Pergamon

Information Processing & Management, Vol. 33, No. 5, pp. 633–644, 1997

0306-4573/97 $17+0.00

PII: S0306-4573(97)00022-8

INFORMATION FILTERING VIA HILL CLIMBING, WORDNET, AND INDEX PATTERNS

KENRICK J. MOCK[1]* and V. RAO VEMURI[2]

[1] Department of Computer Science, University of California at Davis, Davis, CA 95616, USA and
[2] Department of Applied Science, University of California at Davis, Livermore, CA 94550, USA

Abstract—The recent growth of the Internet has left many users awash in a sea of information. This development has spawned the need for intelligent filtering systems. This paper describes work implemented in the INFOS (Intelligent News Filtering Organizational System) project that is designed to reduce the user's search burden by automatically categorizing data as relevant or irrelevant based upon user interests. These predictions are learned automatically based upon features taken from input articles and collaborative features derived from other users. The filtering is performed by a hybrid technique that combines elements of a keyword-based hill climbing method, knowledge-based conceptual representation via WordNet, and partial parsing via index patterns. The hybrid system integrating all these approaches combines the benefits of each while maintaining robustness and scalability.

1. PREVIOUS WORK

The Intelligent News Filtering Organizational System (INFOS) is designed to reduce a user's search burden while browsing a large number of messages. While some filtering systems attempt to summarize input articles or extract key data (Soderland & Lehnert, 1994), the goal of INFOS is to determine whether or not the user is interested, or not interested, in a particular article. For example, a news stream may contain many news articles and INFOS's job is to determine which articles out of the data stream the user may wish to read. The criteria for determining what a user finds interesting varies greatly among individuals, implying that the system must be capable of adapting to variable interests.

One of the most popular methods implemented in filtering systems is the extraction of *keywords* (also referred to as *tokens*) from the text to use as features for classification. Some systems watch for predefined keywords that match an action rule (Stevens, 1992) while others use all words in the input article as features after passing through a stemmer or stop list (Eberts, 1991; Jennings & Higuchi, 1992).

Once keywords have been extracted, a widely used method for classification is *tf-idf* (Salton, 1971). In tf-idf, the *Term Frequency* is coupled with the *Inverse Document Frequency* to provide a metric of relevancy between documents based upon the frequency of occurrence of each feature in an individual document and among all documents. By combining terms from a document and a user model (or query) to form a vector, the document vector closest to the user model vector is retrieved as the best match to classify the input document. Tf-idf is the classification method implemented in NewT (Sheth, 1994) and is also compared against Lang's MDL method in NewsWeeder (Lang, 1995).

DeJong (1982) and Mauldin (1991) have explored knowledge-based approaches to news filtering that are based upon conceptual processes. These approaches attempt to understand the input text in a similar fashion to people, allowing for much greater possibilities than keyword

* To whom all correspondence should be addressed at: Intel Corporation, JF2-76, 2111 NE 25th Ave., Hillsboro OR 97124, USA. E-Mail: Kenrick_J_Mock@ccm.jf.intel.com

systems (Ram, 1992). Both DeJong's FRUMP and Mauldin's FERRET systems are based upon Conceptual Dependency (CD) theory (Schank & Abelson, 1977), which is intended to be an unambiguous representation of knowledge. The systems parse input articles into CD and compare the information with *scripts*, or stereotypical sequences of events, to determine the content of the article.

In addition to filtering techniques based upon article keywords, filtering may also be performed using indirect features. Social or collaborative filtering is a recent area of activity that applies an indirect path of classification to arbitrary types of media (Brewer & Johnson, 1994; Lashkari *et al.*, 1994; Mock & Vemuri, 1994). In collaborative filtering, a population of users read and classify articles. The classifications are made public. The reviews then become an additional input for filtering. As a result, other users may decide to read an article based upon the reviews of their peers.

2. SYSTEM OVERVIEW

The goal of INFOS was to create a usable Usenet news filter with sufficient performance to ease a user's burden during browsing. To this end, INFOS is capable of learning from both user feedback and direct user manipulation. The effectiveness of INFOS was examined via user-testing and comparisons with traditional IR techniques.

From a user perspective, INFOS automatically builds a profile of user interests based upon active feedback and uses the profile to predict if new articles will be of interest. A user begins by browsing through articles. After each article has been read, INFOS asks the user to rate the article as Accepted if the user liked the article, Rejected if the user disliked the article, and Unknown if the user was unsure. The next time the user returns to read messages, new articles will be sorted and marked according to induced interests.

From an architectural perspective, a summary of the filtering and classification process is depicted in Fig. 1. To classify new articles, a simple keyword method named Global Hill Climbing (GHC) is used first. The results discussed in Section 3 indicate that this approach has a low error rate but a fairly large proportion of unknown classifications. However, GHC is useful as a quick and simple first-pass method. Next, GHC is used in conjunction with a Case-Based Reasoning (CBR) module. While considerably slower and more complex than GHC, CBR supports more powerful classifications. Consequently, if the global method returns an unknown classification, then the CBR module is invoked. Details of both methods follow in Sections 3 and 4. After new messages both the GHC and CBR models are updated to reflect user preferences.

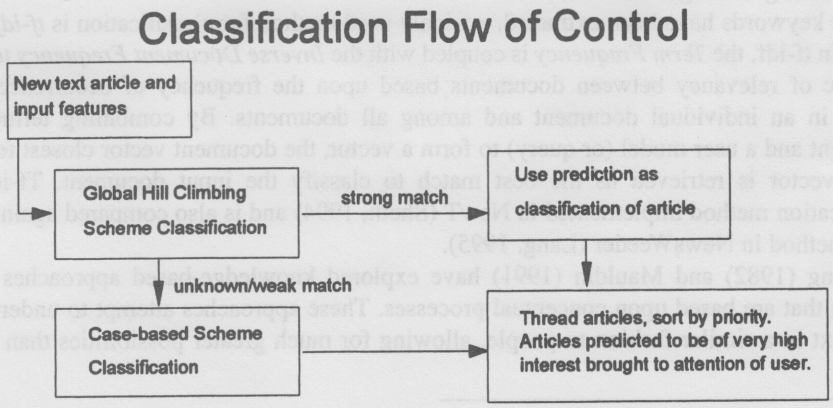

Fig. 1. Classification flowchart.

Table 1. Global hill climbing table of weights

Word	Accepted	Rejected
Agent	8	0
Flames	2	7
mock@intel.com	3	3
Crisha accepted	6	1
Crisha rejected	2	3

3. FILTERING VIA GLOBAL HILL CLIMBING

As a result of usability studies (Mock, 1996), an important requirement for filtering systems is that the user model must be very simple for users to understand. Moreover, the interests of users change quickly, implying the need for user-level filter control. If the model employed in the filtering system is too difficult to change or understand, the typical reader will never use it (Stevens, 1992). For example, a neural network model (Eberts, 1991) is almost impossible for a user to change. Tf-idf is simpler, but naïve users may have difficulty understanding the inverse relationship between the term frequency and document frequency terms. In psychological experiments, subjects naturally prefer negation/addition strategies over reciprocal/inverse strategies (Collis, 1980). The approach taken in INFOS has been to design a system that can be easily modified with a linear control for weighting each term while also supporting adequate filtering performance.

The method implemented in INFOS is a linear discriminant scheme based on a table of features and is termed Global Hill Climbing (GHC). A summary of GHC is described below and details may be found elsewhere (Mock, 1996). A table counts the number of times each feature has been found in each class, where the possible classes are 'Accepted' and 'Rejected'. Since the table contains only one variable per class, it is simple for users to understand and manipulate. The table is created via hill climbing with intrusively collected data. As the user reads messages, she is prompted by INFOS to rank each message read as Accepted, Rejected, or Unknown. The outcomes are then used to increment the table's weights accordingly.

An example of a GHC table is shown in Table 1. The word 'agent' has been extracted from 8 articles as a feature, and all 8 times the article has been marked as accepted. Similarly, the author feature of 'mock@intel.com' has been marked accepted 3 times and marked rejected 3 times. This data indicates that the author 'mock' may not be particularly predictive, but articles containing 'agent' are likely to be of interest. Conversely, articles containing 'flames' are likely to be of no interest. In addition to using words from the articles as features, other types of features such as collaborative data may also be included. This data is collected from other users running the same news system who are willing to share their own reviews with others. In Table 1, user 'Crisha' has accepted 6 articles that the current reader has accepted, and she has rejected 1 article the current user has accepted. Similarly, Crisha has rejected 2 articles the current user has also accepted, and rejected 3 articles the current user has rejected. This table indicates that the current reader's accepted messages strongly correspond with Crisha's accepted messages, while the current reader's rejected messages may not correspond with Crisha's rejected messages. As new articles are read, the table grows and matches user interests more closely.

Once the GHC table is constructed, 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 subtracted from the Rejected percentage exceeds a threshold value A, the message is classified as being of interest. Conversely, if the Rejected percentage subtracted from the Accepted percentage exceeds A, the message is classified as being of no interest. Messages in-between are marked unknown. A was set to 0.15 to allow some margin of difference necessary to classify a message, but this value is user-adjustable. The classification process for a set of feature terms t is expressed mathematically by:

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

$$Class_t = \left\{ \begin{array}{c} (SimilarityPerc(Acc)_t - SimilarityPerc(Rej)_t) > A{:}Accepted \\ (SimilarityPerc(Rej)_t - SimilarityPerc(Acc)_t) > A{:}Rejected \\ else{:}Unknown \end{array} \right\} \quad (2)$$

If only Equations (1) and (2) are used for classification, then 'authors', 'text from the body', 'text from the subject', and 'collaborative' features will all be counted and combined identically. A result of this method is a bias toward features that occur most often. This may not be desired; e.g. the word 'agent' may appear frequently in the body of an article, but the authors' names will only appear once. However, the 'agent' feature is not necessarily more important than the author simply because the structure of a news article dictates that it appear more often.

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

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

$$Class_t = \left\{ \begin{array}{c} (SimilarityCombn(Acc)_t - SimilarityCombn(Rej)_t) > A{:}Accepted \\ (SimilarityCombn(Rej)_t - SimilarityCombn(Acc)_t) > A{:}Rejected \\ else{:}Unknown \end{array} \right\} \quad (3)$$

What values should be assigned to constants K_1 through K_4? Some systems (Jennings & Higuchi, 1992) assume the subject features are most predictive and should have the highest weight. 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 the individual impact.

The data used for the experiments consisted of news articles randomly selected from the *ucd.life* newsgroup. This newsgroup was selected since it covers a variety of general topics relevant to the test participants (all university students). When processing articles, words from each article were passed through a stop list, binary encoded files thrown out, and quoted text from old articles removed. A total of 14 users read and classified 100 sequentially posted messages. From these 100 messages, 50 messages were randomly selected for training, and the system predicted the users' choices for the remaining messages using Equation (2). These predictions were one of three classes: Suggested, Not Suggested, or Unknown. The predictions were then compared to the actual classifications provided by the participants. Results are in Table 2.

All classification methods perform better than chance. Based upon these results, 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 under the 'Combined Features' row in Table 2. While the combined classification scheme results in a slightly lower correct classification rate than the subject-alone method, the error rate is much lower.

Table 2. Classsification accuracy for individual sets of features

Classification feature	Classified correctly (%)	Classified unknown (%)	Classified incorrectly (%)
Author	38.4	46.7	14.9
Subject	52.1	35.5	11.8
Textbody	53.6	27.2	19.2
Collaborative	46.2	41.2	12.6
Combined features	51.5	40.9	7.3

4. CASE-BASED FILTERING VIA WORDNET

While GHC is simple and produces good results, its classification power is limited since it linearly combines all input features through the conditional independence assumption; i.e. it does not disambiguate among different word definitions. The underlying problem is that the system does not understand the actual concept behind the text.

4.1. WordNet

To address these problems, the WordNet knowledge base was incorporated through a Case-Based Reasoning (CBR) module. In CBR, experiences are treated as cases that are used to understand new, similar cases. By retrieving individual cases and using the classification of old cases to classify new articles, the system is capable of avoiding the limitations of linearity. Furthermore, by combining these knowledge-based techniques with the GHC technique previously described, the system is capable of growing through the keyword feature tables while also retaining semantic knowledge through WordNet. Finally, the CBR system also supported the retrieval of previously read articles.

WordNet is a lexical reference system inspired by psycholinguistic theories of human lexical memory that organizes English words into synonym sets (Miller, 1995). Relations link the synonym sets. INFOS uses the hierarchical organization of these synonym sets. With approximately 107,000 noun senses and 27,000 verb senses, WordNet v1.5 is the size of a paperback dictionary.

An example of the WordNet hypernym hierarchy (hypernym is an 'ISA' class/subclass hierarchy) for the word 'hacker' is shown in Fig. 2. When a word is found in its lexicon, all definitions or senses of that word are provided. The figure displays two noun definitions; one for golfer, and the other for programmer. The definitions are organized hierarchically, from specific up to abstract concepts along the ISA links.

```
Sense 1
hacker
  => golfer, golf player, linksman
    => player, participant
      => contestant
        => person, individual, someone, mortal, human, soul
          => life form, organism, being, living thing
            => entity
          => causal agent, cause, causal agency
            => entity

Sense 2
hacker
  => programmer, computer programmer, software engineer
    => engineer, applied scientist, technologist
      => person, individual, someone, mortal, human, soul
        => life form, organism, being, living thing
          => entity
        => causal agent, cause, causal agency
          => entity
```

The sense definitions display information from the specific to the general.

4.2. Indexing

If INFOS indexed articles based upon all the senses of nouns and verbs found in an article, then a large number of irrelevant indices would be created due to multiple word meanings.

Consequently, some disambiguation must be performed upon the concepts referenced from the text. The disambiguation method employed in the CBR section of INFOS is to find appropriate noun or verb phrases based on Paice's index extraction algorithm (Paice, 1989). This algorithm assumes that sentences repeat an underlying concept within a 'topic neighborhood' and that those words occurring with a high frequency are likely to be relevant to the actual topic. These words are then extracted as indices. In INFOS, the basic algorithm has been modified by replacing words with WordNet concepts. Consequently, WordNet concepts are compared and recurring concepts within a neighborhood are extracted as features instead of keywords. Details of the algorithm may be found in Paice's work (Paice, 1989).

After candidate concepts 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). The term *frequency* refers to the number of occurrences of a sense within a single document, while the term *rarity* refers to the number of occurrences of a sense across the English language. Frequency is counted by INFOS, while rarity is determined through WordNet. The two values are multiplied to give a general relevancy statistic. This value is used during retrieval to determine how closely an old article matches a new document.

Sense 1

hacker

=> golfer, golf player, linksman

=> player, participant

=> contestant

=> person, individual, someone, mortal, human, soul

=> life form, organism, being, living thing

=> entity

=> causal agent, cause, causal agency

=> entity

Sense 2

hacker

=> programmer, computer programmer, software engineer

=> engineer, applied scientist, technologist

=> person, individual, someone, mortal, human, soul

=> life form, organism, being, living thing

=> entity

=> causal agent, cause, causal agency

=> entity

Fig. 2. Sample WordNet hypernym hierarchies for the word 'hacker'. The sense definitions display information from the specific to the general.

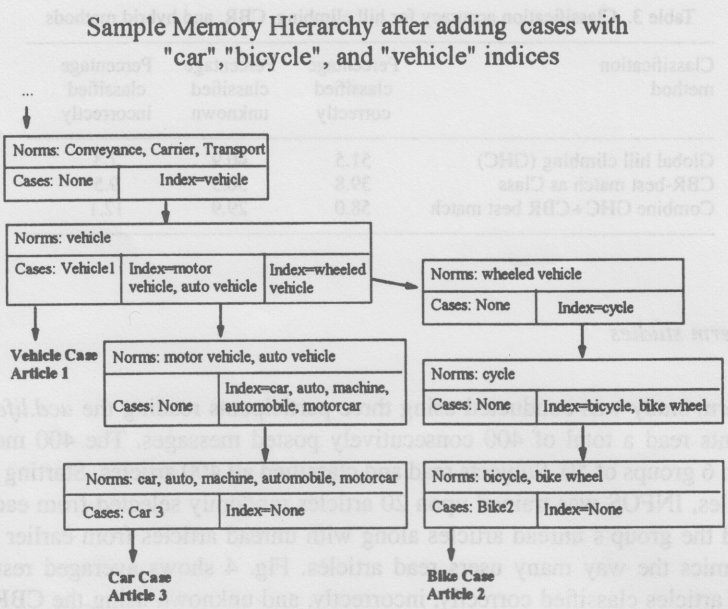

Fig. 3. Sample memory hierarchy for indexing cases.

After all of the WordNet indices have been determined, they are used to index the original document in an abstraction hierarchy. Figure 3 shows a hierarchy with three cases; one article contains the concept 'vehicle', another contains the concept 'bicycle', and the last contains the concept 'car'.

4.3. *Memory retrieval*

After the conceptual hierarchy is created, retrieval is simply a matter of performing a depth-first search through the hierarchy along indices that match the input until cases are retrieved. To support partial matches (e.g. retrieve cases regarding bicycles when the input is about cars), path mismatches in the hierarchy may be traversed until an error threshold value is exceeded. For the experiments, this threshold was set to 15%.

For each case that is retrieved, the overall match value for that case is computed by summing over all n feature queries the distance function as shown in Equation (4). Finally, the retrieved cases are sorted by degree of match. The classification of the old case that best matches the new article is then used as the classification of the new article.

$$Match = \frac{1}{n} \sum_{i=1}^{n} (MatchPercent_i) \times Relevancy_i \tag{4}$$

4.4. *Results*

An identical testing methodology was used to evaluate the CBR scheme as was used with the GHC scheme. Finally, the CBR method was tested in conjunction with the global scheme. In this mode, the global scheme classification was performed first and if the global scheme returned an unknown classification, then the classification of the case-based scheme was used. The global scheme was performed first due to its simplicity and speed compared with the CBR scheme and also its low error. The results in Table 3 indicate that the CBR method alone performs poorly, most likely due to weak disambiguation. The combined CBR and GHC scheme resulted in a higher correct classification percentage than GHC alone, but did add additional error.

Table 3. Classification accuracy for hill climbing, CBR, and hybrid methods

Classification method	Percentage classified correctly	Percentage classified unknown	Percentage classified incorrectly
Global hill climbing (GHC)	51.5	40.9	7.3
CBR-best match as Class	39.8	50.5	9.5
Combine GHC+CBR best match	58.0	29.9	12.1

4.5. *Longer term studies*

A longer term study was conducted using three participants reading the *ucd.life* newsgroup. The participants read a total of 400 consecutively posted messages. The 400 messages were broken up into 6 groups of 50. Subjects read and classified all 400 articles. Starting with the first group of articles, INFOS was trained upon 20 articles randomly selected from each group and then classified the group's unread articles along with unread articles from earlier groups. This procedure mimics the way many users read articles. Fig. 4 shows averaged results with the percentage of articles classified correctly, incorrectly, and unknown using the CBR/Global Hill Climbing method.

Initially, the user model is empty and the results erratic due to the lack of knowledge in the user model. An artifact of the dataset is an initial high classification rate due to a low number of threads, making articles easy to classify. However, after a larger diversity of messages are encountered, the classification rates stabilize at close to 60% correct, 30% unknown, and 10% incorrect. Factors that prevent INFOS from classifying more articles correctly include an influx of new topics that INFOS has not yet learned to classify, and changing user interests.

5. PARTIAL PARSING VIA INDEX PATTERNS

The indexing methods employed in the GHC and CBR sections are based on *bottom-up* knowledge. Given the meaning of individual words, statistics or cases are referenced that may be relevant. In addition to bottom-up recognition, *top-down* methods are also employed by humans through high-level knowledge structures (Schank & Abelson, 1977). These structures require the recognition of concepts at the word, phrase, sentence, and paragraph levels.

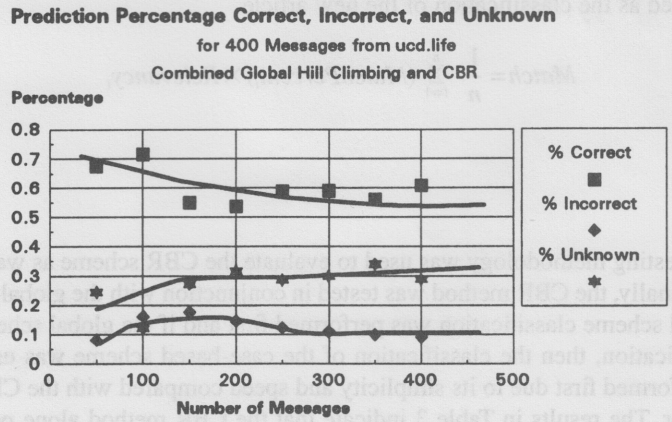

Fig. 4. Long term study for combined CBR/global hill climbing. Performance stabilizes at 60% correct, 30% unknown, and 10% incorrect.

5.1. *Index patterns*

In INFOS, the parsing process is a simplified version of Case-Based Parsing (CBP; Riesbeck & Martin, 1986). CBP is as a recognition process that maps words from input text into the corresponding concepts in memory. In INFOS, the mapping process is from words to the correct concepts in the WordNet abstraction hierarchy. The entity that performs the mapping is called an index pattern.

Index patterns specify lexical and semantic constraints. For example, the index pattern *{<person> 'powers up' <object> => Power-Event}* is activated if the concept <person> is found, followed by the lexical items 'powers up', and finally followed by the concept <object>. In this pattern, the words 'powers' and 'up' are static lexical items while <person> and <object> are conceptual items. Conceptual items are simply abstractions. A concept from a text article matches a conceptual item from an index pattern if the text article's concept falls under the same category as the index pattern's concept. The categorical match is positive if at least one sense of the index pattern's concept is an abstraction of at least one sense of the text article's concept. For example, in Fig. 1, both 'engineer' and 'hacker' would activate <person> since both words are specializations of 'person', but 'life form' would not activate <person> since it is an abstraction. In this manner, index patterns implicitly encode syntactic features of a target language and directly encode lexical and semantic constraints. These constraints result in the disambiguation of the input text.

Rather than parse into instances of knowledge structures to identify cases, the process implemented in INFOS is only concerned with linking from index patterns to a classification. However, parsing into knowledge structures does provide greater flexibility, and remains an area of future work.

Figure 5 depicts the mapping of input text into the index pattern *{<person> <search> <help>}*. If the user is not interested in reading articles with people asking for help, then this pattern could be associated with a rejected classification. In this example, the article text contains the sentence "my employer is looking for assistance..." Since "employer" is a specialization of *<person>*, "looking" is a specialization of *<search>*, "assistance" is a specialization of *<help>*, and all of these components match in the same order specified in the index pattern, then the index pattern is activated and used to classify the text.

5.2. *Experimental results—information filtering via index patterns*

Owing to time constraints and the requirement that users be knowledgeable about WordNet's structure, index patterns were not implemented in the user-tested version of INFOS. Instead, a

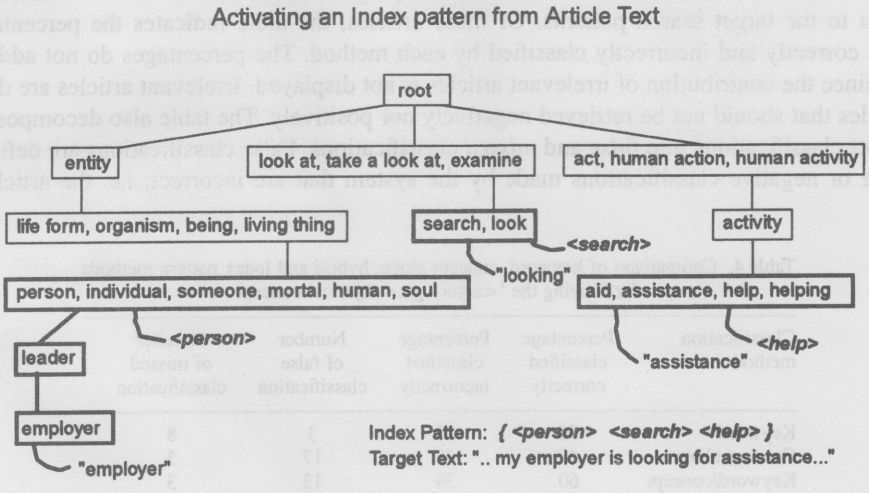

Fig. 5. Activation of index pattern.

preliminary experiment was conducted using 47 consecutively posted articles from the *comp.ai* newsgroup. Specific concepts were selected as targets of interest and others selected as targets of disinterest. All articles were read and classified by the author. The author-judged classifications served as the correct classifications that the system attempted to match.

The theme selected for filtering was the concept of an actor getting an object. Two concepts were selected for positive retrieval: the act of retrieving/finding information and the act of getting some type of system. The index patterns implemented to represent these concepts are:

{<person> <get> <information>=> suggested}
{<person> <search> <information>=> suggested}
{<person> <get> <system>=> suggested}
{<person> <search> <system>=> suggested}

The concepts selected to be filtered include the concepts of searching for a location, searching for a person, and searching for a message (such as help). The index patterns implemented to represent these concepts are:

{<person> <get> <location>=> not suggested}
{<person> <search> <location>=> not suggested}
{<person> <get> <person>=> not suggested}
{<person> <search> <person>=> not suggested}
{<person> <get> <communication>=> not suggested}
{<person> <search> <communication>=> not suggested}

All articles were initially given a weight of 0. Each time a positive index pattern was triggered, the respective article's weight was incremented. Similarly, each time a negative index pattern was triggered, the respective article's weight was decremented. In the end, those articles with a positive weight were classified positively, and those with a negative weight classified negatively.As a benchmark, filtering was also performed using keywords, WordNet concepts alone, and a hybrid keyword/concept-alone method. The concept-alone method is identical to the index pattern scheme, except a pattern was considered to match with a sentence from article text if the concepts in the pattern matched anywhere with concepts in the article's sentence. In other words, syntax and pattern order was ignored.

The keyword method attached an ambivalent weight to the words *get*, *search*, *look*, *find*, a positive weight to the words *information*, *system*, *knowledge*, *program*, *work*, *paper*, *research*, and a negative weight to the words *location*, *direction*, *person*, *communication*, *message*, *help*, *assistance*, *aid*. Articles were classified using the same weight scheme as described above for index patterns. Finally, the keyword/concept-alone hybrid method applied the keyword scheme first, and if no matches were found, then the classification result of the concept-alone scheme was used.

The results are shown in Table 4. The human judge identified 25 of the 47 articles to be relevant to the target search patterns. Of these articles, the table indicates the percentage of articles correctly and incorrectly classified by each method. The percentages do not add up to 100% since the contribution of irrelevant articles is not displayed. Irrelevant articles are defined as articles that should not be retrieved negatively nor positively. The table also decomposes the incorrect classifications into *false* and *missed* classifications. False classifications are defined as positive or negative classifications made by the system that are incorrect; i.e. the article was

Table 4. Comparison of keyword, concept alone, hybrid and index pattern methods for filtering the '<actor>get<object>' concept

Classification method	Percentage classified correctly	Percentage classified incorrectly	Number of false classification	Number of missed classification
Keyword	52	23	3	8
Concept alone	40	42	17	3
Keyword/concept	60	34	13	3
Index patterns	80	17	5	3

actually classified opposite the prediction or was deemed irrelevant. Missed classifications are defined as articles that the system deemed irrelevant but were classified positively or negatively by the human.

Table 4 indicates that the index pattern method performed best, correctly classifying 80% of the articles. The concept-alone scheme performed poorly, primarily due to the false classifications resulting from incorrect word disambiguation. The opposite problem appears for the keyword method. Articles that should be retrieved are missed since specific keywords, not concepts, are being compared. The hybrid scheme improves upon both keyword and concept-alone methods, but at the cost of higher error presumably carried over from the concept-alone scheme.

The errors made by the index pattern method are primarily due to anaphora in the text. The index pattern scheme does not disambiguate pronouns; consequently, relevant patterns may not be activated. To address this problem, further semantic processing needs to be implemented.

6. CONCLUSIONS AND FUTURE WORK

This paper has investigated the application of keyword hill climbing methods, collaborative filtering, knowledge-based systems, and partial parsing to the process of information filtering. Each component adds additional capabilities to the filtering system. The keyword approach scales well and supports user-modifiability and learning. The WordNet component supports higher recall through conceptual understanding of the text. However, the tradeoff is lower precision unless more robust disambiguation is performed to reduce error. This may be addressed through the addition of index patterns. Additionally, while GHC is quick, the existing CBR and WordNet system is slow. Future work is necessary to increase performance and make the creation of index patterns and other high-level constructs easy for users to manipulate and powerful enough to accurately filter articles.

Acknowledgements—This work is supported in part by Apple Computer, Inc. in conjunction with a UC Micro grant. The authors are indebted to Dr. Rao Machiraju and Apple Computer Inc. for their support.

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. (Available at WW: http://www.ics.hawaii.edu/~csdl/urn)

Collis, K. F. (1980). Levels of cognitive functioning and selected curriculum areas. In J. Kirby & J. Biggs (Eds), *Cognition, development, and instruction* (pp. 65-89). New York, NY: Academic Press.

DeJong, G. (1982). An overview of the FRUMP system. In W. G. Lehnert & M. H. Ringle (Eds), *Strategies for natural language processing* (pp. 149-174). Hillsdale, NJ: Lawrence Erlbaum.

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). New York: Institute of Electrical and Electronics Engineers.

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* (pp. 624-643). Barcelona, Spain.

Jennings, A. & Higuchi, H. (1992). A personal news service based on a user model neural network. *IEICE Transactions Inf. and Systems*, E75 D(2), 198–209.

Lang, K. (1995). NewsWeeder: Learning to filter netnews. *Proceedings of the Twelfth International Machine Learning Conference*, Tahoe City, CA, 9–12 July. San Francisco, CA: Morgan Kaufmann.

Lashkari, Y., Metral, M., & Maes, P. (1994). Collaborative interface agents. *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle, WA, 31 July–4 August (pp. 444–449). Menlo Park, CA: AAAI Press; Cambridge, MA: MIT Press.

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

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Mock, K. (1996). Hybrid hill-climbing and knowledge-based techniques for intelligent news filtering. *Thirteenth*

National Conference on Artificial Intelligence, Portland, Oregon. Menlo Park, CA: AAAI Press; Cambridge, MA: MIT Press.

Mock, K., & Vemuri, V. (1994). Adaptive user interface for intelligent information filtering. *Proceedings of the Third Golden West International Conference on Intelligent Systems*, Las Vegas, Nevada (pp. 506–517). Dordrecht; Boston: Kluwer Academic Press.

Paice, C. D. (1989). Automatic generation and evaluation of back-of book indexes. Prospects for Intelligent Retrieval, Informatics 10, *Proceedings of the Aslib Informatics Group and the Information Retrieval Specialist Group of the British Computer Society*, King's College Cambridge, 21–23 March (pp. 506–517). London: Aslib.

Ram, A. (1992). Natural language understanding for information-filtering systems. *Communications of the ACM*, *35*(12), 80–81.

Riesbeck, C. K. & Martin, C. E. (1986). Direct memory access parsing. In J. Kolodner, & R. Riesbeck (Eds), *Experience, memory, and reasoning*. Hillsdale, NJ: Lawrence Erlbaum.

Salton, G. (1971). *The SMART retrieval system: Experiments in automatic document processing*. Englewood Cliffs, NJ: Prentice-Hall.

Schank, R. C. & Abelson, R. (1977). *Scripts, plans, goals, and understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Sheth, B. D. (1994). *A learning approach to personalized information filtering*. Masters Thesis. Department of Computer Science and Engineering, Massachusetts Institute of Technology.

Soderland, S. & Lehnert, W. (1994). Corpus-driven knowledge acquisition for discourse analysis. *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle, WA, 31 July–4 August (pp. 827–832). Menlo Park, CA: AAAI Press; Cambridge, MA: MIT Press.

Stevens, C. (1992). *Knowledge-based assistance for accessing large, poorly structured information spaces*. Doctoral Dissertation. Department of Computer Science, University of Colorado.