Evaluation of Decision Forests on Text Categorization

Hao Chen[a] and Tin Kam Ho[b]

[a]School of Information Mgmt. & Systems, Univ. of California, Berkeley, CA 94720-4600, USA.
hchen@sims.berkeley.edu

[b]Bell Laboratories, Lucent Technologies, 700 Mountain Avenue, Murray Hill, NJ 07974, USA.
tkh@bell-labs.com

ABSTRACT

Text categorization is useful for indexing documents for information retrieval, filtering parts for document understanding, and summarizing contents of documents of special interests. We describe a text categorization task and an experiment using documents from the Reuters and OHSUMED collections. We applied the Decision Forest classifier and compared its accuracies to those of C4.5 and kNN classifiers, using both category dependent and category independent term selection schemes. It is found that Decision Forest outperforms both C4.5 and kNN in all cases, and that category dependent term selection yields better accuracies. Performances of all three classifiers degrade from the Reuters collection to the OHSUMED collection, but Decision Forest remains to be superior.

Keywords: text categorization, decision forest, decision tree, C4.5, k-nearest-neighbor, OHSUMED, Reuters, evaluation, information retrieval

1. INTRODUCTION

Text categorization is the classification of documents with respect to a set of one or more pre-existing categories.[1] The most common application of text categorization is in indexing documents for text retrieval, i.e. in producing document representatives. Manual assignment of subject categories to documents is a widely used form of text representation. Users can mention these subject categories in their requests, possibly enabling a more compact and effective query to be formed. However, manual assignment of categories requires considerable human labor. Moreover, it is subject to inconsistency of judgement of several individuals or even the same individual at different times. Replacing or aiding manual indexing with automated text categorization can reduce the costs substantially and maintain consistency.

Another application of text categorization is within text understanding systems. Categorization may be used to filter out documents or parts of documents that are unlikely to contain extractable data, without incurring the costs of more complex natural language processing.[2]

Finally, the categorization itself may be of direct interest to a human user, as in judging whether a threatening letter against a government official signifies real danger.[3]

A category may be binary (a document either is or is not a member of the category) or graded (a document can have a degree of membership in the category). Binary assignments have been used in most applications. When multiple categories are used, it may be the case that each document is assigned to exactly one category. On the other hand, categories may be assigned independently, with each document falling into all, some, or none of the chosen categories.

A growing number of statistical learning methods have been applied to this problem in recent years, including regression models,[4] nearest-neighbor classifiers,[5] Bayes belief networks,[6] decision trees,[4] rule learning algorithms,[7] neural networks,[8] and inductive learning techniques.[9] Here we report on our experiment with the Decision Forest classifier on the text categorization task. We used the C4.5 decision tree classifier [10] and the k-nearest-neighbor classifier as the baseline algorithms for comparison. We will describe general procedures for text categorization first, and then details of the procedures used in our experiment and associated results.

2. PROCEDURES

Text categorization involves data collection, feature selection and extraction, classification, and evaluation.

2.1. Data Collection

In order to objectively compare different text categorization methods, a standard data collection should be used in the evaluation experiments. However, this appears to be a serious problem. There are several different collections, and even when the same collection is chosen, there are many alternative ways that the data in the collection are used. Two of the most commonly used collections are the Reuters collection[11] and the OHSUMED collection.[12]

2.1.1. Reuters Collection

The Reuters corpus consists of over 20,000 documents appearing on Reuters Newswire in 1987. The original collection containing 22,173 documents (called Reuters-22173) were assembled and indexed with categories by Reuters Ltd. and Carnegie Group, Inc. in 1987. Later, Steve Finch and David D. Lewis cleaned it up by removing 595 duplicated documents, resulting in 21,578 documents (called Reuters-21578). In addition to the text of the news, each document contains a specification of what categories it belongs to. There are 5 category sets: *exchange, organization, people, place*, and *topic.* The "topic" category set which contains 135 categories is the most commonly used in text categorization experiments. A document in the collection can belong to several, one, or no categories.

There are a number of ways how the documents are split into training and test data. Two of the commonly used ones are the *Modified Lewis Split* and *Modified Apte Split.*

- Modified Lewis Split: In this split, 1765 documents are removed from Reuters-21578. The remaining ones from April 7, 1987 and before are assigned to the training set (totaling 13,625 documents), and the ones from April 8, 1987 and after are assigned to the test set (totaling 6,188 documents). Note that some of the documents in the training and test test have no topic category.
- Modified Apte Split: Remove all the documents that have no topic category from the training set and test set of the Modified Lewis Split. The resulting training set containing 9,603 documents and the resulting test set containing 3,299 documents constitutes the Modified Apte Split. 8,676 out of the original 21,578 documents are not used in either the training and test set.

2.1.2. OHSUMED collection

The OHSUMED corpus, developed by William Hersh and colleagues at the Oregon Health Sciences University, is a subset of the documents in the MEDLINE database. It consists of 348,566 references from 270 medical journals from the years 1987 to 1991. The documents were manually indexed using subject categories (Medical Subject Headings, or MeSH; about 18,000 categories defined) in the National Library of Medicine.

MeSH terms consist of a *main heading* optionally flagged with *subheadings* and *importance markers.* A total of 14,626 distinct main headings occur in the OHSUMED records. Of the 348,566 OHSUMED records, all but 23 have MeSH categories assigned, but only 233,445 of them have abstracts.

2.2. Feature Selection and Extraction

Feature selection chooses which features to be used in classification. In text categorization, features are often measures of frequencies of words appearing in a document. It is preferable to use less features than the raw measurements (say, frequency of each word), so that classification will be performed in a feature space of a lower dimensionality. By reducing the dimensions of the feature space, it not only increases the efficiency of the training and test processes, but also reduces the risk of overfitting the model to data. Feature extraction computes the chosen features from an input document. In statistical classification, features are represented in a numerical vector, which is subsequently used by the classifiers. Feature selection involves stop word removal, stemming, and term selection.

2.2.1. Stop Word Removal

Words used in text indexing and retrieval are called *terms*. According to the *term discrimination model*,[13] moderate frequency terms discriminate the best. High frequency words, which are called *stop words*, have low information content, and therefore have weak discriminating power. They are removed according to a list of common stop words.

2.2.2. Stemming

Stemming reduces morphological variants to the root word. For example, “asks”, “asked”, and “asking” are all reduced to “ask” after stemming. This relates the same word in different morphological forms and reduces the number of distinctive words. The *Porter stemmer* is a commonly used stemmer.[14]

2.2.3. Term Selection

Even after the removal of stop words and stemming, the number of distinct words in a document set may still be too large, and most of them appear only occasionally. In addition to removing high frequency words, the term discrimination model suggests that low frequency words are hard to learn about and therefore do not help much. They should be removed to reduce the dimensions of the vector space as well.

Y. Yang and J. Pederson performed a comparative study on five feature selection methods: Document Frequency (DF), Information Gain (IG), Mutual Information (MI), χ_2-test (CHI), and Term Strength (TS).[15] It was found that IG and CHI were the most effective, while DF thresholding performed similarly. They found strong correlations between the DF, IG, and CHI values of a term, which suggests that DF thresholding, the simplest method with the lowest cost in computation, can be reliably used instead of the very computationally expensive IG and CHI.

Document Frequency is the number of documents that a term appears in. All terms whose document frequencies are above a threshold are selected. There are two different ways that document frequency is calculated:

- Category independent term selection. In this case, document frequency of each term is calculated from all the documents in the collection. The same selected set of terms are used on each category.
- Category dependent term selection. In this case, for each category, document frequency of each term is calculated from only those documents belonging to that category. As a result, different sets of terms are selected for different categories.

2.2.4. Feature Extraction

After the terms are selected, for each document a feature vector is generated whose elements are the feature values of each term. A commonly used feature value is the TF (Term Frequency) × IDF (Inverse Document Frequency) measure, defined as follows:

$$\begin{aligned} \text{TF} &= \text{Number of occurrence of a term in a document} \\ \text{IDF} &= \log \frac{\text{Number of documents in the set}}{\text{Number of documents where this term appears}} \end{aligned}$$

2.3. Classification

A number of classifiers have been tried on text categorization. Y. Yang evaluated the performance of 14 statistical classifiers on the Reuters collection and OHSUMED collection.[16] The classifiers are: CONSTRUE, Decision Tree, Naive Bayes, SWAP-1, Neural Networks, CHARADE, RIPPER, Rocchio, Exponentiated Gradient, Widrow-Hoff, Sleeping Experts, LLSF, kNN, and WORD. Note that not every classifier was tested on every collection. In Yang's evaluation kNN is among the best performing classifiers.

In our experiment, we focused on the evaluation of the Decision Forest classifier on text categorization. We compare its accuracies to those of C4.5 and kNN.

2.3.1. Binary Classification vs. Category Ranking

Category ranking ranks all the categories that a document is deemed belonging to. It assigns weights to each of these categories. On the other hand, binary classification determines whether a document belongs to each of the categories. In our experiment only binary classification was studied.

2.3.2. kNN Classifier

Given a document in the test set, the k-nearest neighbor classifier (kNN) ranks its nearest neighbors among the training documents according to a distance measure, and uses the most frequent categories of the k top-ranking neighbors to predict the categories of the input document. In binary classification, each training document is first coded with a "yes" or "no" for each category, and each test document is decided on that category according to whether there are more "yes" instances or "no" instances among the k top-ranking neighbors from the training set.

2.3.3. C4.5 Classifier

C4.5 is a decision tree classifier that was developed by Quinlan.[10] The training algorithm constructs a decision tree by recursively splitting the data set using a test of maximum *gain ratio*, subject to the constraint that *information gain* due to the split must also be large. The tree is grown until it partitions the feature space into regions (leaves) containing only one class or classes that cannot be further separated, or when an early stopping criterion is triggered which prevents over-fragmentation of the space. The tree can be pruned back based on an estimate of error on unseen cases. During classification a test vector is evaluated according to the chosen tests at each split, and when it arrives at a leave, estimates are given for probabilities of its belonging to each category. In binary classification, for each category a tree is built using all the training data labeled as "yes" or "no" for that category. Test documents are assigned "yes" or "no" for each category according to the corresponding tree.

2.3.4. Decision Forest

A Decision Forest[17] is a collection of decision trees together with a decision combination function. Multiple decision trees are constructed systematically by pseudo-randomly selecting subsets of components of the feature vector, that is, each tree is constructed in a randomly chosen subspace. The classifier decides by maximizing an average of the estimates of posterior probabilities given by individual trees. It has been demonstrated that in many practical applications the method outperforms single decision trees constructed using all the available features, regardless of the procedures used to construct the individual trees. In binary classification, for each category a forest is built using all the training data labeled as "yes" or "no" for that category. Test documents are decided similarly according to the corresponding forest.

2.4. Evaluation

2.4.1. Evaluation of Category Ranking

The performance of category ranking can be evaluated in terms of *precision* and *recall*, computed at any threshold on the ranked list of categories of each document:

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

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

2.4.2. Evaluation of Binary Classification

The category assignment of a binary classifier can be evaluated using a two-way contingency table (Table 1):

	YES is correct	NO is correct
Assigned YES	a	b
Assigned NO	c	d

Table 1. Contingency Table

Precision is defined as $a/(a+b)$, and recall is defined as $a/(a+c)$. For evaluating performance average across categories, there are two conventional methods, namely *macro-averaging* and *micro-averaging*. Macro-averaged performance scores are computed by first computing the scores for the per-category contingency tables and then averaging these per-category scores to compute the global means. Micro-averaged performance scores are computed by first creating a global contingency table whose cell values are the sums of the corresponding cells in the per-category contingency tables, and then use this global contingency table to compute the micro-averaged performance scores. There is an important distinction between macro-averaging and micro-averaging. Micro-averaging performance scores give equal weight to every document, and is therefore considered a per-document average. Likewise, macro-average performance scores give equal weight to every category, regardless of its frequency, and is therefore a per-category average.

F_1 measure combines the precision and recall into one measure, which is defined as:

$$F_1 = \frac{2rp}{r+p}$$

where r and p are recall and precision respectively.

3. THE EXPERIMENT

The experiment was carried out using kNN, C4.5, and Decision Forest classifiers on both the Reuters and OHSUMED collections.

3.1. Text Collections

Two text collections were used in the experiments.

- Reuters-21578 collection with Modified Apte Split. There are 9,603 documents in the training set, 3,299 documents in the test set, and 8,676 documents are unused. Out of the 5 category sets, the topic category set contains 135 categories, but only 95 categories have at least one document in the training set. These 95 categories were used in the experiment.
- OHSUMED collection. Our experiment used only the 233,445 records that have both abstracts and MeSH categories. We chose our training set from the 183,229 documents published during 1987 to 1990, and the testing set from the 50,216 documents during 1991. We focused on the set of 119 MeSH categories in the Heart Disease subtree of the Cardiovascular Diseases *tree structure*. We used only the 75 categories with 15 or more training documents. This resulted in a training set of 12,327 documents and a testing set of 3,616 documents.

3.2. Feature Selection and Extraction

1. Stop word removal: stop words (from a list of 430, provided by David Lewis) were removed from each document.
2. Stemming: Porter's stemmer was applied.
3. Term selection: Terms were selected based on Document Frequency thresholding. Both category independent and category dependent term selection were performed on the Reuters collection, but only category dependent term selection was performed on the OHSUMED collection.
   - In the category independent term selection on the Reuters collection, 779 terms were selected.
   - In the category dependent term selection on the Reuters collection, an average of 367 terms were selected per category. Each category has a different number of selected terms.
   - In the category dependent thresholding on the OHSUMED collection, an average of 626 terms were selected per category. Each category has a different number of selected terms.
4. Feature extraction: A feature vector was created for each document. Each element in the feature vector is the product of Term Frequency and Inverse Document Frequency of each term.

3.3. Classification

Three classifiers were used in the experiment: Decision Forest, C4.5, and kNN. Binary classification was performed on all documents on each category. The kNN classifier used Euclidean distance as the metric. The C4.5 trees were constructed and pruned using all the default parameters. The forests used oblique hyperplanes derived by central axis projection for splits, and contained 19 trees each. Different parameter settings were tried for the Decision Forest (with $f = 0.25$ and $f = 0.5$, f being the fraction of all features to be used in each tree, choices of values following[17]) and kNN (with $k = 30$, $k = 45$, and $k = 65$, choices of values following[15]).

3.4. Evaluation

The results were evaluated by precision, recall, and F_1-value, all micro-averaged across different categories. We chose micro-averaging over macro-averaging because the number of documents belonging to each category varies considerably among different categories, so that the micro-averaging, a per-document averaging, makes more sense.

4. RESULTS AND DISCUSSIONS

4.1. Results

The results on the Reuters-21578 collection are shown below. In Table 2 are results with category independent term selection, while in Table 3 are results with category dependent term selection.

Classifier	Parameter	Precision	Recall	F$_1$ Value
kNN	k=30	86.88%	57.62%	69.29%
	k=45	89.00%	54.81%	67.84%
	k=65	89.30%	52.96%	66.49%
C4.5		73.77%	60.62%	66.55%
Decision Forest	f=0.25	89.13%	60.16%	71.84%
	f=0.50	88.44%	63.11%	73.66%

Table 2. Results on Reuters-21578 with category independent term selection

Classifier	Parameter	Precision	Recall	F$_1$ Value
kNN	k=30	88.53%	52.88%	66.21%
	k=45	89.49%	49.24%	63.52%
	k=65	90.04%	46.69%	61.50%
C4.5		75.91%	65.06%	70.07%
Decision Forest	f=0.25	89.37%	62.36%	73.46%
	f=0.50	88.48%	65.57%	75.32%

Table 3. Results on Reuters-21578 with category dependent term selection

The results on OHSUMED collection, using category dependent term selection, are shown in Table 4.

Classifier	Parameter	Precision	Recall	F$_1$ Value
kNN	k=30	80.02%	35.24%	48.93%
	k=45	82.40%	31.43%	45.50%
	k=65	83.17%	28.92%	42.92%
C4.5		63.87%	60.17%	61.96%
Decision Forest	f=0.25	84.11%	50.44%	63.06%
	f=0.50	82.60%	54.88%	65.95%

Table 4. Results on OHSUMED collection with category dependent term selection

4.2. Discussions

There is a trade-off between precision and recall. Precision can usually be increased at the expense of recall, and vice versa. The F_1 measure is derived from both precision and recall, therefore taking the trade-off into account. We use the F_1 measure mainly when examining performances. However, by the nature of its design, kNN tends to have higher precision than recall, especially when k is large. The reason is that all k neighbors have to vote to produce a decision, and if k is larger, it is more difficult to achieve an agreement (therefore lower recall), but whenever a decision is made it is more likely to be correct (therefore higher precision). This effect should be noted when taking the F_1 value as the sole performance standard.

In each of the three tests, Decision Forest outperforms C4.5 and kNN classifiers. In the two tests with category dependent term selection, C4.5 outperforms kNN. However, in the test with category independent term selection, kNN outperforms C4.5.

In the tests on the Reuters collection, category dependent term selection outperforms category independent term selection with both the Decision Forest and C4.5 classifiers. This is intuitive because category dependent term selection is more precise for the binary classification performed on each category. However, the results also show that the opposite is true of the kNN classifier on the Reuters collection, using each of the three parameter settings ($k = 30$, $k = 45$, and $k = 65$), and the lower F_1-values are due mostly to failure of recall for many categories. Note that number of terms used in category dependent selection is highly variable, ranging from 41 terms for some category to 965 for another category. There appears to be no simple answer to the question why kNN behaves this way.

In each of the three tests, kNN performs the best with parameter $k = 30$, followed by $k = 45$, and then $k = 65$. This suggests that in future investigations even smaller value of k should be considered. Decision Forest performs better with parameter $f = 0.5$ than $f = 0.25$, which is consistent with previous results from other domains.

The performances of all three classifiers degrade from the Reuters collection to the OHSUMED collection. The performance of kNN classifier degrades more drastically ($\approx 26\%$) than C4.5 ($\approx 12\%$) and Decision Forest ($\approx 12\%$). It should be noted that the distribution of the documents across categories is highly uneven in the Reuters collection, with about half of the documents belonging to one of two largest categories, and almost all other categories are very small. In the OHSUMED collection, the distribution is more even though there are still several large categories. We believe that such a difference in the distributions affects the kNN recall rate more than the others, due to the fact that there are more confusion classes in a fixed size neighborhood.

5. CONCLUSIONS

We experimented with text categorization using three classifiers and two standard document collections. We found that Decision Forest classifiers are very useful in this task, and are substantially better than single decision trees (C4.5) and kNN classifiers which were shown previously among the best performing statistical methods. We expect that the superiority of Decision Forests will be maintained in other tasks involving categorization of passages of text.

We also noticed that there are many difficulties in making a fair comparison with results of other classifiers obtained by other groups outside this experiment. Besides the implementation of the classifiers, many details can differ in the design of the experiments, including the choices of training and testing sets, list of stop words, stemming algorithms, and selection of the feature terms as well as the way their frequencies are counted. Since any of these procedures can have a significant effect on the performance, caution should always be taken in making references to published performance figures.

ACKNOWLEDGMENTS

We would like to thank David Lewis at AT&T Labs Research for providing us with part of the data and for very helpful discussions.

REFERENCES

1. D. Lewis, *Representation and Learning in Information Retrieval.* PhD thesis, Univ. of Massachusetts at Amherst, 1992.
2. K. Dahlgren, C. Lord, H. Wada, J. McDowell, and E. Stabler, "Itp interpretext system: Muc-3 test results and analysis," in *Proceedings of the Third Message Understanding Evaluation and Conference*, Morgan Kaufmann, (Los Altos, CA), May 1991.
3. S. Hardt, "On recognizing planned deception," in *AAAI-88 Workshop on Plan Recognition*, 1988.
4. N. Fuhr, S. Hartmanna, G. Lustig, M. Schwantner, and K. Tzeras, "Air/x – a rule-based multistage indexing systems for large subject fields," in *Proceedings of RIAO'91*, pp. 606–623, 1991.
5. Y. Yang and C. Chute, "An example-based mapping method for text categorization and retrival," *ACM Transaction on Information Systems* , pp. 253–277, 1994.
6. D. Lewis and M. Ringuette, "Comparison of two learning algorithms for text categorization," in *Proceedings of the Third Annual symposium on Dcoument Analysis and Information Retrieval (SDAIR'94)*, 1994.
7. C. Apte, F. Damerau, and S. Weiss, "Towards language independent automated learning of text categorization models," in *Proceedings of the 17th Annual ACM/SIGIR Conference*, 1994.
8. E. Wiener, J. Pedersen, and A. Weigend, "A neural network approach to topic spotting," in *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, 1995.
9. D. Lewis, R. Schapire, J. Callan, and R. Papka, "Training algorithms for lineare text classifiers," in *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 298–306, 1996.
10. J. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
11. "Reuters collection." `http://www.research.att.com/~lewis/reuters21578.html`.
12. "OHSUMED collection." `ftp://medir.ohsu.edu/pub/ohsumed`.
13. G. Salton, A. Wong, and C. Yang, "A vector space model for automatic indexing," *Communications of the ACM* **18**, pp. 613–620, 1975.
14. W. Frakes and R. Baeza-Yates, eds., *Information Retrieval: Data Structures & Algorithms*, Prentice Hall, 1992.
15. Y. Yang and J. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, 1997.
16. Y. Yang, "An evaluation of statistical approaches to text categorization," Tech. Rep. CMU-CS-97-127, Carneigie Mellon University, 1997.
17. T. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(8), pp. 832–844, 1998.