Hybrid Hill-Climbing and Knowledge-Based Techniques for Intelligent News Filtering

Kenrick J. Mock
Intel Corporation
Intel Architecture Lab, JF2-76
2111 N.E. 25th Avenue
Hillsboro, OR 97124
mock@cs.ucdavis.edu

Abstract

As the size of the Internet increases, the amount of data available to users has dramatically risen, resulting in an information overload for users. This work involved the creation of an intelligent information news filtering system named INFOS (Intelligent News Filtering Organizational System) to reduce the user's search burden by automatically eliminating Usenet news articles predicted to be irrelevant. These predictions are learned automatically by adapting an internal user model that is based upon features taken from articles and collaborative features derived from other users. The features are manipulated through keyword-based techniques and knowledge-based techniques to perform the actual filtering. Knowledge-based systems have the advantage of analyzing input text in detail, but at the cost of computational complexity and the difficulty of scaling up to large domains. In contrast, statistical and keyword approaches scale up readily but result in a shallower understanding of the input. A hybrid system integrating both approaches improves accuracy over keyword approaches, supports domain knowledge, and retains scalability.

Content Areas: software agents
Abstract ID: A626
Word Count: 6728

Hybrid Hill-Climbing and Knowledge-Based Techniques for Intelligent News Filtering

Abstract

As the size of the Internet increases, the amount of data available to users has dramatically risen, resulting in an information overload for users. This work involved the creation of an intelligent information news filtering system named INFOS (Intelligent News Filtering Organizational System) to reduce the user's search burden by automatically eliminating Usenet news articles predicted to be irrelevant. These predictions are learned automatically by adapting an internal user model that is based upon features taken from articles and collaborative features derived from other users. The features are manipulated through keyword-based techniques and knowledge-based techniques to perform the actual filtering. Knowledge-based systems have the advantage of analyzing input text in detail, but at the cost of computational complexity and the difficulty of scaling up to large domains. In contrast, statistical and keyword approaches scale up readily but result in a shallower understanding of the input. A hybrid system integrating both approaches improves accuracy over keyword approaches, supports domain knowledge, and retains scalability.

Content Areas: software agents
Abstract ID: A626
Word Count: 6728

1. The Information Overload Problem

The goal of this project is to predict whether new news articles are likely to be of interest, or not of interest, based upon the prior behavior of the user. Systems that perform this type of intelligent behavior have recently been touted as intelligent "agents" (Riecken, 1994) by the media. The work proposed here follows the same vein; the system is intended to aid the user in her work rather than take over completely, watching and learning what the user does and what the user is interested in so that intelligent filtering may be performed. The filtering task is an extremely fuzzy and difficult problem to solve since users are notorious for their inconsistencies in behavior and interests. From a machine learning perspective, the problem is similar to trying to approximate a curve based upon discrete data points - except in this case, the function the machine is trying to approximate may change at any time.

To illustrate the filtering task, two sample news articles are shown in figure 1. The message headers indicate the author, subject, and newsgroups. The top message may be of interest to AI researchers, while the bottom message is a chain letter. Messages such as the chain letters are often targets readers wish to have filtered out. However, this is dependent upon individual user preferences, as some readers may be interested in chain letters.

Subject: AI, Genetic Algorithms and Artificial Life
From: ai@hyperion.ucdavis.edu
Newsgroups: comp.ai

>what is your work?
My latest research on an artificially evolved critter unifying all aspects
of Artificial Intelligence is available at http://phobos.cs.ucdavis.edu:8001.

Subject: Make Money FAST!
From: grog@netcom.com
Newsgroups: comp.ai, alt.life, alt.food.chocolate

This is a chain letter. Send $5 to everyone on the list, then add yourself.
Joe did and made $5,000,000. D0 NOT break this chain or a terrible fate
shall befall you! Noooo! :-)

Figure 1: Sample News Articles

To date, keyword-based filtering systems have been popular due to their simplicity. However, the performance of keyword systems ultimately suffers due to the "keyword barrier" as described by Mauldin (1991). The keyword barrier arises since keyword systems do not actually understand the semantic content of the input articles. On the other hand, understanding systems consider semantic content by processing articles in a manner similar to humans, breaking the keyword barrier and achieving higher performance. A middle ground may be achieved through a hybrid system that incorporates keywords and limited semantic knowledge.

The expected performance of this type of hybrid system is higher than a keyword system due to the incorporation of semantic knowledge, but lower than a purely knowledge-based system. However, the cost of the hybrid system is much lower than a purely knowledge-based system since the semantic knowledge is limited. The benefit of the hybrid system over a knowledge-based system is the scalability; typical knowledge-based systems require months or years of knowledge engineering, while the proposed hybrid system can be implemented immediately and scale up to large and varied domains.

2. Previous Work

Before a news article may be intelligently processed, the article must first be "understood" to some degree by the system. For information filtering, incoming articles must be understood well enough so that the content can be compared with the user model to determine if there is a match. A common assumption has been that articles need not be understood as well as a human reader in order to determine whether or not interest exists.

Typically, understanding is demonstrated by the extraction of key features from the text, or by providing a summary of the article. The easiest and most direct method of feature extraction is simply to pull *keywords* or *tokens* from the text that match a predefined set of words describing a user's interests or simply to use all of the words in the input article as features (Eberts, 1991; Jennings & Higuchi, 1992). Often, the words are first passed through a *stemmer*

and a *stop list*. A stemmer attempts to strip away word prefixes or suffixes to find the word root for comparison purposes. A stop list is a list containing common words that have no predictive value. These words are thrown out entirely. While the keyword/stemmer/stop list approach can be effective, it is difficult to predefine all relevant keywords that may occur in a text, or text may be worded in a manner that does not match a keyword. Since a keyword system has no semantic information, synonyms or similar concepts such as the words "car" and "automobile" are treated as separate entities.

After keywords have been scanned from news articles, a popular method of indexing the news document with the extracted terms is to use rule-based agents to model a user's usage patterns as in INFOSCOPE (Stevens, 1992), or to couple the term-frequency with the inverse-document frequency. This method is often referred to as *tf-idf* (Salton, 1991). These two terms are combined by multiplying the term-frequency (*tf*) by 1/document-frequency (*idf*) to obtain a metric of relevancy for each term. By combining terms from a document to form a vector, queries can undergo a similar process and the document vector closest to the query vector is retrieved as the best match. Mathematically, the weight of a term *t* with respect to document *i* is described by:

$$weight(t)_i = tf(t)_i \times idf(t)$$

Often the inverse document frequency is scaled to de-emphasize large weights by taking the logarithm of the frequency term *t* appears in all documents. Finally, the similarity of a query vector Q and a document vector V can be computed via the scalar product of the two vectors:

$$Similarity(Q,V) = \sum_t w(t)_q \times w(t)_v$$

To use the tf-idf method for information filtering, the tf-idf statistics are collected for an entire class of news articles. A simple two class system might include articles the user is interested in reading, and articles the user is not interested in reading. The similarity of a new article is computed by comparing it against classes instead of against individual articles, and the class most similar to the new article is used to predict the user's interest in the unread article. The NewT system (Sheth, 1994) is based upon tf-idf and genetic algorithms for news filtering. The tf-idf method is also compared against Lang's MDL method as a baseline for evaluation in NewsWeeder (Lang, 1995).

More human-like approaches to news understanding have been explored by DeJong (1982) and Mauldin (1991). The main advantage of a symbolic, knowledge-based approach is that the input text is understood as a human might understand the text, allowing for much greater understanding (Ram, 1992). One of the earliest knowledge-based approaches to news story understanding is the FRUMP system developed by DeJong . FRUMP is given UPI news stories, processes the story by comparing with stereotypical events through a structure named a script, and provides a summary of the article. Although not designed to filter news, FRUMP actually addresses a more difficult problem - that of story understanding.

A more recent work that also performs script based learning to understand and retrieve usenet news articles is Mauldin's FERRET system (Mauldin, 1991). In Ferret, a query and text

articles are parsed into Schank's Conceptual Dependency (CD) theory (Schank, 1977), which is intended to be an unambiguous representation for knowledge. In Ferret, once news articles and search queries are parsed into CD, predefined scripts are compared to the CD representations. As in FRUMP, the scripts represent stereotypical sequences of events and information. Articles that match the defined scripts may then be disambiguated with the script, classified in terms of their content, and matched with the query. The novel features in Ferret include an online dictionary to augment the understanding process and script learning through genetic algorithms.

Another method of information filtering which has recently attracted attention is collaborative filtering (often called social filtering). This involves the annotation or public review of articles by other users as the articles are read. The reviews then become an input for the text filter. As a result, other users may decide to read an article based upon the reaction of their peers; e.g., user A may choose to read articles only examined by user B or user C. Collaborative systems for filtering mail, usenet news, and WWW documents are currently under investigation (Brewer & Johnson, 1994; Lashkari et. al., 1994; Mock & Vemuri, 1994; Resnick et. al., 1994; Goldberg et. al, 1992).

3. Empirical Motivation for Usenet Information Filtering

Although a variety of systems have been created to filter news, an important basic question has remained unexamined: Is a filter even necessary? The users' current form of browsing may already give adequate performance. How many articles are users currently reading that they would prefer not to read? Conversely, how many articles are users not reading that they would like to read? The following study was conducted to answer these questions.

In this study, the classification of articles was compared when users browsed articles with a conventional news reader to situations when users were forced to read all articles. In a conventional news reader, users are given a list of articles in which the author and subject are displayed as shown in figure 2. While this format for displaying messages is useful for browsing, it can be difficult to find a specific article when there are hundreds or thousands of messages. Additionally, while the author and subject of a message give a good deal of information about the article, this information can sometimes be misleading since no information from the body of the article is displayed while browsing. A common phenomena is for the subject of messages to drift from the original subject as replies are made to the original message, resulting in messages containing the original subject but a completely different content. This experiment investigated whether or not this existing format for displaying messages provided sufficient information for users to pick messages of interest accurately.

a	G Demetriou	1 >Approximate string matching
b	Marvin Minsky Nancy Lebovitz	4 >AI Heaven
c	Richard Ottolini	2 >Does AI make philosophy obsolete?

Figure 2: Sample STRN browser screen, messages sorted by threads

3.1 Empirical Motivation - Experimental Method

The newsgroup selected for this study was the **ucd.life** newsgroup. This newsgroup was selected since all of the subjects in the study were UC Davis students and the newsgroup covers a variety of topics likely to be of interest to the general Davis community. This newsgroup receives moderate traffic (approximately 50 messages a day) so that filtering may be useful. The subject matter varied from want ad postings to crime discussions.

144 sequentially posted messages from the newsgroup were selected. These messages were sorted into threads and displayed to the user in the standard news reader fashion, providing the author and subject, as shown in figure 2. Subjects were first instructed to browse the articles as they normally would, and read those articles that looked interesting. After a subject read an article, the system asked the subject to classify the article as being **accepted** if she was glad she read the article and found it of interest, **rejected** if she really did not want to read the article, or **unknown** if she is unsure or ambivalent. In this manner, all of the articles the subjects decided to read during browsing were assigned a classification of accepted, rejected, or unknown.

After the browsing phase was complete and the subjects were satisfied that they had read all the messages they felt would be of interest, the subjects were instructed to read and classify every message. If the existing methods for displaying articles is sufficient and no filtering is necessary, then the message classifications during the browsing phase should closely match the message classifications from when all messages are read.

A total of 14 unpaid volunteer subjects participated in this study. All subjects were UC Davis students, 2 of them graduate and 12 of them undergraduate students. All subjects were familiar with existing news readers and had read the ucd.life newsgroup in the past. With the exception of the graduate students, the subjects were naive about the purposes of the experiment.

3.2 Empirical Motivation - Results

Collectively, 321 messages were classified as accepted or rejected by the subjects during the browsing phase. Of these 321 messages, 104 were classified differently when the subjects were forced to read all messages, resulting in a flip-flop rate of 32%. The percentage of messages classified as accepted during browsing is 11%, while the percentage of messages classified as rejected during browsing is 5% and the percentage of messages classified as unknown is 84%. 2016 messages were read during the read-all phase. 37% of these messages were marked as accepted, 41% as rejected, and 22% as unknown.

3.3 Discussion - Patterns of Behavior

During browsing, the subjects indicated that they were interested in only 212 messages. However, when the subjects read all messages, they indicated that they were actually interested in many more messages. A total of 697 were accepted, almost three times the number read during browsing. One explanation for these results is that displaying messages by author and subject alone do not provide enough information to allow users to pick the messages they would like to read accurately. Another explanation for the higher acceptance is that increased reading resulted

in increased interest. Reading some messages generated additional interest in similar messages, resulting in more messages classified as accepted.

To increase the chances that missed messages of interest are read, two direct approaches may be used. First, an intelligent filter could identify those messages likely to be of interest and alert the user. This works only as long as the user trusts the system and what messages the filtering system recommends. Second, the interface used for browsing could be improved to include content from the body of each message to give the user a better indication of what the message is about. This should allow readers to make a more informed choice of which message to read. In INFOS, this issue is partly addressed by displaying the first line of the text body in the browsing screen.

On the opposite spectrum of finding articles of interest is rejecting articles not of interest. With over 36% of the messages classified as rejected, this comprises a majority of the three classifications (35% accepted, 29% unknown). This volume of rejected messages indicates that the capability to recognize these articles will certainly aid the reader in selecting relevant articles.

3.4 Inconsistency of User Interests

One of the unexpected results of this study was the high number of flip-flops; subjects who classified a message one way during the browse phase, then later classified the message differently when all messages were read. Out of the 321 messages classified during browsing, 32% of them (104 messages) were changed during the read-all phase. One possible reason for this change is an increased user interest resulting from reading more articles, since the majority of flip flops were from negative to positive. Furthermore, subjects typically read very few messages during the browsing phase. This limited exposure to articles is not enough to gauge accurately what threads of conversations are about.

These flip-flops raise an issue about the maximum performance a filtering system can achieve. With 32% of the classifications changing, a very large error will result due to the fickleness of the readers. Many of these flip-flops stem from the limited number of articles initially read by the user. In an ideal setting, a user would only browse the messages she is interested in reading, and then the system will filter future articles based upon those articles read. However, this study indicates that users do not read enough browsed articles to build up a model of user interests accurately. Some of this error can be reduced by forcing users to read more messages. Nevertheless, in the end, any filtering system is subject to the whims and inconsistencies of the human user, making 100% accuracy virtually impossible to achieve.

These experimental results indicate that the current system of browsing results in many messages that users do not read, but would be interested in reading. Furthermore, the results indicate that users often change their mind about whether they like or dislike a particular article. A news filter would be a great aid in finding articles likely to be of interest that are normally missed, but the accuracy of such a filter will be limited due to human inconsistencies.

4. Global Hill Climbing Filtering Algorithm

The data used for the filtering experiments consisted of the same articles from the **ucd.life** newsgroup selected for the previous experiment. When processing articles, the extracted tokens were first passed through a stop list, but not through a stemmer. Additionally, binary encoded files were thrown out, extraneous header information stripped, and quoted material from old articles removed.

4.1 Global Hill Climbing - A Simple, Keyword Scheme

One of the requirements for the user model is that it must be very simple for users to modify and understand; if the model is too difficult to manipulate, the average user will never use it (Stevens, 1992). In addition to simplicity, the model must also provide for good performance. Consequently, a keyword/feature based system was initially selected for the user model since it is easy to perform computationally and also easy for users to understand.

Based upon the strengths and weaknesses of both Bayesian induction and tf-idf, a simple scheme has been implemented in INFOS which is inspired by both methods. This method, termed Global Hill Climbing, is a linear discriminant method based on a table of features. This table counts the number of times each feature has been found in each class. Since the table contains only one variable per class, it is simple for users to understand and manipulate. The table is created in a hill climbing fashion; as the user reads messages, she indicates whether or not each message read was accepted (liked) or rejected (disliked). The outcome is used to increment the table's weights accordingly.

An example is shown in Table 1. Here, the feature "genetic" has appeared in five accepted articles, the author feature of "grog@ucdavis" has appeared in three accepted articles and one rejected article, etc. This data indicates an interest in articles posted by grog or containing the word "genetic," and a disinterest in articles containing the word "flames." In addition to using words from the articles as features, collaborative review features are also included in the table. These other users are local users running the same news system who are willing to share their own reviews with others. In Table 1, the other user "Kiki" has accepted four articles the current reader has accepted, and Kiki has rejected one article the current user has accepted. Similarly, Kiki has rejected two articles the current user has accepted, and rejected three articles the current user has rejected. This table indicates that the current reader's accepted messages strongly correspond with Kiki's accepted messages, while the current user's rejected messages slightly correspond with Kiki's rejected messages. The table continues to grow as new articles are read.

Word	Accepted	Rejected
genetic	5	0
algorithm	3	3
flames	2	7
grog@ucdavis	3	1
Kiki Accepted	4	1
Kiki Rejected	2	3

Table 1: Global Hill Climbing Table of Weights

Given such a table, classification of new messages is performed by extracting the features from the new article and then computing the sum of all the Accepted and Rejected values from matching features in the table. If the Accepted percentage minus the Rejected percentage exceeds A, the message is classified as being of interest. Conversely, if the Rejected percentage less the Accepted Percentage exceeds A, the message is classified as being of no interest. Messages in between are marked unknown. In INFOS, A was set to 0.15 so that some margin of difference was necessary to classify a message either way. However, this has been left as a user-adjustable setting to allow more aggressive or more conservative classifications to be made. Mathematically, the classification process for a set of feature terms *t* is referenced by:

$$SimilarityPercentage(class)_t = \frac{\sum_t ClassOccurrences_t}{\sum_t TotalOccurrences_t} \quad (1)$$

$$Class_t = \left\{ \begin{array}{c} \left(SimilarityPerc(Acc)_t - SimilarityPerc(\mathrm{Re}\, j)_t\right) > A : Accepted \\ \left(SimilarityPerc(\mathrm{Re}\, j)_t - SimilarityPerc(Acc)_t\right) > A : \mathrm{Re}\, jected \\ else : Unknown \end{array} \right\} \quad (2)$$

The global hill climbing scheme bears some similarities to a Bayesian approach assuming conditional independence among the features. However, by computing sums, the frequency of occurrence for each feature is considered and a cutoff point is established. The system is closer in similarity to the tf-idf method, but it does not explicitly reference the inverse document frequency. However, this term contributes only a small amount compared to the tf term in tf-idf. Even if words that have a large index-document frequency enter into the global hill climbing table, those words that are non-predictive will still have little impact since the accept/reject probabilities will be approximately equal and cancel each other out. For example, the word “the” is not biased towards rejected or accepted articles, and although it will have a high frequency of occurrence, it will not be a factor in classification since both the rejected and accepted categories will contain approximately equal occurrences of the word. As a result, little is lost and the user profile is simple, making user modification of the profile an easy task.

4.2 Assigning Weights

As the algorithm stands, all features are treated equally. Authors, text from the body, text from the subject, and collaborative data are all counted and combined in the same way. While this allows each feature to account for as large or small a contribution as desired, this method is

biased to favor those features that occur most often. For example, the word "computer" is much more like likely to occur in the body of articles in a computer newsgroup, than the author of a particular group. The computer term may appear thousands of times, while an individual author will probably only appear a handful of times. As a result, the contribution from author's terms will be negligible when compared against other more frequently occurring features.

One solution to this problem is to separate the global hill climbing table into a set of individual tables - one table for each type of feature. Percentages of acceptance and rejection can be computed from the features among each table, and then these percentages combined to compute the final classification:

$$SimilarityCombn(Class)_t = \begin{array}{l} K_1 \times SimilarityPerc(Class)_{author} + K_2 \times SimilarityPerc(Class)_{sub} + \\ K_3 \times SimilarityPerc(Class)_{text} + K_4 \times SimilarityPerc(Class)_{collaborative} \end{array}$$

$$Class_t = \left\{ \begin{array}{c} \left(SimilarityCombn(Acc)_t - SimilarityCombn(\mathrm{Re}\, j)_t\right) > A: Accepted \\ \left(SimilarityCombn(\mathrm{Re}\, j)_t - SimilarityCombn(Acc)_t\right) > A: \mathrm{Re}\, jected \\ else: Unknown \end{array} \right\} \quad (3)$$

However, how should these percentages be combined? What values should be assigned to constants K_1 through K_4? Some systems (Jennings & Higuchi, 1992) give higher weight to the subject features on the assumption that these are most predictive. To investigate which terms are actually most predictive, experiments were performed to evaluate the impact of each feature individually. The features were then combined based upon how much impact they showed individually; i.e., the most predictive feature was given the highest weight, and the least predictive features given the lowest weights.

To test the feature's contribution to the classifications, 14 users read 100 sequentially posted messages from the ucd.life newsgroup and marked each as accepted, rejected, or unknown. From these 100 messages, 50 messages were randomly selected for training, and the system predicted the users' choices for the rest of the messages using equation 2 only among one set of features. These predictions were one of three classes: Suggested, Not Suggested, or Unknown. The predictions were then compared to the actual classifications provided by the subjects. The evaluation metric used in this experiment is classification accuracy. In INFOS, accuracy is defined as the percentage of predicted articles that were classified correctly.

The experimental results are shown in Table 2. The subject features results in the highest percentage correct (52%) with the lowest error (12%), probably since subject words are accurate predictors of entire threads that may be of interest. The textbody features actually give the largest percentage correct (54%), but also give the largest error (19%). Collaborative filtering gave the next best results (46%), and author alone was the worst predictor (38%), although not far behind the others. All schemes perform better than chance or by always predicting the most likely class.

Features Used For Classification	**Percentage Classified Correctly**	**Percentage Classified Unknown**	**Percentage Classified Incorrectly**

Author Alone	38.4	46.7	14.9
Subject Alone	52.1	35.5	11.8
Textbody Alone	53.6	27.2	19.2
Collaborative Alone	46.2	41.2	12.6

Table 2: Classification accuracy for individual sets of features.
Results are averaged over 14 subjects, showing percentage classified correctly, incorrectly, and unknown for 100 consecutively posted articles, 50 articles read.

The results from this experiment indicate that the subject features should have the highest weighting, followed by textbody and collaborative data. Author features should have the lowest weighting. A value of 0.35 was assigned to K_2, the subject's weight, 0.25 to K_3 and K_4, the collaborative and textbody weights, and 0.15 to K_1, the author's weight. Using these weights and equation 3, the classification process was rerun and the results shown below.

Percentage Correct Classifications	51.5%
Percentage Incorrect Classifications	7.3%
Percentage Unknown Classifications	40.9%
Within Error, Percent of False Positives:	50%
Within Error, Percent of False Negatives:	50%

The percentage of correct classifications, 51.5%, is slightly lower than using the subject scheme alone, but the error is significantly smaller at 7.3%.

5. Case-Based Reasoning Method

The global hill climbing method's main strength lies in its simplicity, user modifiability, and predictive abilities for features that have been previously encountered. In this case, INFOS builds a compact representation of user interests. However, the global method does have weaknesses. First, the global method is unable to discern fine differences in features because it linearly combines all input features through the conditional independence assumption; e.g., if we are not interested in messages with the features "dynamic" and "algorithms" but we are interested in messages with the features "genetic" and "algorithms", then the global method will be using the same accepted and rejected values for the word "algorithms" and may be unable to classify correctly these articles. Second, the global method has no semantic content about the meaning of words. The system will make separate table entries for the words "bicycle" and "bike" when these words are really referring to the same thing.

The method used in INFOS to address these problems is a case-based reasoning system. By retrieving individual cases and using the classification of those cases to classify new articles, the system is capable of avoiding the limitations of linearity. Furthermore, by designing a case-based reasoning system with semantic knowledge, INFOS is capable of comparing concepts rather than individual words. Finally, a CBR system also provides an excellent opportunity to support information retrieval of previously read articles in addition to information filtering.

5.1 Index Extraction

This work uses both controlled and uncontrolled index extraction as in the CLARIT system (Evans et. al., 1991). In the controlled approach, a predefined list of terms or knowledge structures is used to guide the indexing process. This approach can disambiguate concepts with high accuracy; however, one most have a fully defined knowledge base and predict all the structures that may occur. Currently, this is not possible for new domains. The uncontrolled approach relies on general purpose methods rather than pre-existing domain knowledge to create indices. As a result, indices may not be specific or well-defined as the controlled approach, but the benefit is generality across all domains. This project uses a combination of both approaches in an attempt to acquire the benefits both schemes offer. The controlled approach in INFOS is composed of a knowledge-based method derived from WordNet, while the uncontrolled approach is composed of a keyword-based inverted index using features such as unknown words, author names, or collaborative data.

INFOS uses WordNet (Miller, 1995) to map words into concepts, and these concepts are used as indices rather than the actual words. In the event that a word is missing from the WordNet lexicon, then that word is used in an inverted index to index the source document directly. To narrow the amount of data required for processing articles, INFOS only focuses upon the verbs and nouns indexed in WordNet.

WordNet is a project at Princeton University to create a knowledge-base of English words that includes part of speech identification, synonyms, frequency usage, etc. Concepts are defined in terms of a hierarchical semantic organization; e.g., the word “oak” is defined as a oak-->tree-->plant-->organism, where arrows indicate ISA relationships. Since the current version of WordNet (v1.5) contains approximately 107,000 noun senses and approximately 27,000 verb senses - the size of a paperback dictionary - WordNet is capable of recognizing terms from a broad variety of topics.

An example of the WordNet ISA hypernym hierarchy for the word “ocean” is shown in figure 3. When a word is found in the WordNet lexicon, all definitions or *senses* of that word are provided. In the case of ocean, there are two noun definitions; one for the body of water, and the other indicating a large quantity. These definitions are organized hierarchically, from the most specific up to more abstract concepts.

```
Sense 1
main, ocean, sea, briny
      => body of water, water
         => object, inanimate object, physical object
            => entity

Sense 2
ocean, sea
      => large indefinite quantity
         => indefinite quantity
            => measure, quantity, amount, quantum
               => abstraction
```

Figure 3 : Example WordNet hypernym hierarchies for the word "ocean."
This word has two sense definitions, organized from the specific to the general.

If INFOS indexed news articles based upon all the sense definitions of nouns and verbs found in an article, then a large number of irrelevant indices would be created due to multiple word meanings. Consequently, INFOS attempts to find appropriate noun or verb phrases based upon Paice's index extraction algorithm (Paice, 1989). filtering. This algorithm assumes that sentences repeat an underlying concept within a "topic neighborhood" of a few sentences. Those words occurring with a high frequency are likely to be relevant to the topic at hand.

Paice's algorithm was modified to operate upon WordNet word sense definitions rather than individual words. First, verbs and nouns from each sentence are identified through WordNet and their hierarchical definition referenced. This step results in a linked list of nodes, where each node contains the hypernym sense definition for the nouns and verbs in that sentence. Since each word is expanded into all possible sense definitions of that word, this pool of sense definitions may not accurately reflect the actual topic. For example, in the sentence "the ocean is cold.", both definitions of ocean from figure 3 will be expanded into the node list. However, only the body of water definition is relevant; the large-quantity definition does not apply. To refine the sense definitions and select relevant ones, neighborhoods of sentences are examined and the intersection of sense definitions that match within a specified neighborhood are selected. This process restricts the selected definitions only to those that are reoccurring topic stems and are then more likely to be relevant to the document. Only the first 20 sentences of articles were processed to speed execution in the event of extremely long postings. The assumption was also made that long messages will contain relevant material at the beginning of the article. Details of the algorithm may be found in Paice's work (1989).

After candidate nouns and verbs have been identified, this information is used to index the document. In addition to the sense definition itself as an index, other relevancy statistics are also associated to each term, including frequency and rarity (Evans et. al, 1991). *Frequency* is merely the number of times the term appears in the document / number of times the term has appeared in the domain. This measure operates upon the assumption that domain words appearing often are indicative of the document. *Rarity* is a measure of the expected frequency of a word in general English. In WordNet, this is obtainable through a terms polysemy count. A common word such as "system" has a high value of 15, while a rare English word such as "cilia" has a low value of 2. In INFOS, the rarity R is defined as:

$$R_{sense} = 1 - \left(\frac{Polysemy_{sense}}{Max\_Polysemy} \right)$$

Based on this definition, extremely rare words will have a value of 1, while common words will have a value closer to 0. Extremely common words will likely be filtered out by the stop list.

Once both frequency and rarity have been determined, the two are multiplied together to give a general relevancy statistic for a sense term:

Relevancy $R = Rarity \times Frequency$ for each term. (4)

The relevancy value is stored with each term and is used in memory retrieval to determine how closely an old article matches a new document.

5.2 Indexing of Cases

Once the appropriate noun and verb phrase senses have been extracted from a textual case, the article is saved and the senses used to index the case. When concepts cannot be extracted (e.g., with novel words or domain-specific slang words and expressions) these terms are used directly to index the case through an inverted index. In this way, cases are indexed via both conceptual (controlled) and keyword (uncontrolled) vocabularies, allowing conceptual retrieval when possible, and also keyword retrieval under unforeseen situations so that performance is still possible (Callan & Croft, 1993).

The method in which articles are indexed using the sense definitions is to construct a pointer to the file that contains currently defined sense in a global abstraction hierarchy. An example memory hierarchy with three cases is shown in figure 4. In this example, one article contains the word "vehicle," another article contains the word "bicycle," and the last article contains the word "car."

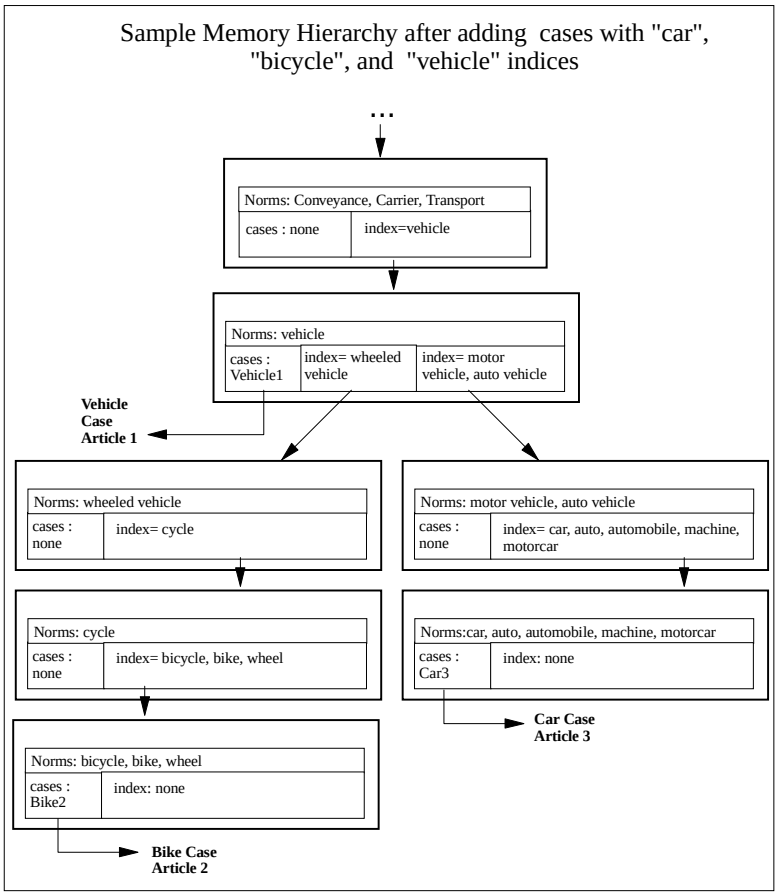

Figure 4: Sample Memory Hierarchy for Indexing Cases

In figure 4, the root node is not shown, but the sub-hierarchy starting at the Conveyance concept is displayed. This node represents the concept regarding items of transport and conveyance. All sub-nodes inherit the norms of their ancestors, hence all nodes located below this must also refer to transportation vehicles. One specific index currently exists from the Conveyance node, and it points to the Vehicle node. In turn, the Vehicle node points to sub-nodes, one regarding cycles and another regarding automobiles. In addition to pointing to sub-nodes, the Vehicle node also has an index to a specific case (news article) referencing vehicles. In a similar fashion, indices from the wheeled vehicle and the auto nodes are further specialized until they too point to actual cases referencing those concepts.

5.3 Memory Retrieval

Case-based memory retrieval involves searching for applicable cases based upon a given set of features. These features are simply WordNet sense indices from a new article that needs to be classified. Case retrieval in INFOS is fairly simple. A depth-first search is performed in the memory abstraction hierarchy along indices that match the input query until cases are retrieved or there are no more input indices to follow. To allow for partial matches (e.g., retrieve cases regarding bicycles when the input is about cars), path mismatches in the hierarchy can be traversed until an error threshold value is exceeded. In INFOS, 85% of the concepts had to match for a case to be retrieved.

For each case that is retrieved, the overall match value for that case is computed by summing over all *n* feature queries the following distance function:

$$Match = \frac{1}{n}\sum_{i=1}^{n}(MatchPercent_i) \times \mathrm{Re}levancy_i \qquad (5)$$

In INFOS, the retrieved cases are sorted by degree of match. The classification statistics of the best matching case can then be used to classify the new article, using the Accepted and Rejected counters for the case and computing a classification via equation 2. The article number can also be displayed as a justification to the user to indicate why INFOS believes new articles should be classified similarly.

5.4 Results of Case-Based Scheme

The same tests and testing methodology that was used to evaluate the global hill climbing scheme was also run with the case-based scheme. Finally, the case-based scheme was tested when used in conjunction with the global scheme. In this mode, the global scheme classification was performed first. If the global scheme returned an unknown classification, then the classification of the case-based scheme was used. The global scheme was performed first because it was found to have a lower error rate, and is also quicker than the CBR method. If a simple method can provide the correct classification, it is reasonable to use that method before more complex ones are attempted. The overall results indicate that the semantic and keyword based filtering provided by the combined best match CBR and global hill climbing scheme

performs best. The hybrid method improved the correct classification percentage over the global hill climbing method alone.

A summary of the results is shown in Table 3 depicting the Global Hill Climbing method (GHC), CBR alone using the best match to classify of the most highly ranked case as the classification, and the CBR scheme combined with the global hill climbing method. The combined CBR scheme using best match with global hill climbing performed best, although it had a slightly higher error rate than the global hill climbing method alone.

Classification Method	Percentage Classified Correctly	Percentage Classified Unknown	Percentage Classified Incorrectly	Percent False Positives	Percent False Negatives
Global Hill Climbing (GHC)	51.5	40.9	7.3	50	50
CBR - Best Match as Class	39.8	50.5	9.5	77	33
Combine GHC + CBR Best Match	58.0	29.9	12.1	62	37

Table 3: Classification accuracy for Hill Climbing, CBR, and Hybrid Methods

The results from this experiment indicate that the global hill climbing method still has the lowest error but the combined scheme provides the best correct classification rate. The case-based scheme will have some poor indices due to the sense disambiguation problem that can allow irrelevant cases to be retrieved. Consequently, the CBR method has a higher error rate than the global hill climbing method. When combined with the global hill climbing scheme, the best match CBR method does achieve a higher correct classification percentage at 58%, although it suffers from a slightly higher error rate of 12%.

6. Future and Ongoing Work

In addition to the hybrid methods of global hill climbing and case-based reasoning via WordNet, ongoing work with INFOS also incorporates genetic algorithms to explore the news space, and index patterns or scripts to parse input articles more accurately. Other areas of proposed work include modifications for INFOS to run offline, a graphical user interface, self-modifying parameters, the incorporation of other knowledge bases such as CYC (Lenat, 1995), and the application of filtering to the World Wide Web and intelligent tutoring systems. Additional information is available from the WWW at:

http://phobos.cs.ucdavis.edu:8001/~mock/INFOS/infos.html

7. Summary

As the information age grows in scale, the amount of incoming data becomes too large for humans to handle. The internet has been growing a tremendous rate. Gigabytes of news

articles flow through the internet daily, and World Wide Web pages number around 10 million. The central issue in this work addresses methods to model user interests automatically so that this data, usenet news articles in particular, can be filtered intelligently. However, in order to be a useful tool, the user model must be capable of adapting to user interests, articles must be displayed to give as much information as possible so users can intelligently browse and select articles to read, users must be capable of modifying and understanding the user model constructed for them, and the news filtering system must give accurate predictions.

Knowledge-based systems have the advantage of analyzing input text in detail, but at the cost of computational complexity and the difficulty of scaling up to many domains or domains of large scale. In contrast, statistical and keyword approaches scale up readily but are limited to a shallower understanding of the input. A hybrid system implemented in INFOS that integrated a keyword hill climbing method with a case-based reasoning method improved classification over the keyword method and provided scalability along with domain knowledge. However, the case-based approach did introduce some additional error due to the lack of robust disambiguation.

8. References

Brewer, R.S. & Johnson, P.M. (1994). Toward Collaborative Knowledge Management within Large, Dynamically Structured Information Systems. *Internal Research Report, Collaborative Software Development Laboratory, Department of Information and Computer Sciences, University of Hawaii*. WWW: http://www.ics.hawaii.edu/~csdl/urn.

Callan, J.P. Croft, W.B. (1993). An Approach to Incorporating CBR Concepts in IR Systems. *Proceedings of the 1993 Spring Symposium on Case-Based Reasoning and Information Retrieval*, AAAI Press, pp 28-32.

DeJong, G. (1982). An Overview of the FRUMP System. In W.G. Lehnert & M. H. Ringle (Eds.), Strategies for Natural Language Processing, Hillsdale, NJ: Lawrence Erlbaum, pp. 149-174.

Eberts, R. (1991). Knowledge Acquisition Using Neural Networks for Intelligent Interface Design. *Proceedings of the 1991 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1331-1335.

Evans, D.A. Ginther-Webster, K. Hart, M. Lefferts, R.G. Monarch, I.A. (1991). Automatic Indexing Using Selective NLP and First-Order Thesauri. *Proceedings of the Intelligent Text and Image Handling Conference,* Barcelona, Spain. pp 624-643.

Goldberg, D., Nichols, D., Oki, B., Terry, D. (1992). Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, **35** (12), pp. 61-70.

Jennings, A. & Higuchi, H. (1992). A Personal News Service Based on a User Model Neural Network. *IEICE Transactions Inf. & Systems*, **E75** D(2), pp. 198-209.

Lang, K. (1995). NewsWeeder: Learning to Filter Netnews. *Proceedings of the Twelfth International Machine Learning Conference.*

Lashkari, Y., Metral, M., & Maes, P. (1994). Collaborative Interface Agents. *Proceedings of the Twelfth National Conference on Artificial Intelligence,* pp. 444-449.

Lenat, D.B. (1995). CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM*, **38** (11), pp. 32-38.

Mauldin, M. L. (1991). Conceptual Information Retrieval: A case study in Adaptive Partial Parsing. Kluwer Academic Publishers. Norwell, MA.

Miller, G.A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, **38** (11), pp. 39-41.

Mock, K., Vemuri, V. (1994). Adaptive User Interface for Intelligent Information Filtering. *Proceedings of the Third Golden West International Conference on Intelligent Systems*, pp 506-517.

Paice, C.D. (1989). Automatic Generation and Evaluation of Back-of Book Indexes. *Prospects for Intelligent Retrieval, Informatics 10*, Cambridge MA.

Ram, A. (1992). Natural Language Understanding for Information-Filtering Systems. *Communications of the ACM*, **35** (12), pp. 80-81.

Resnick, P., et al. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *Internal Research Report, MIT Center for Coordination Science.*

Riecken, D. (1994). Intelligent Agents. *Communications of the ACM*, **37** (7), pp. 18-21.

Salton, G. The SMART Retrieval System: Experiments in automatic document processing. Englewood Cliffs, NJ: Prentice-Hall, 1971.

Schank, R.C. & Abelson, R. (1977). Scripts, Plans, Goals, and Understanding. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Sheth, B.D. (1994). A Learning Approach to Personalized Information Filtering. Masters Thesis. Department of Computer Science and Engineering, Massachusetts Institute of Technology.

Stevens, C. (1992). Knowledge-Based Assistance for Accessing Large, Poorly Structured Information Spaces. Ph.D. Dissertation Thesis. Department of Computer Science, University of Colorado.