The breakeven points for the two algorithms, where the above curves intersect the line *recall=precision*, were 0.65 for PropBayes vs. 0.67 for DT-min10 on Reuters, and 0.48 vs. 0.46 on MUC-3.

The decision tree method performed particularly well on the Reuters task at high recall levels. At a recall level of 0.95, DT-min10 has a precision of 0.28 vs. 0.15 for PropBayes.

PropBayes does not use large number of features effectively, but neither can it achieve good precision at high recall with small numbers of features. Attempting high recall with small feature sets means that often a single feature, often unreliable, will result in a document being assigned to a category. DT-min10's stepwise feature selection enables it to use more features, but demands more training examples. This may be why DT-min10's advantage at high recall does not appear on the smaller MUC-3 data set.

Indirect comparisons with other text categorization methods indicate that our performance is competitive with that of other approaches to text categorization. For instance, the operational AIR/X system uses both rule-based and statistical techniques to achieve a microaveraged breakeven point of approximately 0.65 in indexing a physics database [FHL+91].

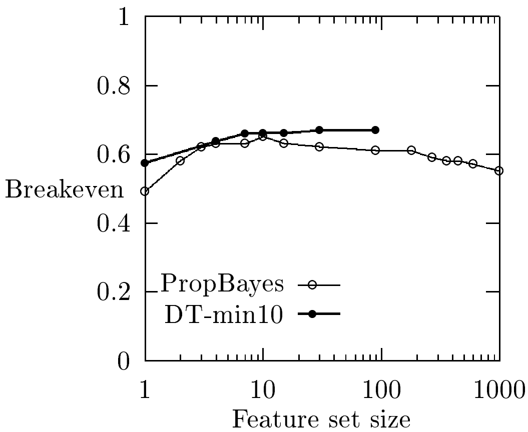
The CONSTRUE rule-based text categorization system achieves a microaveraged breakeven of around 0.90 on a different, and possibly easier, testset drawn from the Reuters data [HW90]. This level of performance, the result of a 9.5 person-year effort, is an admirable target for learning based systems to shoot for.

Comparison with published results on MUC-3 are difficult, since we simplified the complex MUC-3 task. However, in earlier experiments using the official MUC-3 testset and scoring, PropBayes achieved performance toward but within the low end of official MUC-3 scores [Lew91a].

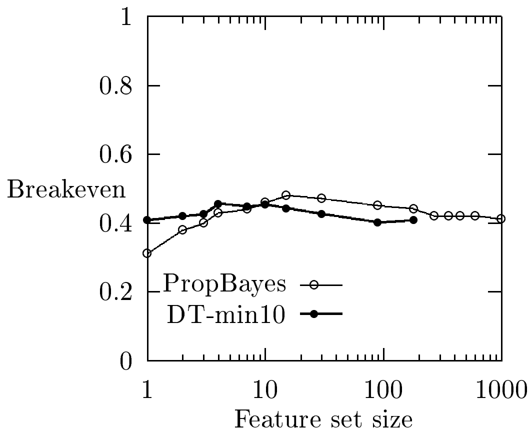
6 Feature Set Size

We experimented with varying the number of features chosen by information gain for each category to study the effect of feature set size on performance.

Graph 2a: effect of feature set size, Reuters



Graph 2b: effect of feature set size, MUC3



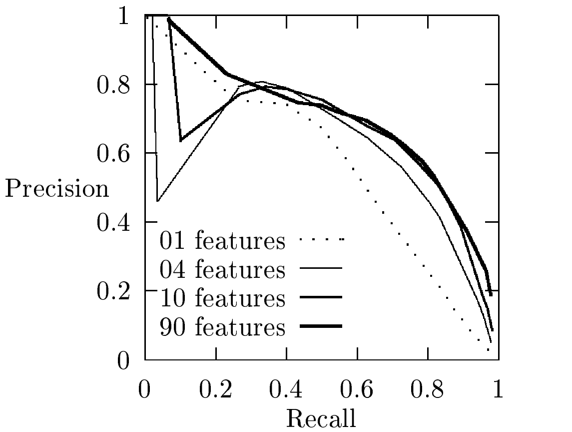
For PropBayes, performance peaks at around 10 features for the Reuters task and 15 features for the MUC-3 task. With more features performance starts to decline. A primary cause of this phenomenon, refered to as "the curse of dimensionality" [DH73], is overfitting. As an increasing number of parameters are estimated from a fixed amount of data, we induce a model of the noise as well as the true relationships in the training set. Besides overfitting, other factors such as the lower quality of the additional features and increasing violation of the independence assumptions made by PropBayes probably also play a role in peaking.

DT-min10 evaluates the quality of each feature in the context of the features above it in the tree. This provides a more accurate measure of a feature's usefulness than our initial filtering on information gain, but only when there are enough training examples satisfying the particular path through the tree for this measurement to be accurate. On the Reuters data, DT-min10 continues to improve through 90 features (Graph 2a); it peaks at 4 to 10 features on the smaller MUC-3 data set (Graph 2b).

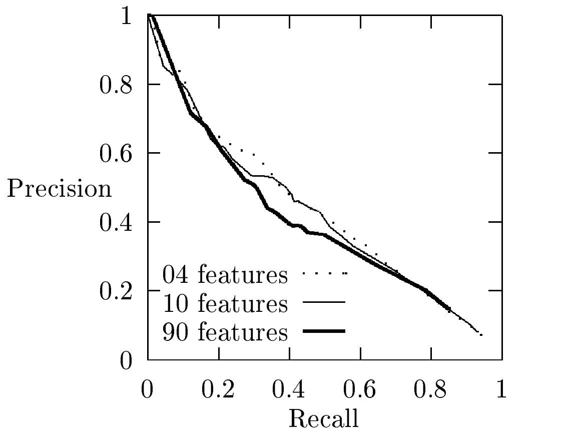
7 Recall/Precision Tradeoffs for DT-min10

Trading off recall and precision in a decision tree is worth a closer look. The following two graphs show DT-min10 curves for several interesting values of feature set size.

Graph 3a: DT-min10 feature sets: Reuters



Graph 3a: DT-min10 feature sets: MUC-3



The first thing we notice in Graph 3a is the erratic behavior on the Reuters data as recall decreases. Precision approaches an asymptote at approximately 80%, and when we try to push it farther, precision varies erratically. This results from our method of adjusting the recall/precision tradeoff for the decision tree method. In order to achieve high precision, we must set a high threshold for the probability at each leaf of the decision tree: eventually, only the subset of leaves with 100% precision on the training set is chosen. Such leaves are often based on a small number of examples, and may be in a bad neighborhood of the tree, and so may not yield the best performance on the test set. To a certain extent, this problem is inherent in the simplicity of the decision tree model, in which each estimated probability corresponds to a disjoint set of training instances. However, more sophisticated treatment of estimation from small samples might help [GC90].

The second thing we notice is that performance seems to approach an asymptote on the Reuters data as the number of features increases. This suggests we have avoided overfitting in the range of feature set sizes tested. In fact, performance at 90 features is the best of all feature set sizes at high recall levels.

On the MUC-3 data the recall/precision tradeoff is smooth; however, overfitting is clearly a problem. Performance decreases from 4 to 10 to 90 features in the middle part of the graph.

8 Temporal Structure in Text Categorization

Microaveraged performance, particularly at low recall, is dominated by behavior of the system on a few categories with many positive instances on the training set. We therefore decided to specifically examine performance on the most frequent category Reuters category, EARN (corporate earnings). Examining the decision tree built by DT-min10 showed that the conjunction of abbreviations *vs* **AND** *cts* was an extremely powerful feature on the training set, taking on the value **True** for 36 negative examples and 1720 positive examples, but was only moderately good on the test set with 251 negative examples and 862 positive ones.

In looking at the test documents, we realized that *vs* **AND** *cts* stories were almost always either stock dividend announcements (not a Reuters category) or more full earnings reports (EARN). It also appeared that the stories we were getting wrong were clustered temporally. This was confirmed strikingly when we grouped stories by month. Only 61.3% of the 6746 test stories appeared on the newswire in June or October. However, *all* of the 251 test stories satisfying *vs* **AND** *cts* and not categorized by EARN fell in these two months. Essentially all of these were dividend stories, and unfortunately dividend stories had been severely underrepresented in the months from which our training set was drawn.

It seems likely that as we look at the behavior of individual categories in more detail we will find other temporal phenomena, and this clearly will have to be a major concern in future work.

9 Discussion and Future Work

The primary influence on inductive learning applied to text categorization is the large number of features that natural language provides. Most documents can be uniquely identified by a small combination of words, making overfitting inevitable in the absence of some form of feature selection or pruning.

In fact, feature selection mechanisms themselves can overfit. On the MUC-3 corpus, DT-min10 was aided by an initial feature selection step even though it incorporates its own feature selection mechanism. This is in agreement with Almuallin and Dietterich's results suggesting that the decision tree learning algorithms ID3 and FRINGE are not effective at minimizing number of features used in the face of many irrelevant features

[AD91]. The success of our relatively simple pre-filtering strategy suggests that more research on feature selection would be profitable, even for relatively well-understood learning algorithms.

The temporal nature of category distribution, as exhibited by the Reuters EARN category, has rarely been discussed, but our results show it to pose a serious dilemma for the text categorization researcher or practitioner. The easiest approach to eliminating the effect of temporal fluctuations, and undoubtedly increasing performance, would be to use a random rather than chronological partitioning into training and test sets. This may be useful when it is desirable to study specific phenomena while controlling for temporal variation, but ignores a major issue for operational systems and in the worst case results in unwitting training on test data. (The Reuters text stream, for instance, contains many duplicate and near-duplicate documents.)

A better approach would be to use data sets including at least two years of a text stream, enabling seasonal fluctuations to be investigated directly. Categorization algorithms should not merely cope with temporal fluctuations but should make use of them. We plan to investigate making available to our standard learning algorithms features corresponding to time of publication, day, date of the week, month, and so on. More generally, we plan to investigate incremental learning algorithms that are designed to track concept drift [Fis87] and to see how the idea of cyclical changes in concept definition might be used.

Raw performance is not the only characteristic of interest in a learning algorithm for text categorization. For applications it is likely that manual editing or tuning of induced concept descriptions will be desirable. This may favor a symbolic concept description, such as those produced by DT-min10, over a probabilistic classifier, as produced by PropBayes.

10 Summary

Text categorization is both an important application area for machine learning, and a testbed for studying a number of issues that are likely to be important as learning algorithms increasingly face the real world: large, sparse feature sets of erratic quality, overlapping category definitions, and the time-varying nature of data streams. Our experiments show that current inductive learning algorithms can achieve respectable performance on text categorization tasks without additional knowledge engineering.

We also found that feature selection was of crucial importance, and that the contextual feature selection mechanism used by decision trees could be aided by an initial filtering of features using a global evaluation measure.

Acknowledgements

Thanks to Bruce Croft and Paul Utgoff for suggestions on this work. Ken Church alerted us to the problem of time-varying catgories and provided other helpful suggestions. Tom Mitchell provided support and a sounding board. Steve Harding and Peter Shoemaker provided programming support. Howard Turtle and Wray Buntine made software available. This research was supported by AFOSR under grant AFOSR-90-0110. Many thanks to Phil Hayes, Carnegie Group, and Reuters for making available the Reuters text categorization test collection, and to Beth Sundheim for making available the MUC-3 corpus.

References

[AD91] Almuallim, Hussein and Dietterich, Thomas G. Learning with many irrelevant features. AAAI-91, pages 547–552, 1991.

[BT91] Balcom, Laura Blumer and Tong, Richard M. Advanced Decision Systems: Description of the CODEX system as used for MUC-3. In *Proceedings of the Third Message Understanding Evaluation and Conference*, Los Altos, CA: Morgan Kaufmann, May 1991.

[BFL+88] Biebricher, Peter, Fuhr, Norbert, Lustig, Gerhard, Schwantner, Michael, and Knorz, Gerhard. The automatic indexing system AIR/PHYS—from research to application. In *Eleventh International Conference on Research & Development in Information Retrieval*, pages 333—342, 1988.

[BB63] Borko, Harold and Bernick, Myrna. Automatic document classification. *Journal of the Association for Computing Machinery*, pages 151–161, 1963.

[Bun90] Buntine, Wray. A theory of learning classification rules. PhD thesis, School of Computing Science, University of Technology, Sydney, February 1990.

[Bun91] Buntine, Wray. Introduction to IND and recursive partitioning. Technical Report. RIACS/NASA Ames Research Center, September 1991.

[COL83] Cerny, Barbara A., Okseniuk, Anna, and Lawrence, J. Dennis. A fuzzy measure of agreement between machine and manual assignment of documents to subject categories. In *Proceedings of the 46th ASIS Annual Meeting*, page 265, 1983.

[Cla85] Clancey, William J. Heuristic classification. *Artificial Intelligence*, 27:289–350, 1985.

[Cle91] Cleverdon, Cyril W. The significance of the Cranfield tests of index languages. In *Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1991.

[CFAT91] Crawford, Stuart L., Fung, Robert M., Appelbaum, Lee A., and Tong, Richard M. Classification trees for information retrieval. In *Eighth International Workshop on Machine Learning*, pages 245–249, 1991.

[CMSW91] Creecy, Robert H., Masand, Brij M., Smith, Stephen J., Waltz, David L. Trading MIPS and memory for knowledge engineering: automatic classification of census returns on a massively parallel supercomputer. Technical Report TMC-192, Thinking Machines Corp., Cambridge, MA, April 1991.

[DLW+91] Dahlgren, Kathleen, Lord, Carol, Wada, Hajime, McDowell, Joyce, and Stabler, Jr., Edward P. ITP Interpretext system: MUC-3 test results and analysis. In *Proceedings of the Third Message Understanding Evaluation and Conference*, Los Altos, CA: Morgan Kaufmann, May 1991.

[DeJ82] DeJong, Gerald. An overview of the FRUMP system. In Lehnert, Wendy G. and Ringle, Martin H., editors, *Strategies for Natural Language Processing*, pages 149–176. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1982.

[DGCN91] Dolan, Charles P., Goldman, Seth R., Cuda, Thomas V., and Nakamura, Alan M. Hughes Trainable Text Skimmer: Description of the TTS system as used for MUC-3. In *Proceedings of the Third Message Understanding Evaluation and Conference*, Los Altos, CA: Morgan Kaufmann, May 1991.

[DR91] Deogun, Jitender S. and Raghavan, Vijay V. Description of the UNL/USL system used for MUC-3. In *Proceedings of the Third Message Understanding Evaluation and Conference*, Los Altos, CA: Morgan Kaufmann, May 1991.

[DH73] Duda, Richard O. and Hart, Peter E. *Pattern Classification and Scene Analysis.* Wiley-Interscience, New York, 1973.

[Fis87] Fisher, Douglas H. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.

[FHL+91] Fuhr, Norbert, Hartmann, Stephan, Lustig, Gerhard, Schwantner, Michael, Tzeras, Konstadinos, and Knorz, Gerhard. AIR/X—a rule-based multistage indexing system for large subject fields. In *RIAO 91 Conference Proceedings: Intelligent Text and Image Handling*, pages 606–623, 1991.

[GC90] Gale, William A. and Church, Kenneth W. Poor estimates of context are worse than none. In *Speech and Natural Language Workshop*, pages 283–287, San Mateo, CA: Morgan Kaufmann, June 1990.

[GSM91] Grishman, Ralph, Sterling, John, and Macleod, Catherine. New York University description of the PROTEUS system as used for MUC-3. In *Proceedings of the Third Message Understanding Evaluation and Conference*, Los Altos, CA: Morgan Kaufmann, May 1991.

[Har88a] Hardt, S. L. On recognizing planned deception. In *AAAI-88 Workshop on Plan Recognition*, 1988.

[HW90] Hayes, Philip J. and Weinstein, Steven P. CONSTRUE/TIS: a system for content-based indexing of a database of news stories. In *Second Annual Conference on Innovative Applications of Artificial Intelligence*, 1990.

[Hob91] Hobbs, Jerry R. SRI International: Description of the TACITUS system as used for MUC-3. In *Proceedings of the Third Message Understanding Evaluation and Conference*, Los Altos, CA: Morgan Kaufmann, May 1991.

[JR90] Jacobs, Paul S. and Rau, Lisa F. SCISOR: Extracting information from on-line news. *Communications of the ACM*, 33(11):88–97, November 1990.

[Lew91a] Lewis, David D. Data extraction as text categorization: An experiment with the MUC-3 corpus. In *Proceedings of the Third Message Understanding Evaluation and Conference*, Los Altos, CA: Morgan Kaufmann, May 1991.

[Lew91b] Lewis, David D. Evaluating text categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 312–318, February, 1991.

[Lew92] Lewis, David D. Representation and learning in information retrieval. PhD thesis, Computer Science Dept., Univ. of Massachusetts at Amherst, February 1992. Technical Report 91–93.

[Lew92b] Lewis, David D. An evaluation of phrasal and clustered representations on a text categorization task. Submitted to *ACM SIGIR-92*.

[Mar61] Maron, M. E. Automatic indexing: An experimental inquiry. *Journal of the Association for Computing Machinery*, 8:404–417, 1961.

[Qui86a] Quinlan, J. R. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[Utg86] Utgoff, Paul E. Shift of bias for inductive concept learning. In Michalski, Ryszard S., Carbonell, Jaime G., and Mitchell, Tom M., editors, *Machine Learning. An Artificial Intelligence Approach. Volume II*, pages 107–148. Morgan Kaufmann, Los Altos, CA, 1986.

[van79] van Rijsbergen, C. J. *Information Retrieval.* Butterworths, London, second edition, 1979.

[VS87] Vleduts-Stokolov, Natasha. Concept recognition in an automatic text-processing system for the life sciences. *Journal of the American Society for Information Science*, 38:269–287, 1987.

[SS89] Sahin, Kenan and Sawyer, Keith. The Intelligent Banking System: natural language processing for financial communications. In Schorr, Herbert and Rappaport, Alain, editors, *Innovative Applications of Artificial Intelligence*, pages 43–50. AAAI Press, Menlo Park, CA, 1989.