

Using NLP for Machine Learning of User Profiles¹

+*Eric Bloedorn, and +Inderjeet Mani

+Artificial Intelligence Technical Center, The MITRE Corporation
W640,1820 Dolley Madison Boulevard, McLean, VA 22102
{bloedorn, imani}@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 becomes increasingly important. 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. The research focuses on the importance of a suitable generalization hierarchy and representation for learning profiles which are predictively accurate and comprehensible. In our experiments 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 and the promise of combining natural language processing techniques with machine learning (ML) to address an information retrieval (IR) problem.

Keywords: information filtering, machine learning, natural language processing, generalization hierarchy

1. Introduction

As more information becomes available electronically, especially on the Internet, a number of new possibilities have arisen for finding information of interest. In addition to being able to retrieve documents of interest by sending a query to a Web search engine (such as AltaVista, Lycos, Excite, Infoseek), it is also possible to submit queries in advance and be notified about relevant articles as soon as they become available. Information filtering, as [3][13] point out, is an information access activity similar to information retrieval, where the queries (called profiles) represent evolving interests of users over a long-term period, and where the filters are applied to dynamic streams of incoming data. Information filtering systems build representations of the incoming data and match them against user profiles so as to decide whether the data is likely to be relevant to a particular user. There are many ways in which these profiles can be constructed. In some approaches the user constructs them directly, in others, the system tries to learn the user's profile by watching what the user does, or by soliciting feedback from the user or peer group. 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. To do so, we investigate an approach based on machine learning (ML). In particular, our work focuses on the problem of determining if a richer representation for text improves the accuracy of ML-generated user interest profiles, and if so which parts of that representation contribute to this improvement.

¹ This work was funded by MITRE under the MITRE Sponsored Research program.

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 profile learning, and the promise of combining natural language processing (NLP) techniques with machine learning to address an information retrieval (IR) problem. In what follows, we first motivate our work by discussing previous work related to (Section 2). In Section 3, we discuss the distinguishing features of our approach, explaining our method of representing text as well as the approach to learning. Section 4 describes the details of our experiments, and Section 5 summarizes.

2. Previous Work

There are a number of new information push services (e.g., PointCast, LA Times Hunter, ZDNet and Reference.com) providing a degree of information filtering. The PointCast system for example allows the user to build an interest profile from a predefined set of broad subject categories such as ‘sports’ and ‘weather’, as well as more specialized query terms, such as *NFL* or *NBA*. The FarCast system, like PointCast provides a set of predefined categories, but also allows the user to perform their own queries directly. NewsPage Direct uses Salton’s SMART system to provide personalized access to news. In this approach the user builds an interest profile from a set of over 2,500 topics ordered in a hierarchy. The user may browse the hierarchy while selecting leaf nodes to include in the profile.

All these systems, while providing much quicker access to relevant information than was earlier possible, suffer from the difficulties inherent in profile construction. It is often very difficult for the user to come up, without a great deal of trial and error, with a profile which captures their interests. The most common current approaches require the user to provide either a list of ‘key words’ or to select from a set of pre-defined categories. The key word approach is primarily a string-matching operation. If the string, or some morphological variant, is present, a match is made and the document is returned. String matching suffers from problems of polysemy, the presence of multiple meanings for one word (e.g. “Bank” as a financial institution, or part of a river), and synonymy, multiple words have the same meaning (e.g. “make”, “manufacture” and “produce” all refer to the production of items. In the categorical approach (e.g. Yahoo) humans provide the classification for each document. Both methods suffer from problems. The problem with this categorical approach is that it does not support interests that do not fit neatly into the pre-defined categories provided. For example, PointCast doesn’t allow the user to specify interests that fall across, or outside the predefined boundaries (e.g., *water sports*), or that are more specific (e.g., *NFL Central Division news*). If the provided categories don’t fit the user’s need, the customization provided is of little use.

In order to deal with the difficulty of coming up with suitable profiles, there are filtering systems which instead of requiring direct explicit definition of interest allow the user to specify one or more sample articles as reflective of her interests. These systems then notify the user when similar articles become available. Increasingly, such systems have started to use machine learning to discover users’ interests [9] [10] [23]. Filters have been constructed for many applications including web pages and USENET news, such as NewT [26], Webhound [22], WebWatcher [1], WebLearner [34], NewsWeeder [21], FishWrap [8] and Downtown (<http://www.incommon.com>). Downtown includes WiseWire which uses a neural network in a collaborative filtering mode to learn user interest profiles based on feedback from other users. There are also machine learning approaches which, though not focused exclusively on the information filtering task, demonstrate effectively how learning can be used to improve queries. Relevance feedback approaches, e.g., [2] [7] [15] [16] [37] [38] [40] [44], are a form of supervised

learning where a user indicates which retrieved documents are relevant or irrelevant. These approaches have investigated techniques for automatic query reformulation based on user feedback, such as reweighting terms in the query and query expansion (adding terms from positive examples to the query).

3. Our Approach

In all the above approaches, the system is unable to learn *generalizations* about the user's interests. Although clearly requiring additional knowledge and processing in acquiring and updating a hierarchy, a method for learning interest profiles which makes use of a generalization hierarchy has a number of advantages. For example, if a user likes articles on *scuba*, *whitewater rafting*, and *kayaking*, a learning agent 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. Thus, generalization could provide a powerful mechanism for capturing and communicating representations of user's interests, as well as justifying recommendations. More importantly the description provided by the generalization could help make the profile more intelligible to the user, which in turn could help establish trust. Of course, if the user's interest is highly specific, the ability to generalize may not be desired and could be better served by systems which specify articles based on the presence of specific strings.

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 [18] for detailed references), they have not been used to learn generalized profiles. In this paper, we describe a profile representation which exploits (together with other features) summary-level features, extracted from text using language processing techniques, linked to a generalization hierarchy from a thesaurus.

In addition to exploiting generalizations, the approach we have developed is designed to meet a variety of requirements. The approach must be scaleable to large collections of documents. A system using this approach should be able to quickly determine the user's needs through a combination of user provided information and feedback. The solicitation of feedback must not be intrusive. (In general, the amount of user interaction must be much less than would be required to directly construct profiles, or the approach won't be worthwhile.) Last, but not least, to be useful as a filtering system providing an intelligent 'push' capability, the system must be able to adapt as users' interests change.

In order to meet this latter requirement, some assumptions must be made about the user's interest. The most important assumption is that the user's interest is fairly stable. Without a stable interest, a learning approach which depends on past feedback will only hinder the retrieval of new information. This assumption, however, still allows the interest of the user to change slowly over time. (Interestingly, not much is known about how users' interests change, although tracking changing concepts is an area of increasing interest in machine learning [46] [27] [28].

We report on experiments evaluating the effect of various document representations on profile learning with a variety of learning algorithms. Our experiments were conducted in the context of a content-based profiling and summarization system for on-line newspapers on the World Wide Web, the IDD News Browser. In this system 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. Unlike the collaborative approaches described above, feedback is not pooled across users. Collaborative systems are well matched to problems of getting recommendations on subjects like movies or restaurants where the set of available object of interest is fairly small and the number of users is large. In the domain of news filtering neither the large number of like-minded individuals, nor a small set of objects under review is often available. News articles are updated frequently and thus most of the documents of interest are unreviewed by other potential advisors. For this reason the collaborative approach was not appropriate here.

3.1. Representing Text

A traditional way of representing text in the information retrieval community is in the form of long vectors of key words (filtered by various statistical techniques). Such representations, which have been strongly motivated by considerations of robustness in terms of extractability, often have a high dimensionality; for example, it is not uncommon to consider 10^5 or more features due to the large vocabulary of a text collection. Dimensionality reduction, e.g., techniques like Latent Semantic Indexing [11], has therefore been a long-standing issue in information retrieval. Also, a keyword vector does not provide us with a representation from which generalizations are possible. Now, one form of dimensionality reduction is to represent text in terms of classes of related words, labeled in some manner, rather than representing all the class members themselves. If these classes were hierarchically related, such classes could be used in generalization. In addition, it has been recognized, e.g., [36], that in relating words, it is useful to go beyond morphological relations (often approximated by stemming procedures), to capture relations of meaning. To do so, a representation should at least be able to group together synonyms and keep apart homonyms (i.e., a word having several unrelated meanings - e.g., *bank* the financial institution and river *bank*). However, it may be unwise to dispense with keyword approaches altogether, as such approaches have also been used very successfully in information retrieval.

We therefore decided to represent text in terms of classes of related words, in the form of subject categories from a thesaurus, and to use these along with a reduced set of keywords in a vector representation. In addition, based on recent findings by [6] on improved retrieval performance on TIPSTER queries using proper names, and to further strengthen profile intelligibility, we decided to exploit names of People, Organizations, and Places (POLs) in our representation. These are common in news texts, and can be robustly extracted from unrestricted text (e.g., MUC). We now discuss each of these representational features in turn.

3.1.1. Subject Categories

Our first set of summary level features assigns thesaural categories (or subjects) to segments of text based on the Subject Field Coder (SFC) [24] [25] (from TextWise, Inc.). In attempting to associate thesaural categories with texts, one well-known problem is that of word-sense disambiguation, in this

case deciding which of several thesaurus categories are the most likely ones for a term in the text. The SFC approach exploits evidence from local context (e.g., unambiguous words) along with evidence from large-scale statistics (e.g., pairwise correlations of subject categories) to disambiguate word sense, producing a vector representation of a text's subject categories.² SFC results 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 generalization hierarchy, which came to us from TextWise Inc.'s thesaurus consists of three levels. In this hierarchy the top level has 7 values, level 2 has 20 values and level 3 has 124 values. The SFC subject attributes provide values for individual documents at the lowest level (level 3) of the hierarchy. Although this hierarchy covers a fairly wide set of subjects as required for our newspaper application, it does not have a great deal of depth to allow users more specific profiles to be differentiated. To remedy this, we extended the set of terminal categories under one of the categories, **medicine**, to include another 16 lowest level categories. In Figure 1, we show a fragment of the extended hierarchy under **sci+tech** (scientific and technical).

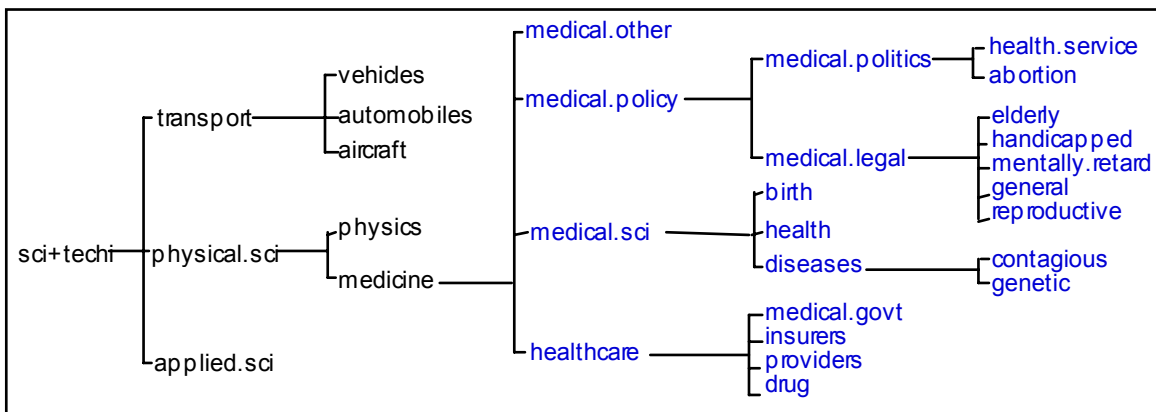


Figure 1. A fragment of the generalization hierarchy used in the experiments. New categories are shown in blue.

As our experience with the SFC hierarchy shows, the design and ability of the generalization hierarchy to be updated is of importance to the success of the method. Work on automatically constructing a hierarchy of concepts directly from data is extensive and includes work from a number of different research groups. One approach is to attempt to induce word categories directly from a corpus based on statistical co-occurrence [12] [14] [32]. Another approach is to merge existing linguistic resources such as dictionaries and thesauri [19] [20], or tuning a thesaurus (e.g., WordNET [33]) using a corpus [17]. Although there exists extensive research the problem of inducing clusters directly from data (e.g. [42] [43]) great difficulties remain in representing text and in insuring that discovered clusters are meaningful to the user. Our work here focuses on the problem of determining if a richer representation for text improves the accuracy of ML-generated user interest profiles, and if so which parts of that representation contribute to this improvement. Thus, although work on automated methods for constructing and updating the hierarchy are relevant they are outside the current scope.

² An earlier version of the SFC was tested on 166 sentences from the Wall Street Journal (1638 words) giving the right category on 87% of the words [24].

3.1.2. Keywords

Document keywords were also extracted by using a term-frequency inverse-document-frequency (tf.idf) calculation [41], which is a well-established technique in information retrieval. The weight of term k in document i is represented as:

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

Where 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. Features describe subjects (x1..x5), people (x6..x59), organizations (x60..x104) locations (x105..x140) and the top n statistical keywords (x140..x140+n), where n was varied from 5 to 200. Attributes (x1..x5) are linked to the thesaural generalization hierarchy.

3.1.3. Proper Names

We included features in our document representation involving terms relating to People, Organizations, and Locations (POLs) (along with their respective attributes). These features were provided by a name tagger [29] which classifies names in unrestricted news articles in terms of a hierarchy of different types of POLs along with their attributes (e.g., a person's title, an organization's business, a city's country, etc.) The name tagger combines evidence from multiple knowledge sources, each of which uses patterns based on lexical items, parts of speech, etc., to contribute evidence towards a particular classification³.

3.1.4. Summary-Level Representation

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

In short, these vectors correspond to an attribute value representation, where the domains of the subject attributes (x1..x5) are linked to a thesaural generalization hierarchy⁴. In comparison with a long keyword vector representation, this summary-level encoding allows for a more efficient representation, potentially reducing the dimensionality of the feature space used in learning. This is especially

³ In trials against hand-tagged documents drawn from collections of newswire texts, the tagger had an average precision-recall of approximately 85%.

⁴One problem with representing document subjects in this way is that the learning algorithms have a difficult time representing the simple concept of "one of x1..x5 has the subject y". We have begun testing a constructive induction approach based on AQ17-DCI [4] which appears to eliminate this problem and improve performance.

important when attempting to exploit symbolic learning algorithms, since long keyword vectors are ill-suited to current symbolic classification algorithms. Problems with applying symbolic approaches directly to domains which include continuous attributes, large numbers of attributes, and attributes with complex interaction (e.g. learning even and odd parity from a vector of bits) are well documented [5]. Thus, this compact representation allows the power of symbolic machine learning methods to be exploited.

Table 1. Description of attributes extracted from text.

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.

However, while this representation provides a great deal of information about an individual document and the subject attributes (x1..x5) reflect certain implicit associations between items, the representation is still quite coarse; for example, its attributes do not explicitly represent features of events, or other interesting associations between items, though such associations may be implicit in the data. (A somewhat richer associative representation for text is described in [30][31], where the representation is used for text summarization.)

3.2. Learning

Our representational decisions suggested some requirements for 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 [45] and C4.5-Rules [35] 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. AQ15c makes use of a generalization hierarchy in an attribute domain definition by climbing the hierarchy during rule construction.

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. The leaf nodes of tree are then pruned to prevent overfitting to the training data.

Another stated goal of this work is for a method which can automatically update learned profiles as the user's interest evolves. Work on developing methods for building symbolic learning algorithms capable of learning *evolving* concepts is also an active area [46] [27] [28]. Maloof ([27][28]) has developed a partial memory approach to this problem for the AQ15c program which appears to perform well on a variety of tasks with minimal memory requirements. Other work in this area could also be applied to the C4.5 learning system. One approach would be to use a window of most recent examples on which to train C4.5. Widmer ([48]) describes a variety of techniques for varying the size of the window to improve performance and for learning meta-rules for capturing repeating context. Incremental learning adds an additional layer of complexity to the problem that we are currently not addressing. However, we feel that the lessons learned on the static problem is an appropriate first step and will also be of use when dealing with changing user interest.

Discovering users' changing interests without requiring a large amount of feedback from the user is a difficult demand for any learning algorithm, but a common demand in this domain. Systems which are not capable of reasonable predictive performance based on little feedback will quickly lose the interest of the user, who will likely have little patience for training a new system. This differs from many IR problems which have large quantities of data available for generation of statistical correlations between words and documents. A method which makes use of a generalization hierarchy is one way of bootstrapping the system with the knowledge necessary to capture user interest quickly.

4. Some Experiments with this Approach

4.1. 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 summary level features used for generalization were useful in profile learning. Additionally it was important to determine if the representation used was useful for both a heuristic method (AQ) and a statistical method (C4.5 Rules). In order to make the generalization hierarchy available to both AQ15c hierarchical domain definitions for attributes the SFC extracted attributes x1 through x5 were provided as shown in Figure 1. For C4.5 the hierarchy was made available through an extended attribute set. In this extension, based on advise from Quinlan (Quinlan, 1992), we extended the attribute set to include attributes which describe nodes higher up on the generalization hierarchy. Additional attributes were added which provided the values of the subject attributes at each of the six levels higher in the tree from the leaf node. These experiments are also reported in [5].

Each of the experiments involved selecting a source of documents, vectorizing them, selecting a profile, partitioning the source documents into documents of interest to the user (positive examples) and not of interest (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 1. Precision and recall are standard metrics in the IR community, and predictive accuracy is standard in the ML community. The TFIDF scores are shown for $n=5$ (5 keywords); there was no appreciable difference for $n=200$

4.2. Metrics

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 in table 2.

Table 2. Description of metrics used.

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
Averaged Precision (Recall):	Average of Precision (Recall) over all test runs.
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

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.

4.3. Experiment 1: USMED

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 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. Feedback on only a small number of examples is quite typical of real-world applications 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”

Table 3. Predictive Accuracy, Average Precision, and Average Recall of learned profiles on USMED problem.

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	
C4.5-Rules	USMED	0.39	0.74	0.79	0.76	0.07/0.30	0.89/0.60	0.97/0.60	0.90/0.67	

The predictive accuracy/precision and recall results (Table 3) show that the most predictively accurate profiles generated (red) come from the SFC only feature set extended with the hierarchical domains. There was little difference in performance for C4.5-Rules between the SFC only feature set and the ALL feature set. In fact the precision/recall total for these two sets are equal. AQ15c shows a drop in performance with the larger feature set suggesting the additional attributes caused some degree of overfit to the data. This result shows that where the hierarchy is available and well suited to the task, the learners can find accurate descriptions of the user’s interest. Interestingly C4.5 Rules shows increasing performance as more complex level features are made available. With the TFIDF set, C4.5 Rules gives an average accuracy of only 39%, which increases to 74% when the POL set is used, 76% with the ALL set, and 79% when only the SFC set is used. AQ15c made better use of the available keywords in TFIDF, but did not perform as well with either the POL or ALL sets. All differences between the best and worst predictive accuracies (in blue) are significant to the 90% level and were calculated using a student t-test.

4.4. Experiment 2: TREC 122

As we have discussed in the introduction, the goal of this work is to develop a method whereby accurate, simple, comprehensible profiles may be generated with as little user feedback as possible. The previous experiment showed how our approach would operate with only 18 positive and 20 negative examples. Even this amount may be more than many users are willing to provide. This previous experiment, however is slightly unrealistic in its near balance of negative and positive examples. In this next example we use a more common situation where the number of positive examples is much fewer than the number of negative. This next experiment also exploited the availability of a standard test collection, the TREC-92 collection. The same generalization hierarchy used in the previous experiment was used here. 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”

Table 4. Predictive Accuracy, Average Precision, and Average Recall of learned profiles on T122 problem.

Method	Learning Problem	Predictive Accuracy					Average Precision/ Average Recall			
		TFIDF	POL	SFC	ALL	TFIDF	POL	SFC	ALL	
AQ15c	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	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	

The results of profile learning for the T122 dataset is shown in Table 4. While not showing the dramatic improvements possible with just SFC attributes and a generalization hierarchy as in the USMED problem, these results do show how summary level features alone (POL, SFC) and in combination (ALL) provide a useful representation for learning interest profiles. An examination of the rules shows that the ALL performance is based on the use of SFC and POL terms almost exclusively. AQ15c rules make no use of TFIDF attributes in when learning from the ALL set, and C4.5-Rules make use of TFIDF attributes only occasionally and in all of these cases uses the attribute only when the keyword is “company” or “patients”. It appears the combination of POL attributes such as company names (e.g. Warner Lambert, Cancer Institute) with general medical SFC categories as ‘Institutions, Science and Technology, Nature and Anatomy’) provided a useful base for discriminating the documents about Cancer research from others in this collection. Again, all differences between the best and worst predictive accuracies (in blue) are significant to the 90% level and were calculated using a student t-test.

These results suggest that profile learning using summary-level features (POL or SFC) alone or in combination with term-level features (ALL) provides useful information for characterizing documents. When a relevant generalization hierarchy is available, as witnessed by the superior performance of the SFC in the USMED, these summary features alone can be more predictively accurate than using only term-level features (TFIDF). When such background knowledge is not available, or when it is difficult to determine the specific class of the document (‘business and economics ‘ was the top subject assigned by the SFC to many of the positive examples in this experiment), the hybrid representation worked best. The ALL set was the best feature set for both C4.5-Rules and AQ15c for T122. Our general conclusion is that these results reveal that the hybrid representation can be useful in profile learning.

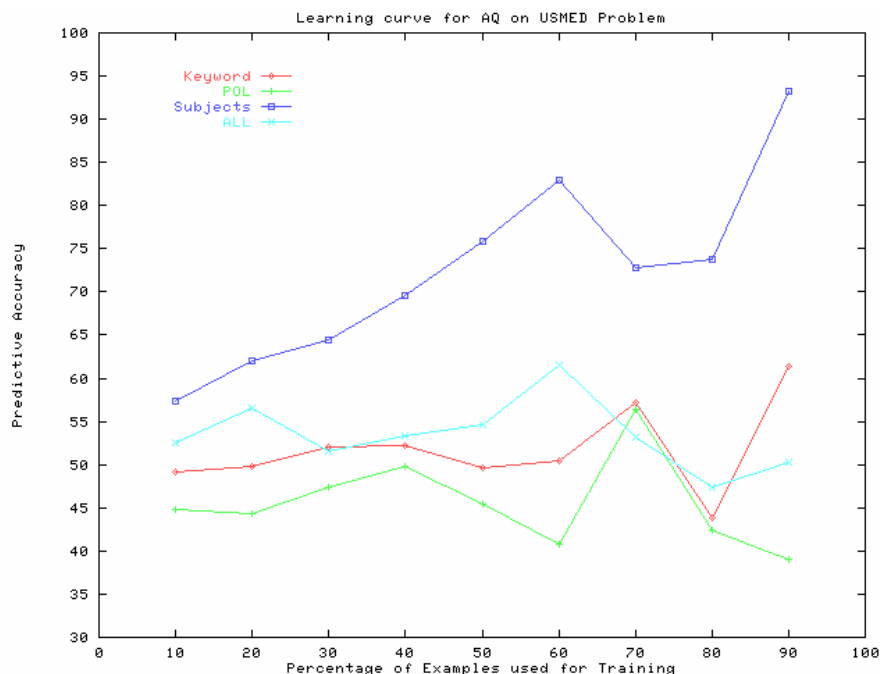


Figure 2. Predictive accuracy as a function of the percentage of examples used during training using KEYWORD, POL, SUBJECT and ALL feature sets (AQ15c on the USMED dataset).

4.5. 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 2⁵, the learning curves for profiles learned by AQ15c in the USMED problem are very unstable. The presence of a generalization hierarchy while learning results in profiles which are predictively accurate and more stable than profiles learned from other feature sets. 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.

4.6. 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 attributes representing the higher levels in the tree).

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 T122 problem sets, due probably to the difficulty of the SFC in assigning a single category to describe companies research on medical problems. These results are detailed in Table 5. 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.

Table 5. 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 red.

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

⁵Note that Figure 2 shows a graph of average predictive accuracy 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, which might, for example, measure precision and recall at different cutoffs.

4.7. Intelligibility of Learnt Profiles

The use of a generalization hierarchy is motivated by the need of the system to provide clear explanations for its decisions. These explanations allow the user to better trust the learning because the learned rules can be directly viewed and understood. They also allow the system to provide reasonable generalizations of the user's interest which it can then use to suggest new articles. Rules provide a knowledge representation that is intelligible to a wide audience as long as the features are clear and the rules are short. The following profile induced by AQ illustrates the intelligibility property. It shows a generalization from terminal vector categories contagious and genetic present in the training examples to medical science, and from the terminal category abortion up to medical policy.

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

C4.5 also produced many simple and accurate rules. One of the profiles generated by C4.5-Rules for the USMED dataset and ALL feature set shown below shows how C4.5 focused on the POL tags to come up with a simple description of the USMED articles in the training set.

```
IF POLtag5_location = US or POLtag1_honorific = Dr.  
THEN article is of interest
```

Although intelligibility is hard to quantify, 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 TFIDF profiles were also quite short, but given their low overall performance they would not be useful.

4.8. Comparison with Word-Level Relevance Feedback Learning

Relevance feedback has also been used as an effective method for building user interest profiles (or queries). Results by Salton ([39]) and Robertson and Sparck -Jones ([37]) as cited in Harman ([16]) "have shown very significant improvements in performance using relevance feedback for small test collections". 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 [7]. 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 [7]:

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

Where r = number of relevant documents, dw_{ik} = tf weight of term k in document I , s = number of non-relevant documents, and the tuning parameters $\alpha = 8$, $\beta = 16$, and $\gamma = 4$. 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) [41]:

Table 6. Comparing Predictive Accuracy, Average Precision / Average Recall for tf.idf terms.

Learning Method	Predictive Accuracy		Average Precision/ Average Recall	
	USMED	T122	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	0.76	0.97/0.60	0.70/0.67

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

Where t = total number of terms in collection, dw_{ik} = tf.idf weight of term k in document I , 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 6 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.

5. Summary

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.

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 will have the option of correcting the induced profiles used to recommend new articles. We also hope to investigate the use of constructive induction to automate the search for an improved representation [4].

References

1. Armstrong, R.; Freitag, T.; Joachims, T. and Mitchell, T., WebWatcher: A learning apprentice for the World Wide Web, in *Proceedings of the 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, AAAI Press, 1995.
2. Belew, R., Adaptive Information Retrieval: Using a Connectionist Representation to Retrieve and Learn about Documents, in *Proceedings of the*, Cambridge, MA, 11–20, 1989.
3. Belkin N. and Croft, B., Information Filtering and Information Retrieval: Two Sides of the Same Coin?, in *Communications of the ACM*, 35 (12), 29-38, 1992.
4. Bloedorn, E.; Michalski, R. and Wnek, J., Multistrategy Constructive Induction: AQ17-MCI, in *Proceedings of the Second International Workshop on Multistrategy Learning*, AAAI Press, 188-203, 1993.
5. Bloedorn, E., Mani, I., MacMillan, T.R., Representational Issues in Machine Learning of User Profiles. in *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, AAAI Press, Portland, OR., 433-438,1996.
6. Broglio, J. and Croft, B., Query Processing for Retrieval from Large Text Bases. in *Proceedings of Human Language Technology Workshop*, 1993.
7. Buckley, C.; Salton, G. and Allan, J., The Effect of Adding Relevance Information in a Relevance Feedback Environment, in *Proceedings of the Seventeenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, ACM, Dublin, Ireland, 1994.
8. Chesnais, P., Mucklo, M., and Sheena, J., The Fishwrap Personalized News System, in *IEEE Proceedings of the Second International Workshop on Community Networking Integrating Multimedia in the Home.*, IEEE, 1995.
9. Cohen, W., Text Categorization and Relational Learning, in *Proceedings of the Twelfth International Conference on Machine Learning*, Morgan Kaufmann, Lake Tahoe, CA, 1995.
10. Cohen, W. and Singer, Y., Context Sensitive Learning Methods for Text Categorization, in *Proceedings of the Nineteenth International Conference on Research and Development in Information Retrieval*, ACM, Zurich, Switzerland, 1996.

11. Deerweester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., and Harshman, R., Indexing by Latent Semantic Indexing. *J. Am. Soc. Inf. Sci.* 41(6), 391-407, 1990.
12. Evans, D.; Hersh, W.; Monarch, I.; Lefferts, R. and Henderson, S., Automatic Indexing of Abstracts via Natural-Language Processing Using a Simple Thesaurus”, *Medical Decision Making*, 11(supp), S108-S115, 1991.
13. Foltz, P. and Dumais, S., Personalized Information Delivery: An Analysis of Information-Filtering Methods. *in Communications of the ACM* , 35(12), 51-60. 1992.
14. Finch, S. and Chater, N., Learning Syntactic Categories: A Statistical Approach, in *Neurodynamics and Psychology* (M. Oaksford and G.D.A. Brown, eds.), Academic Press, Chapter 12, 1994.
15. Haines, D. and Croft, B., Relevance Feedback and Inference Networks, *in Proceedings of the Sixteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, ACM, Pittsburgh, PA, 1993.
16. Harman, D., Relevance Feedback Revisited., *in Proceedings of the Fifteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, ACM, Copenhagen, Denmark. 1992.
17. Hearst, M. and Schutze, H.. 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, 1992.
18. 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, 1995.
19. Klavans, J., Chodorow, M., and Wacholder, N., Building a Knowledge base from parsed definitions, *In Natural Language Processing: The PLNLP Approach.* (Jansen, K.; Heidorn, G.; and Richardson, S., eds.), Kluwer Academic Publishers, Chapter 11, 1992.
20. Knight, K., and Luk, Steve.. Building a Large-Scale Knowledge Base for Machine Translation, *in Proceedings of the Thirteenth National Conference on Artificial Intelligence*, AAAI Press, 773-778, 1994.
21. Lang, K., NewsWeeder: Learning to Filter Netnews, *In Proceedings of the Twelfth International Workshop on Machine Learning*, Tahoe City, CA, 331-339, 1995.
22. Lashkari, Y.; Metral, M. and Maes, P., Collaborative interface agents, *in Proceedings of the Twelfth National Conference on Artificial Intelligence.* AAAI Press, Seattle, WA, 1994.
23. Lewis, D., An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task, *in Proceedings of the Fifteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, ACM, Copenhagen, Denmark, 37-50. 1992.

24. Liddy, E. and Myaeng, S., DR-LINK's Linguistic-Conceptual Approach to Document Detection, in *Proceedings of the First Text Retrieval Conference*, Natl. Institute of Standards and Technology, Gaithersburg, MD, 1992.
25. Liddy, E. and Paik, W., Statistically Guided Word-Sense Disambiguation, in *Proceedings of the AAAI Fall Symposium Series: Probabilistic Approaches to Natural Language*. Menlo Park, Calif.; AAAI-Press, 1992.
26. Maes, P., Agents That Reduce Work and Information Overload, *Communications of the ACM*, 37(7), 31-40, 146-147, 1994.
27. Maloof, M.A., A Method for Partial Memory Incremental Learning and its Application to Computer Intrusion Detection, in *Proceedings of the Sixth International Conference on Tools with Artificial Intelligence*, 392-397, Washington, D.C., 1995.
28. Maloof, M.A., Progressive Partial Memory Learning, *Ph.D dissertation*. George Mason University, Fairfax, VA, 1996.
29. Mani, I. and MacMillan, T., Identifying Unknown Proper Names in Newswire Text, in *Corpus Processing for Lexical Acquisition* (J. Pustejovsky, ed.), MIT Press, 1995
30. Mani, I., and Bloedorn, E., Multi-document Summarization by Graph Search and Matching, in *Proceedings of the Fourteenth National Conference on Artificial Intelligence.*, AAAI Press, Providence, RI, 622-628, 1997.
31. Mani, I., and Bloedorn, E., Summarizing Similarities and Differences Among Related Documents, in *Proceedings of RIAO 97 (Computer Assisted Information Searching on the Internet)*, Centre de Hautes Etudes Internationales d'Informatique Documentaires, Montreal, Canada, 1997.
32. McMahon, J. and Smith, F., Improving Statistical Language Model Performance with Automatically Generated Word Hierarchies, *Computational Linguistics*, 22(2), 217-247, 1996.
33. Miller, G.; Beckwith, R.; Fellbaum, C.; Gross, D. and Miller, K., Introduction to WordNet: An on-line lexical database. *Journal of Lexicography*, 3 (4), 235-244, 1990.
34. Pazzani, M.; Nguyen, L. and Mantik, S., Learning from Hotlists and Coldlists: Towards a WWW Information Filtering and Seeking Agent, in *Proceedings of the Sixth International Conference on Tools with Artificial Intelligence*, 392-397, Washington, D.C., 1995.
35. Quinlan, J., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA, 1992.
36. Riloff, E., and Lehnert, W., Information Extraction as a Basis for High-Precision Text Classification, in *ACM Transactions on Information Systems*, ACM, July 1994. 12(3), 296-333, 1994
37. Robertson, S. and Sparck-Jones, K., Relevance Weighting of Search Terms. *Journal of the American Society for Information Science*, 27(3), 129-146, 1976.

38. Rocchio, J., Relevance Feedback in Information Retrieval, in *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall, 313-323, 1971.
39. Salton, G., *The SMART System*, Englewood Cliffs, N.J., Prentice-Hall, Inc., 1971.
40. Salton, G. and Buckley, C., Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*, 88-297, 1990.
41. Salton, G. and McGill, M., *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.
42. Stepp, R. and Michalski, R.S., Conceptual Clustering: Inventing Goal-Oriented Classifications of Structured Objects, in *Machine Learning: An Artificial Intelligence Approach* (R. Michalski, J. Carbonell, and T. Mitchell eds.), Morgan Kaufmann, 1986.
43. Stutz, J. and Cheeseman, P., AutoClass - a Bayesian Approach to Classification, in *Maximum Entropy and Bayesian Methods* (J. Skilling and S. Sibisi. Dordrecht eds.), Kluwer Academic, Cambridge, 1994.
44. Schutze, H.; Hull, D. and Pedersen, J., A Comparison of Classifiers and Document Representations for the Routing Problem. in *Proceedings of the Eighteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, ACM, Seattle, WA, 1995.
45. Wnek, J.; Kaufman, K.; Bloedorn, E. and Michalski, R., Selective Inductive Learning Method AQ15c: The Method and User's Guide, *Reports of the Machine Learning and Inference Laboratory*, ML95-4, George Mason University. 1995.
46. Widmer, G., On-Line Metalearning in Changing Contexts: MetaL(B) and MetaL(IB), in *Proceedings of the 3rd International Workshop on Multistrategy Learning (MSL96)*. AAAI Press, Menlo Park, CA, 1996.