Representational Issues in Machine Learning of User Profiles

+*Eric Bloedorn, +Inderjeet Mani, and +T. Richard MacMillan

+Artificial Intelligence Technical Center
The MITRE Corporation, Z401
7525 Colshire Drive, McLean, VA 22102
{bloedorn,imani,macmilla}@mitre.org

*Machine Learning and Inference Laboratory
George Mason University, Fairfax, VA 22030

Abstract

As more information becomes available electronically, tools for finding information of interest to users become increasingly important. Building tools for assisting users in finding relevant information is often complicated by the difficulty in articulating user interest in a form that can be used for searching. The goal of the research described here is to build a system for generating comprehensible user profiles that accurately capture user interest with minimum user interaction. Machine learning methods offer a promising approach to solving this problem. The research described here focuses on the importance of a suitable generalization hierarchy and representation for learning profiles which are predictively accurate and comprehensible. In our experiments using AQ15c and C4.5 we evaluated both traditional features based on weighted term vectors as well as subject features corresponding to categories which could be drawn from a thesaurus. Our experiments, conducted in the context of a content-based profiling system for on-line newspapers on the World Wide Web (the IDD News Browser) demonstrate the importance of a generalization hierarchy in obtaining high predictive accuracy, precision and recall, and stability of learning.

Introduction

As more information becomes available on the Internet, the need for effective personalized information filters becomes critical. In particular, there is a need for tools to capture profiles of users' information needs, and to find articles relevant to these needs, as these needs change over time. Information filtering, as (Belkin and Croft 92), (Foltz and Dumais 92) point out, is an information access activity similar to information retrieval, but where the profiles represent evolving interests of users over a long-term period, and where the filters are applied to dynamic streams of incoming data. The research described here automates the task of building and adapting accurate and comprehensible individualized user profiles and focuses on the importance of a suitable generalization hierarchy and representation for learning.

Our research builds on two particular traditions involving the application of machine learning to information access: empirical research on relevance feedback within the information retrieval community, and interdisciplinary work involving the construction of personalized news filtering agents. We will now introduce these briefly, to better motivate and distinguish our work.

Relevance feedback approaches are a form of supervised learning where a user indicates which retrieved documents are relevant or irrelevant. These approaches, e.g., (Rocchio 1971), (Robertson & Sparck-Jones 1976), (Belew 1989), (Salton & Buckley 1990), (Harman 1992), (Haines & Croft 1993), (Buckley, Salton, & Allan 1994), have investigated techniques for automatic query reformulation based on user feedback, such as term reweighting and query expansion. While this body of work is not necessarily focused exclusively on the information filtering problem, it demonstrates effectively how learning can be used to improve queries.

Work on the application of machine learning techniques for constructing personalized information filters has gained momentum in recent years. Some early MIT Media Lab work used a genetic algorithm approach to generate new profiles, which were evaluated based on user feedback (Sheth & Maes 1993), (Sheth 1993). One of the goals of that approach was "exploratory behavior.... so as to explore newer domains that might be of interest to the user." (Sheth & Maes 1993). Since that time, a number of other systems for personalized information filtering have appeared on the scene, such as NewT (Maes 1994), Webhound (Lashkari, Metral, & Maes 1994), WebWatcher (Armstrong et al. 1995), WebLearner (Pazzani et al. 1995) and NewsWeeder (Lang 1995).

One of the motivations for our approach was the discovery that the above research had paid little attention to learning generalizations about user's interests. For example, if a user likes articles on *scuba*, *whitewater rafting*, and *kayaking*, a system with the ability to generalize could infer that the user is interested in *water sports*, and could communicate this inference to the user. Not only would this be a natural suggestion to the user, but it might also be useful in quickly capturing their real interest and suggesting what additional information might be of interest. Such an approach could exploit a concept hierarchy or network to perform the generalizations. While thesauri and other conceptual representations have been the subject of extensive investigation in both query formulation and expansion (e.g., see (Jones et al. 1995) for detailed references), they have not been used to learn generalized profiles.

In order to investigate this further, we decided to use features which would allow us to exploit categories for generalization, where the categories could be drawn from a thesaurus. One well-known problem which arises here is that of word-sense disambiguation, in this case deciding which of several thesaurus categories are the most likely ones for a term. We decided to apply the approach used by (Liddy & Paik 1992) (Liddy & Myaeng 1992), which exploits evidence from local context and large-scale statistics. This resulted in our using the Subject Field Coder (SFC) (Liddy and Myaeng 1992) (Liddy and Paik 1992) (from TextWise, Inc.), which produces a vector representation of a text's subject categories, based on a thesaurus of 124 subject categories (the SFC is discussed in more detail in the next section). We therefore decided to use a vector of subject categories in our document representation, with the SFC thesaurus being used for generalization. In order to compare the influence of these features on learning compared to more traditional features based on weighted term vectors, we developed a hybrid representation which combined the two types of features.

A personalized news filtering agent which engages in exploratory behavior must gain the confidence of the user. In many practical situations, a human may need to validate or edit the system's learnt profiles; as (Mitchell et al. 1994) point out, intelligibility of profiles to humans is important in such situations. We speculated that the use of such a hybrid representation which exploits summary-level features such as subject categories would increase the intelligibility of profiles. To further strengthen profile intelligibility, we also decided to include other summary-level features in our document representation, involving terms relating to people, organizations, and places (along with their respective attributes). These features were provided by a name tagger (discussed in the next section). That such features could help profile learning was suggested in part by some recent query reformulation research (Broglio & Croft 1993), which had shown improved retrieval performance on TIPSTER queries using such features.

In summary, our experiments evaluated the effects of different subsets of features on the learning of intelligible profiles. Our experiments were conducted in the context of a content-based profiling system for on-line newspapers on the World Wide Web, the IDD News Browser (Mani et al. 1995). In this system, which is in use at MITRE, the user can set up and edit profiles, which are periodically run against various collections built from live Internet newspaper and USENET feeds, to generate matches in the form of personalized newspapers. These personalized newspapers provide multiple views of the information space in terms of summary-level features. When reading their personalized newspapers, users provide positive or negative feedback to the system, which are then used by a learner to induce new profiles. These system-generated profiles can be used to make recommendations to the user about new articles and collections. The experiments reported here investigate the effect of different representations on learning new profiles.

Text Representation

As mentioned earlier, we used a hybrid representation with three different sources of features. We now describe these in turn.

The Subject Field Coder (SFC) (Liddy & Myaeng 1992) (Liddy & Paik 1992) (from TextWise, Inc.) produces a summary-level semantic representation of a text's contents, based on a thesaurus of 124 subject categories. Text summaries are represented by vectors in 124-dimensional space, with each vector's projection along a given dimension corresponding to the salience in the text of that subject category. The overall vector is built up from sentence-level vectors, which are constructed by combining the evidence from local context (e.g., unambiguous words) with evidence from large-scale statistics (e.g., pairwise correlations of subject categories). An earlier version of the SFC, which used subject codes from Longman's Dictionary of Contemporary English (LDOCE), was tested on 166 sentences from the Wall Street Journal (1638 words). It gave the right category on 87% of the words (Liddy & Myaeng 1992).

The second extraction system we used was the IDD POL Tagger (Mani et al 1993), (Mani & MacMillan 1995) which classifies names in unrestricted newswire text in terms of a hierarchy of different types of people (military officers, corporate officers, etc.), organizations (drug companies, government organizations, etc.), and places (cities, countries, etc.), along with their attributes (e.g., a person's title, an organization's business, a city's country, etc.) The tagger combines evidence from multiple

Features	Description
x1..x5	Top 5 subject categories as computed by the SFC text classifier.
x6..x59	POL people tags as computed by the IDD POL tagger. For each person identified, the vector contains the following string features: (name, gender, honorific, title, occupation, age). 9 people (each with these subfields) are identified for each article.
x60..x104	POL organization tags as computed by the IDD POL tagger. For each organization identified, the vector contains the following string features: (name, type, acronym, country, business). 9 organizations (each with these subfields) are identified for each article.
x105..x140	POL location tags as computed by the IDD POL tagger. For each location identified, the vector contains the following string features: (name, type, country, state) 9 locations (each with these subfields) are identified for each article.
x141..x141+n	The top n ranked tf.idf terms t1...tn are selected over all articles. For each article, position k in t1...tn has the tf.idf weight of term tk in that article.

Figure 1. A description of the features used to represent text

knowledge sources, each of which uses patterns based on lexical items, parts of speech, etc., to contribute evidence towards a particular classification. In trials against hand-tagged documents, the tagger was shown as having an average precision-recall accuracy (the average of precision and recall at a particular cutoff) of approximately 85%, where precision is calculated as the ratio of the Number of Correct Program Tags to the Number of Program Tags and recall is the ratio of the Number of Correct Program Tags to the Number of Hand Tags.

The statistical features we used were generated by a term-frequency inverse-document-frequency (tf.idf) calculation (Salton & McGill 1983)(Sparck-Jones 1972), which is a well-established technique in information retrieval. The weight of term k in document i is represented as:

$$dw_{ik} = tf_{ik} * (\log_2(n) - \log_2(df_k) + 1)$$

tf_{ik} = frequency of term k in document i

df_k = number of documents in which term k occurs.

n = total number of documents in collection

Given these three sources of features, we developed a hybrid document representation (Figure 1), described as follows: Features describe subjects (x1..x5), people (x6..x59), organizations (x60..x104) and locations (x105..x140) present in each news article. The top n statistical keywords are also included in the vector describing the article (x141..x141+n), where n was varied from 5 to 200. For convenience, x6..x140 are referred to as POL features.

Generalization Hierarchy

The hierarchy came to us from TextWise Inc.'s thesaurus. The SFC subject vectors describing individual articles use terms from the lowest level (terminals) of the hierarchy, which initially consisted of 124 categories. Although this thesaurus covers a fairly wide set of subjects-as required in our newswire application-it only has three levels, and as such does not have a great deal of depth. We extended the set of terminal categories under **medicine**, to include another 16 lowest level categories. In Figure 2, we show a fragment of the extended hierarchy under **sci+tech** (scientific and technical).

Learning Method

Our representational decisions suggested some constraints

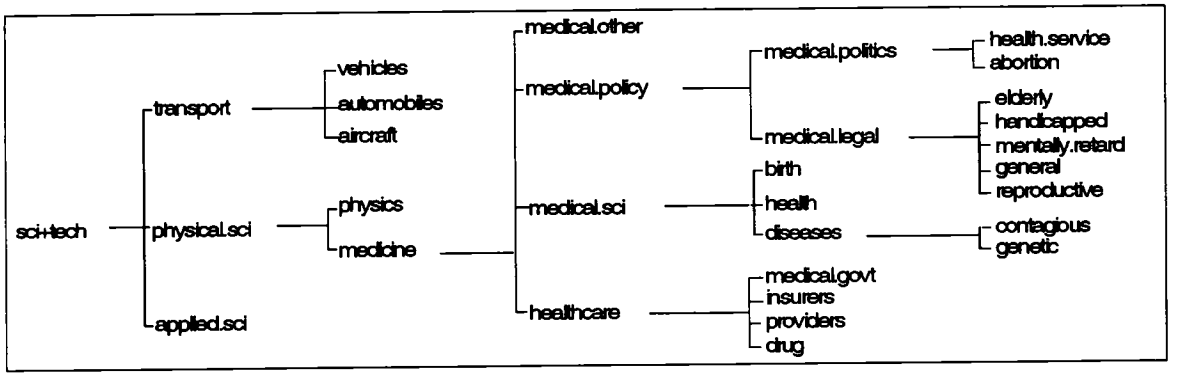

Figure 2. A fragment of the generalization hierarchy used in the experiments

on the learning method. We wanted to use learning methods which performed inductive generalization, where the SFC generalization hierarchy could be exploited. Also, we required a learning algorithm whose learnt rules could be made easily intelligible to users. We decided to try both AQ15c (Wnek, Bloedorn, & Michalski 1994) and C4.5-Rules (Quinlan, 1992) because they meet these requirements (the generalization hierarchy is made available to C4.5 by extending the attribute set), are well-known in the field and are readily available.

AQ15c is based on the A^q algorithm for generating disjunctive normal form (DNF) expressions with internal disjunction from examples. In the A^q algorithm rule covers are generated by iteratively generating stars from randomly selected seeds. A star is a set of most general alternative rules that cover that example, but do not cover any negative examples. A single 'best' rule is selected from this star based on the user's preference criterion (e.g. maximal coverage of new examples, minimal number of references, minimal cost, etc.). The positive examples covered by the best rule are removed from consideration and the process is repeated until all examples are covered.

C4.5-Rules, which is part of the C4.5 system of programs, generates rules based on decision trees learned by C4.5. In C4.5 a decision tree is built by repeatedly splitting the set of given examples into smaller sets based on the values of the selected attribute. An attribute is selected based on its ability to maximize an expected information gain ratio. In our experiments we found the pruned decision rules produced the most accurate predictions.

Another advantage of experimenting with these two learning methods is that we get to see if the representation we have developed for this problem is truly providing useful information, or if it is just well-matched to the bias of the selected learning algorithm. The learning preference in AQ15c is controlled by the preference criteria, which by default, is to learn simple rules. The preference for C4.5 is to select attributes which maximize the information gain ratio. This can sometimes lead to different hypotheses. Because of its ability to learn rules with internal disjunction AQ15c can easily learn rules which are conjunctions of many internal disjunctions. This type of concept is not easily represented in decision trees and thus not likely to be found by C4.5-Rules. We thought this may give an advantage to AQ15c in this domain, but based on our experimental results described below, it appears our representation is well suited to either learning bias.

Experimental Design

The goal of these experiments was to evaluate the influence of different sets of features on profile learning. In particular, we wanted to test the hypothesis that semantic features used for generalization were useful in profile learning. Each of the experiments involved selecting a source of documents, vectorizing them, selecting a profile, partitioning the source documents into documents relevant to the profile (positive examples) and irrelevant to the profile (negative examples), and then running a training and testing procedure. The training involved induction of a new profile based on feedback from the pre-classified training examples. The induced profile was then tested against each of the test examples. One procedure used 10 runs in each of which the examples were split into 70% training and 30% test (70/30-split). Another procedure used a 10-fold cross-validation, where the test examples in each of the 10 runs were disjoint (10-fold-cross).

The metrics we used to measure learning on the USMED and T122 problems include both predictive accuracy and precision and recall. These metrics are defined as shown in Figure 3. Precision and recall are standard metrics in the IR community, and predictive accuracy is standard in the ML community. Predictive accuracy is a reasonable metric when the user's objective function assigns the same cost to false positives and false negatives. When the numbers of false positives, true positives, false negatives, and true negatives are about equal, predictive accuracy tends to agree with precision and recall, but when false negatives predominate there can be large disagreements.

Our first experiment exploited the availability of users of the IDD News Browser. A user with a "real" information need was asked to set up an initial profile. The articles matching his profile were then presented in his personalized newspaper. The user then offered positive

Metric	Definition
Predictive Accuracy:	# examples classified correctly / total number of test examples.
Precision:	# positive examples classified correctly / # examples classified positive, during testing
Recall:	# positive examples classified correctly / # known positive, during testing
Precision Learning Curve:	Graph of average precision vs. % of examples used in training
Recall Learning Curve:	Graph of average recall vs. % of examples used in training
Averaged Precision (Recall):	Average of Precision (Recall) over all test runs.

Figure 3. Metrics used to measure learning performance

Learning Method	Learning Problem	Predictive Accuracy				Average Precision/ Average Recall			
		TFIDF	POL	SFC	ALL	TFIDF	POL	SFC	ALL
AQ15c	USMED	0.58	*0.48*	**0.78**	0.55	0.51/1.00	0.45/0.45	**0.78/0.73**	*0.52/0.34*
	T122	*0.39*	0.59	0.59	**0.76**	0.36/0.88	0.43/0.66	*0.50/0.33*	**0.79/ 0.48**
C4.5-Rules	USMED	*0.39*	0.74	**0.79**	0.76	*0.07/0.30*	0.89/0.60	**0.97/0.60**	0.90/0.60
	T122	*0.64*	0.65	0.68	**0.76**	*0.0/0.0*	0.64./0.22	0.58 /0.55	**0.70/ 0.67**

Table 1. Predictive Accuracy, Average Precision, and Average Recall of learned profiles for a given feature set (averaged over 10 runs). (Best profiles generated are in boldface, outlined in thick lines. Worst profiles generated are in italics, outlined in double lines.)

and negative feedback on these articles. The set of positive and negative examples were then reviewed independently by the authors to check if they agreed in terms of relevance judgments, but no corrections needed to be made. In order to ensure that a relevant generalization hierarchy would be available for the learner, we extended the broad-subject thesaurus of the SFC to include several nodes under medicine. This involved adding in terms for medicine into the thesaurus. The details of the test are:

Source: Colorado Springs Gazette Telegraph (Oct. through Nov. 1994) Profile: "Medicine in the US" (USMED) Relevance Assessment: users, machine aided Size of collection: 442 Positive Examples: 18 Negative Examples: 20 Validation: "70/30-split"

Our next experiment exploited the availability of a standard test collection, the TREC-92 collection. The same generalization hierarchy used in the previous experiment was used here too. The idea was to study the effect that these changes in the hierarchy would effect learning of the other topics. The details of the test are:

Source: Wall Street Journal (1987-92), Profile: "RDT&E of New Cancer Fighting Drugs" (T122) Relevance Assessment: provided by TREC, Size of collection: 203, Positive Examples: 73, Negative Examples: 130, Validation: "10-fold cross"

Experimental Results

In our first set of experiments we applied AQ15c and C4.5-Rules to the USMED and T122 datasets. Here AQ15c has the hierarchy available to it in the form of hierarchical domain definitions for attributes x1 through x5. C4.5 has a hierarchy available to it through an extended attribute set. In this extension, based on a pointer from Quinlan (Quinlan, 1995), we extended the attribute set to include attributes which describe nodes higher up on the generalization hierarchy. A total of eighteen additional attributes were added (six for each non-null subject attribute) which provided the values of the subject attributes at each of the six levels higher in the tree from the leaf node. Because the tree was unbalanced some of the additional attributes took dummy values for some examples.

Predictive Accuracy

The predictive accuracy results (Table 1) show that the most predictively accurate profiles generated (boldface, outlined in thick lines) come from either the SFC or ALL feature sets, and the poorest profiles (italics, outlined in double lines) come from the POL or the TFIDF featureset. The TFIDF scores are shown for n=5; there was no appreciable difference for n=200. All differences between the best and worst predictive accuracies are significant to the 90% level and were calculated using a student t-test.

From this we can infer that, for topics such as these, profile learning using summary-level features (POL or SFC) alone can sometimes be more accurate in terms of predictive accuracy than using term-level features (TF.IDF) alone. In particular, having a generalization hierarchy available and relevant (tuned to the topic) is useful, as witnessed by the superior performance of the SFC in the USMED. Also, as shown above, the use of a combination of all the features (ALL) was significantly better for the T122 problem. This was true of C4.5-Rules and AQ15c which performed best with the ALL featureset. Our general conclusion is that these results reveal that the hybrid representation can be useful in profile learning.

Precision and Recall

The precision and recall results (Table 1) correspond fairly well with the predictive accuracy results. The best results (calculated as the sum of precision and recall) occur for the same feature sets as was found for predictive accuracy. The poorest profiles, however, were quite varied, with all of the featuresets except POL giving the

worst result at some point. The USMED SFC result shows in a rather dramatic way how the presence of a relevant generalization hierarchy was able to improve performance. To the extent that such comparisons are possible, it is worth noting that our scores for T122 can be compared with scores on T122 reported in the literature: [Schutze, Hull & Pedersen 1995, p. 235] report Non-Interpolated Average (NIA) Precision for T122 of 0.524 (using a non-linear neural net) and 0.493 (using a linear neural net). However, average precision is a different metric from NIA-Precision, and we did not compute the latter.

Learning Curves

An examination of the learning curves also revealed some interesting results. Normally one expects a learning curve to show a steady increase in performance as the percentage of training examples increases. However, except for the learning curve for the SFC dataset shown in Figure 4[1], the learning curves for profiles learned by AQ15c in the USMED problem are very unstable[2]. The presence of a generalization hierarchy while learning results in profiles which are predictively accurate and more stable than profiles learned from other featuresets. This suggests that the generalization hierarchy is providing a deeper understanding of the needs of the user and is more robust to the particular set of training examples currently used. Stability of learned profile performance is extremely important in achieving user trust in the automatically generated profiles.

Intelligibility of learnt profiles

A system which discovers generalizations about a user's interests can use these generalizations to suggest new articles. However, as mentioned earlier, in many practical situations, a human may need to validate or edit the system's learnt profiles. Intelligibility to humans then becomes an important issue. The following profile induced by AQ illustrates the intelligibility property. It shows a generalization (see Figure 2) from terminal vector categories contagious and genetic present in the training examples to medical.sci (i.e., medical science), and from the terminal category abortion up to medical.policy (medical policy).

IF *subject1 = nature or physical science &*
subject2 = nature or medical science or medical policy or human body
THEN article is of interest

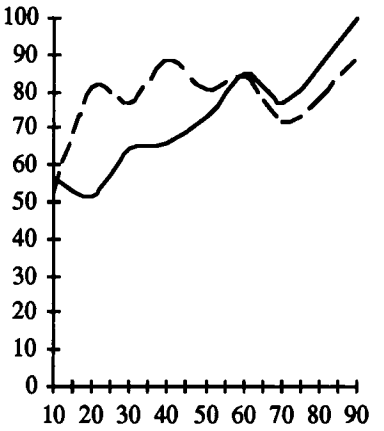

Figure 4. Precision (dotted line) and Recall (dark line) Learning Curve. (AQ15c using SFC features on the USMED dataset)

Although intelligibility is hard to pin down, there are various coarse measures of rule intelligibility that one can use. For one thing, one might assume that users prefer more concise rules. We examined profile length, measured as the number of terms on the left hand side of a learnt rule. Here we observed that using ALL the features led to more complex profiles over time, whereas using only subsets of features other than POL leveled off pretty quickly at profiles with well under 10 terms. The SFC profiles, which exploited generalization, were typically short and succinct. The tf.idf profiles were also quite short, but given their low overall performance they would not be useful.

Effect of Generalization Hierarchy

In our next set of experiments we tried to isolate the effects of the generalization hierarchy on the C4.5 learning algorithm by evaluating the performance of profiles learned from C4.5 with the hierarchical information (in the form of the extended attribute set) against C4.5 without the hierarchy (with the original x1..x5 attributes, but without the additional 18 attributes).

[1]Note that Figure 4 shows a graph of average precision and average recall versus the percentage of examples used in training. This is not to be confused with the typical precision/recall curves found in the information retrieval literature (e.g., (Harman 94, p. A5-A13)), which might, for example, measure precision and recall at different cutoffs.

[2]For reasons of space, the entire set of learning curves (Precision, Recall, and Predictive Accuracy learning curves for each of AQ15c and C4.5 on T122 and USMED, for each of POL, ALL, TF.IDF, and SFC) are not shown here.

Learning Problem	Generalization hierarchy attributes present ?	Predictive Accuracy		Average Precision/ Average Recall	
		SFC	ALL	SFC	ALL
USMED	No	0.46	0.76	0.47/0.23	0.89/0.67
	Yes	**0.79**	0.76	**0.97/0.60**	0.90/0.60
T122	No	0.68	0.73	0.58/0.55	0.64/0.74
	Yes	0.68	0.76	0.58/0.55	0.70/ 0.67

Table 2. The effect of generalization hierarchy attributes on predictive accuracy, precision and recall performance for C4.5-learned rules. Significant changes are boxed in thick lines, with the significant effect of generalization shown in boldface.

We found that the extension improved the performance significantly (99% confidence) for the USMED dataset and SFC feature set: predictive accuracy improved from 0.46 to 0.79 while precision/recall improved from 0.47/0.23 to 0.97/0.60. However, it did little to improve the performance for the other problem sets. These results are detailed in Table 2. With these additional attributes the best USMED results for both AQ and C4.5 was with the SFC generated attributes, and with the background knowledge of a generalization hierarchy. The best results for the T122 problem were obtained when all the generated features were available. This reinforces (with evidence from two learning algorithms) that our earlier conclusion that the hybrid representation is useful in profile learning, and that having a generalization hierarchy available and relevant (tuned to the topic) is useful.

Comparison with word-level Relevance Feedback Learning

Although our previous experiments had shown that machine learning methods learning from a hybrid document representation resulted in profiles which were predictively accurate and intelligible, they did not reveal if the traditional relevance feedback approach may not work just as well. In order to compare our results with a traditional relevance feedback method we applied a modified Rocchio algorithm to the two information retrieval tasks (USMED and T122) described earlier.

The modified Rocchio algorithm is a standard relevance feedback learning algorithm which searches for the best set of weights to associate with individual terms (e.g. tf-idf features or keywords) in a retrieval query. In these experiments individual articles are represented as vectors of 30,000 tf-idf features. Our Rocchio method is based on the procedure described in (Buckley, Salton, & Allan 1994). As before, the training involved induction of a new profile based on feedback from the pre-classified training examples, as follows. To mimic the effect of a user's initial selection of relevant documents matching her query, an initial profile was set to the average of all the vectors for the (ground-truth) relevant training documents for a topic. This average was converted from a tf.idf measure to a tf measure by dividing each tf.idf value by the idf. The profile was then reweighted using the modified Rocchio formula below. This formula transforms the weight of a profile term k from p-old to p-new as follows (Buckley, Salton, & Allan 1994):

$$\text{p-new}_k = (\alpha * \text{p-old}_k) + \left(\frac{\beta}{r} * \sum_{i=1}^{r} dw_{ik}\right) - \left(\frac{\gamma}{s} * \sum_{i=1}^{s} dw_{ik}\right)$$

r = number of relevant documents
s= number of non-relevant documents (all non-relevant documents)
dw_{ik} = tf weight of term k in document i
$\alpha = 8$ $\beta = 16$ $\gamma = 4$ (tuning parameters)

During testing, the test documents were compared against the new profile using the following cosine similarity metric for calculating the degree of match between a profile j (with the tf weights converted back to tf.idf weights) and a test document i (with tf.idf weights) (Salton & McGill 1983):

Learning Method	Predictive Accuracy USMED	T122	Average Precision/ Average Recall USMED	T122
Rocchio	0.49	0.51	0.52/0.53	0.39/0.27
Best AQ15c (SFC)	0.78	0.76	0.78/0.73	0.79//0.48
Best C4.5 (ALL)	0.76	0.73	0.90/0.60	0.64/0.74

Table 3 Comparing Predictive Accuracy, Average Precision / Average Recall for tf.idf terms

$$c_{ij} = \frac{\sum_{k=1}^{t} (dw_{ik} * qw_{jk})}{\sqrt{\sum_{k=1}^{t} dw_{ik}^2 * \sum_{k=1}^{t} qw_{jk}^2}}$$

t = total number of terms in collection
dw_{ik} = tf.idf weight of term k in document i, as before
qw_{jk} = tf.idf weight of term k in profile j

The cutoff for relevance was varied between 0 and 1, generating data points for a recall-precision curve. A best cutoff (which maximizes the sum of precision and recall) was chosen for each run. The results in Table 3 show that the machine learning methods represented by the best runs from AQ15c and C4.5 outperform the tf-idf based Rocchio method on both the T122 and USMED problems in terms of both predictive accuracy and predictive precision and recall. This performance difference may be due to an inability of the weighted term representation to accurate capture either the USMED or T122 profiles, or it may be due to the way term weights are learned. Further experiments will be necessary to pinpoint the cause for this performance difference.

Conclusion

These results demonstrate that a relevant generalization hierarchy together with a hybrid feature representation is effective for accurate profile learning. Where the hierarchy was available and relevant, the SFC features tended to outperform the others, in terms of predictive accuracy, precision and recall, and stability of learning. Other features and combinations thereof showed different learning performance for different topics, further emphasizing the usefulness of the hybrid representation. These results also confirm the suspicion that tuning a thesaurus to a particular domain will generally yield better learning performance. In this connection, the work of [Evans et al. 91a], [Evans et al. 91b] on thesaurus discovery and [Hearst and Schutze 93] on thesaurus tuning is highly relevant. In the latter work, thesaural categories extracted automatically from Wordnet [Miller et al 90] were extended with terms from a corpus. We can imagine a similar technique being used to augment the thesaurus used by the SFC.

Having assessed the basic performance of the profile learning capability, our next step will be to track the performance of the learner over time, where users of the IDD News Browser (information specialists in the MITRE Library) will have the option of correcting the induced profiles used to recommend new articles. For this to work, we will have to decide whether and how forgetting should take place. We also hope to allow the user to extend the representation space by defining new features, based, for example, on patterns seen in the learned rules, or user knowledge. We expect to touch on a number of specific user interface issues in the course of this work.

References

Armstrong, R.; Freitag, T.; Joachims, T.; and Mitchell, T. 1995. WebWatcher: A learning apprentice for the World Wide Web, In Proceedings 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, AAAI Press.

Belew, R. 1989. Adaptive Information Retrieval: Using a Connectionist Representation to Retrieve and Learn about Documents, ACM SIGIR, 11-20.

Belkin N. and Croft, B. 1992. Information Filtering and Information Retrieval: Two Sides of the Same Coin?, *CACM*, December 1992, 35, (12): 29-38.

Bloedorn, E.; Michalski, R. and Wnek, J. 1993. Multistrategy Constructive Induction: AQ17-MCI, In Proceedings of the Second International Workshop on Multistrategy Learning, 188-203.

Bloedorn, E. and Michalski, R. 1996. The AQ17-DCI system for Data-Driven Constructive Induction. In Proceedings of the International Symposium on Methodologies for Intelligent Systems. Forthcoming.

Broglio, J. and Croft, B. 1993. Query Processing for Retrieval from Large Text Bases. In Proceedings of Human Language Technology Workshop.

Buckley, C.; Salton, G. and Allan, J. 1994. The Effect of Adding Relevance Information in a Relevance Feedback Environment. ACM SIGIR 1994.

Foltz, P. and Dumais, S. 1992. Personalized Information Delivery: An Analysis of Information-Filtering Methods. *CACM* 35 (12):51-60.

Evans, D.; Hersh, W.; Monarch, I.; Lefferts, R. and Henderson, S. 1991a. Automatic Indexing of Abstracts via Natural-Language Processing Using a Simple Thesaurus", *Medical Decision Making* 11 (supp), S108-S115.

Evans, D.; Ginther-Webster, K.; Hart, M.; Lefferts, R. and Monarch, I. 1991b. Automatic Indexing using Selective NLP and First-Order Thesauri. In Proceedings of RIAO-91. 624-644.

Haines, D. and Croft, B. 1993. Relevance Feedback and Inference Networks. ACM SIGIR 1993.

Harman, D. 1992. Relevance Feedback Revisited. ACM SIGIR 1992.

Harman, D. 1994. Overview of the Third Text Retrieval Conference (TREC-3). Computer Systems Laboratory, National Institute of Standards and Technology.

Hearst, M. and Schutze, H. 1992. Customizing a Lexicon to Better Suit a Computational Task. In Proceedings of the ACL SIGLEX Workshop on Acquisition of Lexical Knowledge from Text, Columbus, Ohio.

Jones, S.; Gatford, M.; Robertson, S.; Hancock-Beaulieu, M. Secker, J. and Walker, S. Interactive Thesaurus Navigation: Intelligence Rules OK? *Journal of the American Society for Information Science*, 46 (1):52-59.

Lang, K. 1995. NewsWeeder: Learning to Filter Netnews. In Proceedings of the Twelth International Workshop on Machine Learning. 331-339.

Lashkari, Y.; Metral, M. and Maes, P. 1994. Collaborative interface agents. In Proceedings of the Thirteenth National Conference on Artificial Intelligence. AAAI Press.

Liddy, E. and Myaeng, S. 1992. DR-LINK's Linguistic-Conceptual Approach to Document Detection. In Proceedings of the First Text Retrieval Conference. Natl. Institute of Standards and Technology.

Liddy, E. and Paik, W. 1992. Statistically Guided Word-Sense Disambiguation. In Proceedings of the AAAI Fall Symposium Series: Probabilistic Approaches to Natural Language. Menlo Park, Calif.; American Association for Artificial Intelligence.

Maes, P. 1994. Agents That Reduce Work and Information Overload. *CACM*, 37 (7):31-40, 146-147.

Mani, I.; MacMillan T.; Luperfoy, S. Lusher, E. and Laskowski, J. 1993. Identification of Unknown Proper Names in Newswire Text. In Proceedings of the ACL SIGLEX Workshop on Acquisition of Lexical Knowledge from Text.

Mani, I. and MacMillan, T. 1995. Identifying Unknown Proper Names in Newswire Text. in J. Pustejovsky, ed., *Corpus Processing for Lexical Acquisition*, MIT Press.

Menczer, F.; Willuhn, W. and Belew, R. 1994. An Endogenous Fitness Paradigm for Adaptive Information Agents. In Proceedings of the CIKM'94 Workshop on Intelligent Information Agents.

Millet, G.; Beckwith, R.; Fellbaum, C.; Gross, D. and Miller, K. 1990. Introduction to WordNet: An on-line lexical database. *Journal of Lexicography*, 3 (4):235-244.

Mitchell, T.; Caruana, R. Freitag, D. McDermott, J. and Zabowski, D. 1994. Experience with a Learning Personal Assistant. *CACM* 37(7):81-91.

Pazzani, M.; Nguyen, L. and Mantik, S. 1995. Learning from Hotlists and Coldlists: Towards a WWW Information Filtering and Seeking Agent. In Proceedings of the AI Tools Conference.

Quinlan, J. 1992. *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Quinlan, J. 1995. personal communication.

Robertson, S. and Sparck-Jones, K. 1976. Relevance Weighting of Search Terms. *Journal of the American Society for Information Science* 27 (3):129-146.

Rocchio, J. 1971. Relevance Feedback in Information Retrieval. in *The SMART Retrieval System: Experiments in Automatic Document Processing*. 313-323, Prentice-Hall.

Salton, G. and Buckley, C. 1990. Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*, :88-297

Salton, G. and McGill, M. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

Schutze, H.; Hull, D. and Pedersen, J. 1995. A Comparison of Classifiers and Document Representations for the Routing Problem. ACM SIGIR 1995.

Sheth, B. 1993. A Learning Approach to Personalized Information Filtering. M.S. Thesis, Department of Electrical Engineering and Computer Science, MIT.

Sheth, B. and Maes, P. 1993. Evolving Agents for Personalized Information Filtering. In Proceedings of the Ninth IEEE Conference on Artificial Intelligence Applications.

Sparck-Jones, K. 1972. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation* 28 (1):11-20.

Wnek, J.; Kaufman, K.; Bloedorn, E. and Michalski, R. 1995. Selective Inductive Learning Method AQ15c: The Method and User's Guide. Reports of the Machine Learning and Inference Laboratory, ML95-4, George Mason Univ.