An Adaptive Algorithm for Learning Changes in User Interests

Dwi H. Widyantoro, Thomas R. Ioerger, John Yen
Department of Computer Science
Texas A&M University
College Station, TX 77844-3112
dhw7942,ioerger,yen@cs.tamu.edu

Abstract

In this paper, we describe a new scheme to learn dynamic users' interests in an automated information filtering and gathering system running on the Internet. Our scheme is aimed to handle multiple domains of long-term and short-term user's interests simultaneously, which is learned through positive and negative user's relevance feedback. We developed a 3-descriptor approach to represent the user's interest categories. Using a learning algorithm derived for this representation, our scheme adapts quickly to significant changes in user interest, and is also able to learn exceptions to interest categories.

Keywords
Information Filtering, Intelligent Agents.

1 Introduction

The spread of the World Wide Web and online news sources on the Internet recently has changed the way people locate information and their news reading habits. As more online news sources become available on the Internet, people have more options to read news articles that they think are interesting. However, selecting relevant articles from a group of news articles on various topics and online sources is still considered a time consuming process. Although search engines can help finding relevant news articles, it still requires the user to describe interests each time the user wishes to pull the news. Recent efforts have been devoted to overcome this problem by personalizing an information filtering system. This system takes into account the user profile information to present relevant information to its user effectively.

We have developed *Alipes*, a personalized news agent that gathers articles periodically from various online news sources and filters them on behalf of its users [16]. *Alipes* maintains the profiles of its users based on which a set of relevant news articles from the World Wide Web is recommended to its users. Moreover, *Alipes* adapts to the dynamics of the user's interests by learning from the user's feedback. This paper describes a new scheme for learning user interest that has been incorporated in *Alipes*. Our scheme is able to adapt to the dynamic nature of users' interests, which can change from slowly to suddenly, from one domain to another, over a very short to very long period.

Most previous information filtering systems on the Internet, for example *WebMate* [5] use a keyword vector to represent categories of user interest. Incremental learning algorithms with such a representation have trouble adapting in an appropriate time frame as interests slowly or quickly shift focus. Our approach uses a 3-descriptor scheme to represent a category of interest in a profile and its learning algorithm. In this scheme, an interest category consists of three descriptors: one long-term descriptor to maintain long-term interests, and other two descriptors, positive and negative, to keep up with short-term interests. This approach is similar to an incremental method for learning in domains with concept drift, where multiple concept representations that generalize examples over different window sizes are maintained simultaneously [14, 15]. Compared to systems that mainly use a single-descriptor model for interest category representation, the 3-descriptor scheme has several advantages. The 3-descriptor scheme allows learning of long-term and short-term interests simultaneously, and also handles exceptions of interests within an interest category. This capability cannot be achieved using the single-descriptor representation.

The rest of this paper will be organized as follows. Related work and its limitations will be described briefly in the following section. The third section describes our approach for modeling and learning of user profiles. Then, a brief description of the evaluation methodology and results is presented in the fourth section, followed by a conclusion.

2 Previous Work

There are many systems that have recently been described for news and information filtering. *Webmate* keeps track of user interest through multiple TF-IDF vectors [5]. *Fab* is an adaptive system for Web page recommendation which represents user profiles as a single feature vector [2] and handles multi-topic interests [3]. *Syskil & Webert* is an intelligent agent that represents a profile as Boolean features and uses a Naive Bayesian classifier to determine whether a Web page is relevant or not [10]. Lang compared various alternatives to learn a static user profile in his *NewsWeeder*, a newsnet filtering system [7]. Neural networks have also been explored to learn user profiles for topic spotting [18] and for filtering news articles on the Internet [13]. In *NewT* [12] and *Amalthea* [9], genetic algorithm is employed to learn user interests. Incremental relevance feedback is a common method used to learn user profile for information filtering in these systems. Allan explored the effectiveness of this method and demonstrated that good results can be obtained using only a few judgments [1].

Although the performance of these systems improves after learning a user profile, most of them do not address the effectiveness of their approaches to adapting to changing interests and handling exceptions of interests within an interest category. Except in works by [4, 8, 9, 12], their evaluation assumes that the user's interests do not change during the evaluation process. In real life, however, both the user's long-term and short-term interests usually change over time. Long-term interests are interests that result from an accumulation of experiences over a long time-span. Meanwhile, short-term interests are interests in events on a day-to-day basis which change over a short period. Therefore, the capability to adapt to these changes effectively and to handle exceptions to categories is still an open problem, and these issues will be addressed in this paper.

3 Modeling and Learning User Profile

The capability to model and learn a user profile is at the heart of a personalized information filtering system. We will describe in this section our approach to designing a profile representation, how to use the representation for information filtering, and how to develop a learning algorithm that adapts to the dynamics of the user's interests.

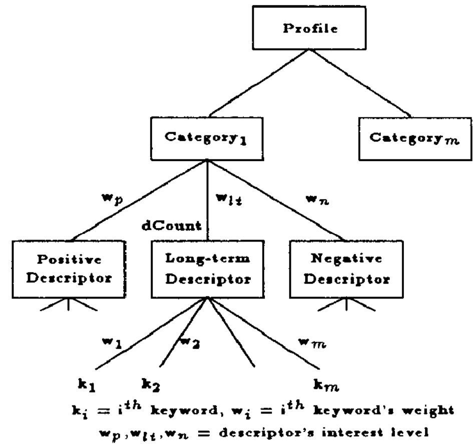

Figure 1: A 3-descriptor Representation

3.1 Profile Representation

The basic structure of an interest category representation is a feature vector. It contains a list of keywords and each keyword is weighted according to its degree of importance. There are several keyword weighting schemes that have been developed such as TF-IDF [11] and LSI [6]. In this work, we use the TF-IDF weighting scheme to assign the keywords' weights due to its appropriateness for use in an online learning algorithm. Based on TF-IDF, the keyword importance is proportional to the frequency of occurrence of each term in each document and inversely proportional to the total number of documents in a document collection in which the term occurs [11]. It assumes that keywords appearing in fewer documents discriminate better than the ones appearing in more documents. The weight of keyword i in a document d is then defined as follows:

$$w_i^d = tf_i^d . \log \frac{N}{df_i} \tag{1}$$

where N is the number of documents in the document collection, df_i is the document frequency of term i, and tf_i^d is the frequency of term i in document d. The m highest weighted terms and bigrams (pairs of adjacent words) are then kept and normalized such that $|w_j^d| = 1$ for j = $[1..m]$. The terms are extracted from a document that has been pre-processed by: removing HTML tags and links, Java scripts and stop words [1], stemming words [2].

In a 3-descriptor representation, an interest category C is composed of three descriptors: a positive d_p^c, a negative d_n^c, and a long-term d_{lt}^c descriptor. Each of the descriptors consists of a list of pairs of keywords and their

[1]The stop list consists of 293 common words, like "a", "the", "although", etc.

[2]We use Porter's stemming algorithm to find the root of words and thus reduce the number of terms.

weights. Figure 1 illustrates the structure of a profile using this scheme. The *positive* and *negative* descriptors maintain a feature vector learned from documents with positive and negative feedback respectively, while the *long-term* descriptor maintains the feature vector of a document from both types of feedback. Each descriptor also has an interest weight to represent the interest level of the corresponding interest category's descriptor. Interest weights w_p^c, w_n^c and w_{lt}^c are used to describe the level of interest in positive, negative and long-term descriptors of interest category C, respectively. The range of w_p^c and w_n^c is [0,1], while the range of w_{lt}^c is (-1,1). Negative and positive values of w_{lt}^c describe the uninterestingness and interestingness in the domain of interest represented by the feature vector of the long-term descriptor. In addition to the long-term descriptor, a document counter *dCount* is maintained to keep the total number of documents that have been observed. Formally, the representation of interest category C can be written as follows.

$$Cat_c = \langle (w_p^c, d_p^c); (w_n^c, d_n^c); (dCount, w_{lt}^c, d_{lt}^c) \rangle \quad (2)$$

The user may have multiple interest categories, and the profile of a user P having n interest categories is represented as:

$$Profile_p = \{Cat_1^p, Cat_2^p, ..., Cat_n^p\} \quad (3)$$

Based on the above representation, the document-filtering process is performed and a learning algorithm to model user profiles is developed to accommodate intuitively defined behavior of changing user's interests.

3.2 Information Filtering

The information-filtering process is performed by selecting the n most relevant documents from a set of documents. For each document in the set, the interestingness of the document is assessed according to the match to an interest category of the profile and the degree of interest in that category. The assessment is calculated as a numeric value ranging from -1 to 1 that is assigned to a document as the score of the document with respect to the profile being considered. A positive value of the score indicates that the document is interesting to some degree. Conversely, a document with a negative score is uninteresting. Based on these scores, the documents are ranked, and the n most relevant documents are obtained from the n top ranked documents.

Given a document feature vector fv_d, the score of fv_d for a profile P is computed as follows.

1. Calculate the relevance of each category C in profile P with the document being examined. The category relevance is defined by equation 4 as the maximum similarity between a document feature vector and either the feature vector of positive, negative or long-term descriptor, where the cosine similarity is employed to measure the similarity between the two vectors.

$$\begin{aligned}\mathrm{Rel}(Cat_i, fv_d) = \max\{&\mathrm{Sim}(d_p^i, fv_d), \\ &\mathrm{Sim}(d_n^i, fv_d), \mathrm{Sim}(d_{lt}^i, fv_d)\}\end{aligned} \quad (4)$$

where

$$\mathrm{Sim}(d^i, fv_d) = \frac{d^i \cdot fv_d}{|d^i| \cdot |fv_d|} \quad (5)$$

2. Calculate the score of each descriptor in the category C with the greatest relevance:

$$\begin{aligned} w_{pos} &= w_p^c * \mathrm{Sim}(d_p^c, fv_d) \\ w_{neg} &= w_n^c * \mathrm{Sim}(d_n^c, fv_d) \\ w_{long} &= w_{lt}^c * \mathrm{Sim}(d_{lt}^c, fv_d) \end{aligned} \quad (6)$$

where

$$c = \arg\max_i\{\mathrm{Rel}(Cat_i, fv_d)\}. \quad (7)$$

3. Compute the final document score as follows.

$$\begin{aligned}\mathrm{Score}(Profile_p, fv_d) = \; & \max(w_{long}, w_{pos}) + \\ & \min(w_{long}, -w_{neg}) \end{aligned} \quad (8)$$

The final value of the document score is a fusion between the score of positive w_{pos} and negative w_{neg} interest. The score of long-term interest w_{long} contributes to either the positive or negative interest depending on the sign of its value.

3.3 Learning User Profiles

The learning algorithm in *Alipes* allows incremental and online learning, which enables reactive learning as well as long-run learning. For the clarity of presentation, the learning algorithm will be presented in a high-level description prior to explaining the details of the algorithm that follows.

3.3.1 *Learning Algorithm*

The learning process in a personalized information filtering system relies on a user's feedback. Using the feedback information, the profile is modified such that it will be incorporated in future information-filtering tasks. The feedback consists of feedback type *fbType*, document to be learned fv_d and learning rate α. The feedback type can be positive or negative to represent that the user likes or dislikes the document's content. The learning rate represents the strength of the user's preference (e.g. very interesting, interesting, not bad, uninteresting etc.) and its range is (0,1]. In general, the algorithm to modify a user profile P is defined as follows.

Input: *fbType*, fv_d and α
Output: *modified P*

1. Find the most relevant category C in profile P
2. **If** Rel$(Cat_c, fv_d) \geq \theta$ **then**
3. LearnUserFeedback (*P*, *fbType*, fv_d, α)
4. **Else**
5. CreateNewCategory (*P*, *fbType*, fv_d, α)
6. **End if**

Finding the most relevant category in the above algorithm is the same process of finding the greatest category relevance in document scoring described earlier. A *threshold* constant θ is defined to determine when the highest similarity to an existing category is low enough to justify creating a new interest category. This process is used to learn various categories of interests based on the category relevance measure and the threshold constant. How to set this value will be addressed later in the evaluation section. The learning process in step 3 includes updating the descriptor feature vectors and modifying the long-term and short-term descriptors' interest weights.

3.3.2 *Updating the Feature Vectors of Descriptors*

The modification of a descriptor's feature vector with the feature vector of a sample document should accommodate the learning of short-term and long-term interests. Short-term interests tend to be reactive so that feedback will be incorporated immediately in future information-filtering. On the contrary, long-term interests change gradually. The modification of a long-term interest area should be sufficiently small that it will preserve the feature vector of documents from past feedbacks while still considering the contribution of document feature vector from the most recent one. Taking all these into account, the updating of a descriptor feature vector in category C is as follows:

$$d^c_{(new)} = d^c_{(old)} * (1 - \alpha) + fv_d * \alpha \tag{9}$$

where d^c is either d^c_p for positive feedback, d^c_n for negative feedback, or d^c_{lt} for both positive and negative feedback. The learning rate α is used to adjust the contribution of the learned document. For the short-term descriptors (e.g. d^c_p and d^c_n), the value of α is obtained directly from the user's preference when giving feedback. A high learning rate results in a significant contribution of the learned document to the positive or negative descriptor. Therefore, the modification of these descriptors will be in line with the user's preference, and will determine the reactive behavior of short-term interests. However, the learning rate for the long-term descriptor d^c_{lt} is determined inversely by $dCount$, the number of example documents that have been learned so far. The value of α in equation 9 is derived by equation 10 to modify the feature vector of the long-term descriptor.

$$\alpha = \frac{1}{dCount + 1} + 0.05 \tag{10}$$

As more feedback is learned, the contribution of the most recently learned document becomes smaller and therefore the previously learned interests are still preserved. The constant 0.05 is used to prevent a complete stoppage of learning, since α would otherwise converge to 0 in the limit (after learning many documents). Thus, it allows the long-term descriptor to keep learning regardless of the number of previously learned examples.

3.3.3 *Learning Short-term Interest Weights*

As mentioned earlier, the descriptor feature vector represents the interest area, and the degree of interest in the area is denoted by the descriptor's interest weight. The learning of short-term interests is performed by modifying the positive and negative descriptors' interest weights, w^c_p and w^c_n. These weights are updated to reflect the user's short-term interests so that any feedback (positive or negative) will be incorporated immediately in future information filtering. Specifically, the update of these interest weights is performed by increasing the corresponding interest weight according to the level of confidence obtained from the relevance feedback, and by decreasing the interest weight of the opposite descriptor. The amount of reduction in interest weight of the opposite descriptor is proportional to the learning rate and the similarity between the feature vector of the learned document and the one of the opposite descriptor. Equations 11 and 12 express these update rules for learning positive feedback.

$$w^c_{p(new)} = w^c_{p(old)} + (1 - w^c_{p(old)}) * \alpha \tag{11}$$
$$w^c_{n(new)} = w^c_{n(old)} * (1 - \alpha * \mathrm{Sim}(d^c_n, fv_d)) \tag{12}$$

For learning negative feedback, the same formulas are used by changing w_p with w_n and d_n with d_p, and vice versa.

3.3.4 *Learning Long-term Interest Weights*

In a long-term descriptor, the modification of the descriptor's interest weight w^c_{lt} should capture a reluctance of the interest to change after learning in the long run. For this motivation, a bipolar sigmoid function is used to govern the change of the long term descriptor's interest weight so that the change of w^c_{lt} will be more gradual. The function ranges from -1 to 1 where the lower and upper limit can be approached using argument values $-\infty$ and $+\infty$ respectively. By defining the ordinate (y-axis) of the function to be the value of w^c_{lt}, the use of this function is expressed in equation 13.

$$w^c_{lt(new)} = f(f^{-1}(w^c_{lt(old)}) \pm \alpha) \tag{13}$$

where $f(x)$ is a bipolar sigmoid function.

$$f(x) = \frac{2}{1+\exp^{-x}} - 1 \tag{14}$$

First, the current value $w^c_{lt(old)}$, the ordinate, is projected to its abscissa using the inverse of bipolar sigmoid function. Second, the learning rate α is then added to the abscissa value for positive feedback or subtracted from the abscissa value for negative feedback. Finally, the new abscissa value is projected back to its ordinate as the new value of $w^c_{lt(new)}$. The input α to update $w^c_{lt(new)}$ is obtained from the user's preference rather than the one derived in equation 10. So the same amount of effort to change the level of long-term interests is required as to build them.

3.3.5 *Creating New Interest Categories*

The learning of new interests from positive feedback is initialized as follows:

$$d^c_{lt} = fv_d \qquad w^c_{lt} = f(\alpha)$$

$$d^c_p = fv_d \qquad w^c_p = \alpha \tag{15}$$

$$d^c_n = \{\} \qquad w^c_n = 0 \tag{16}$$

where $f(\alpha)$ is the bipolar sigmoid function. The assignment of w^c_{lt} and w^c_p uses equations 13 and 11 respectively by setting their initial values to zero. For negative feedback, $w^c_{lt} = f(-\alpha)$ and the assignment of equations 15 and 16 are swapped one of another so that $d^c_n = fv_d, w^c_n = \alpha, d^c_p = \{\}$ and $w^c_p = 0$.

4 Evaluation

Experimental evaluation has been conducted to measure the performance of *Alipes* to learn user's interests from user feedback. The main objectives are to evaluate the adaptability of the 3-descriptor scheme to the changing interests of the user and the ability of the scheme to handle exceptions to categories.

4.1 Method

4.1.1 *Data*

Documents used in our experiment are news articles in HTML format collected from 12 different online newspapers and magazines (Yahoo and Excite's Sport News, UsaToday, USNews, Fortune, PCWeek, PCMagazine, BusinessWeek, Windows, People, Time and Internet World), at different times. The collection contains 1427 documents with six different general topics: world, financial, health, weather, technology and sport news. The length of each processed document varies with an average number of distinct terms of 228.

4.1.2 *Procedure*

The experimental procedure to evaluate the adaptability of the 3-descriptor model to the changing interest of the user is designed to simulate the application of the scheme in a news agent. A detailed description of this procedure is described in [17]. Starting with an empty profile, the system provides a recommendation of 10 articles to the user. The user (real or simulated) examines the articles and gives feedback on whether each article is interesting or uninteresting with a degree of confidence. The system then learns from the user's feedback and modifies its profile. At this point, one cycle of evaluation ends. The next cycle starts using the most recently modified profile. At each cycle, the system's performance is measured, and a different set of 200 documents is selected to be filtered. This simulates the changing of news articles in online daily newspapers or weekly magazines that may have overlapping topics. To observe how well the system adapts to the changes of interest, the user's interest is *inverted* at the twentieth cycle by swapping the interesting and uninteresting domains of interest. In these experiments, the user is simulated by a target profile containing a list of interesting and uninteresting domains of interest. Following the experiment by Moukas and Zacharia [9], the positive or negative feedback is given based on the similarity between the examined article and the target profile. We use *accuracy* as the measure of system's performance. Accuracy is defined as the percentage of the n highest ranked documents (in this experiment n=10) recommended by the system that agree with those selected by the target profile, assuming they use the same document set.

To measure the 3-descriptor scheme's performance to handle exceptions of interest, a different experimental procedure was used, which will be described later.

4.1.3 *Performance Comparison*

To compare the performance between our 3-descriptor model and a single-descriptor scheme, we used the algorithm for learning user profiles employed in the *Webmate* system [5]. The algorithm was chosen due to its similarity in profile representation and in interest-domain clustering [3]. To make the algorithm comparable with the one developed in our work, bigram identification was added, and reward for keywords appearing in a document's title and header, applied in the original algorithm, was eliminated in the modified algorithm. An important difference between *WebMate* and *Alipes* is that *WebMate* was not originally designed to learn from negative feedback. To make the comparison fair, we implemented a version of the system that could do

[3]Experiments in comparing Alipes with the Rocchio algorithm are currently in progress.

this (referred to *Webmate-neg* below). The learning of negative feedback is performed by subtracting the feature vector of a learned document from the matching category in the profile.

4.2 Results

We conducted initial experiments to determine the optimal threshold value θ for creating new categories and the optimal number of keywords for representing feature vectors. By varying the threshold values from 0.05 to 0.65, we found that the optimal setting, where it gives the highest average accuracy, was 0.25. Having lower or higher threshold values, which leads to fewer or more categories, degrades the system performance. Similarly, by varying the number of keywords obtained from 20 to 220 highest weighted keywords, the optimal value for this parameter was found to be 90. Fewer keywords tend to remove important keywords while too many keywords will add more noise. The experimental results described in this sub section use these parameter values.

4.2.1 *Accuracy*

Figure 2 shows that the performance of the system employing a 3-descriptor model(*Alipes*) in general outperforms the one using a single-descriptor model(*Webmate* or *Webmate-neg*). The performance in both the 3-descriptor and single-descriptor models increases rapidly after the first iteration. However, the system's performance is erratic afterwards because a different set of documents is used in each round. In the subsequent iterations, therefore, the new document set may not provide documents representing a previously learned interest category, and may introduce other documents that match the target profile but have not yet been learned by the system. After the target profile is inverted at the twentieth iteration, the system's performance drops to its lowest value. It takes a short time for our 3-descriptor approach to adapt to this sudden change before the system regains its performance. The recovery process is worse in the single-descriptor case. It takes much more time for the single-descriptor scheme to restabilize after inversion.

We observe that *Alipes* takes only slightly longer to reach its highest accuracy after profile-inversion (about 5 iterations) than when starting from the scratch. The fact that learning new interests from an empty profile is faster than learning the inverted interest is an effect of long-term learning. To increase the interest level of a long-term descriptor from a negative value after the profile is inverted requires more effort (e.g. more feedback from the user)than starting from an empty profile. *WebMate*, however, recovers much more slowly,

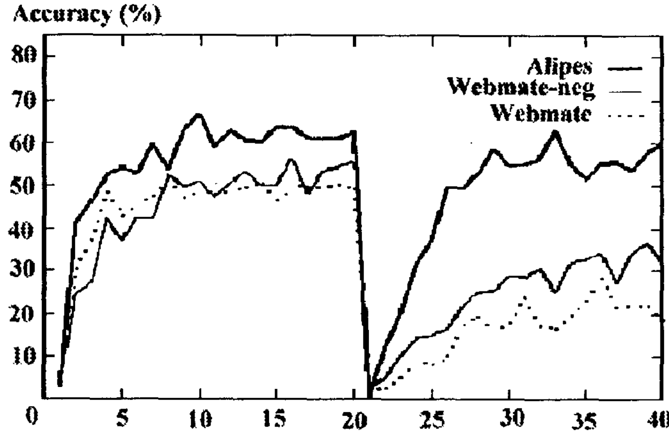

Figure 2: The System's Accuracy (percent of top matches that are relevant). At iteration 20, the target profile was inverted, simulating a dramatic change in interests. *Webmate-neg* is a version of WebMate that learns from both positive and negative feedback, *Webmate* learns only from positive feedback.

and never completely restores its performance from before profile-inversion by the end of the experiment. This is because *WebMate*'s use of single-descriptors to represent category interests is easily confused by a sudden change in interests. Instead of mixing all the feedback, *Alipes* effectively maintains long-term interests, while using its short-term descriptors to provide more reactive behavior.

The role of long-term descriptor is to maintain the past learned interests, and this descriptor basically performs the same function as the single-descriptor scheme in *WebMate*. By making a slight change in this descriptor, most feature vectors of the past learned documents are retained while still incorporating the new learned document. This enables the system to provide recommendations according to the current and the previous learned interests. Therefore, it is not surprising that by adding short-term descriptors as in the 3-descriptor model, its accuracy of prediction is better than the single-descriptor model during the first twenty evaluations.

The role of short-term descriptors becomes apparent in the presence of changing interests of the user. The feature vector update rule in these descriptors allows the system to be responsive to the most recently learned interest, with respect to the confidence level of the user's relevance feedback. On a strong positive feedback, which is given by the user as he/she reacts due to the low performance of the system on the change of user's interests, the learning process in the positive descriptor enables the system to adapt quickly to the user's new interest. The system's responsiveness is strengthened by the negative descriptor when learning a strong negative feedback, which allows the system to quickly exclude uninteresting documents. Additionally, the interaction between the positive and the negative descriptors (e.g. increasing the level of positive interest will reduce the level of negative interest according to the similarity between both interests, and vice versa) makes the adaptation even faster. As a result, the effectiveness of the 3-descriptor scheme over the single-descriptor model to adapt to the drastic change is evident as shown in Figure 2.

Thus, the short-term descriptors explain why the performance improves over a single-descriptor model. The most recently learned interest is significantly taken into account during the information-filtering process. In the single-descriptor model, however, this is not the case. As more documents are learned by the single-descriptor system, the contribution of the new learned interest becomes insignificant. Because the representation of the 3-descriptor scheme is more expressive than the single-descriptor model to capture the user's interests, its accuracy of prediction to recommend interesting documents with respect to the target profile is also better.

We have also conducted experiments that change the target profile more gradually. From these experiments we found that in a setting where the information to be filtered changes slowly over time (e.g. the content of news articles in newspapers or magazines), the difference of performance that is due to the target profile change from the diversity of information sources is less apparent.

4.2.2 *Learning Exceptions to Categories*

In this experiment, our objective was to evaluate the other potential advantage of our 3-descriptor scheme over single-descriptor models: that it can learn interest categories with exceptions. Specifically, the negative descriptor of a category in a user profile allows the system to distinguish (with a unique set of keywords) documents that are related to the overall category but given negative feedback by the user. In this experiment, we attempted to train both *Alipes* (using the 3-descriptor model) and *Webmate* (using a single-descriptor scheme) on a set of Sports documents taken from an online news source, excluding articles about Golf, and then test each system to determine how the use of a negative descriptor affects the ability to rank documents correctly according to this specialized interest area.

The sources of the documents were from the sites described above, which provide documents in a predetermined hierarchy of categories. Three groups of documents were selected: 397 Non-Sports articles, 20 articles about Golf, and 118 articles about Other-Sports. In a given run, 20 Other-Sports articles and 10 Golf articles were chosen at random as a training set. *Alipes* was trained by giving the 20 Other-Sports articles with positive feedback and the 10 Golf articles with negative feedback, while *Webmate* was only trained on the 20 Other-Sports articles (since the original system could only use positive feedback). We also tested a version of *WebMate* (referred to as *Webmate-neg*) that was modified to accept negative feedback, and we gave it both the positive feedback (Other-Sports articles) and negative feedback (Golf articles). Then a separate test set, consisting of 10 randomly-selected other-Sports articles and 5 random Golf articles not used during training, along with the 397 Non-Sports articles, was ranked in terms of user interest by all three systems.

Sports	Average Ranking		
	Alipes	*Webmate*	*Webmate-neg*
Other-Sports	6.1%	5.8%	5.4%
Golf	95.3%	14.3%	10.4%

Table 1: Learning Execptions to Categories. All three systems were trained on articles about all Sports except Golf. *Alipes* and *Webmate-neg* were also given explicit negative feedback about Golf articles. Average rankings of test articles in these categories relative to a large set of Non-Sports articles are shown. Top of ranking = 0%; bottom of ranking = 100%.

This training and testing procedure was repeated 10 times. In each run, we calculated the average ranking of the Other-Sports test documents (the target category) and the Golf test documents (the exception category), and divided these rankings by the total number of documents ranked (412) to get a percentile score (0%=highest interest, top of list; 100%=lowest interest, bottom of list).

Table 1 shows the results of this experiment. In the case of *Alipes*, the Golf documents were consistently ranked at the bottom of the list (95.3%, i.e. within top 393 out of 412 document, on average), and the Other-Sports documents were highly recognized articles in the target category (average ranking of Other-Sports was 6.1%). In contrast, both Golf and Other-Sports documents in *Webmate* were ranked high (14.3% for Golf and 5.8% for Other-Sports). The explanation for this behavior is that the single-descriptor model, when given a wide range of documents about Other-Sports, generalizes this by identifying keywords that are associated with Sports in general. Hence this category covers Golf documents, which unintentionally get ranked high by *Webmate*. In *Webmate-neg*, Golf documents are still ranked high, eventhough they were given explicit negative feedback. The negative feedback causes the keywords that are unique to the exception category to be dropped from the feature vector, but others are retained, which reflects the inadequacy of a single-descriptor representation. Hence both single-descriptor schemes fail to discriminate between the two categories. In contrast, the negative descriptor in *Alipes* can recognize the Golf documents as a negative interest and ranks them very low. This helps avoid the over-generalization of Other-Sports made by the positive descriptor. So the 3-descriptor scheme with negative feedback enables *Alipes* to learn user interest categories with exceptions more accurately.

5 Conclusion

Changing interests are an undeniable fact in real life. The time scale may vary from hours to years long and the degree from slight to extreme change. This paper has described a 3-descriptor scheme and learning algorithm, in an intelligent news filtering system called *Alipes*, to tackle this very important issue. By treating separately the long-term and short-term interests, and handling carefully the interaction between positive and negative interests in short-term interest, the scheme is able to adapt quickly to large changes of interest, and handle exceptions of interests within the broader scope of an interest category. Our experimental evaluation demonstrated the effectiveness of this scheme, which outperforms that of a single-descriptor model.

6 Acknowledgement

We would like to thank Dr. James Wall, LTC Robert J. Hammell, Jianwen Yin, Magy Seif El-Nasr, Linyu Yang, Anna Zacchi and Chris Standish for fruitful discussion. This research was in part supported by ARL project number DAAL01-97-M-0235.

References

[1] Allan, J. 1996. Incremental Relevance Feedback for Information Filtering. In *Proc of the 19th Int'l ACM-SIGIR Conf on Research and Development in Information Retrieval*, 270-278.

[2] Balabanović, M. 1997. An Adaptive Web Page Recommendation Service. In *Proc of the 1st Int'l Conf on Autonomous Agents*, 378-385.

[3] Balabanović, M. 1998. Learning to Surf: MultiAgent Systems for Adaptive Web Page Recommendation. *Ph.D. dissertation*, Dept. of Computer Science, Stanford University.

[4] Billsus, D. and Pazzani, M. 1999. A Personal News Agent that Talks, Learns and Explains. In *Proc of the 3rd Int'l Conf on Autonomous Agents*, Seattle, WA.

[5] Chen, L., and Sycara, K. 1998. WebMate: Personal Agent for Browsing and Searching. In *Proc of the 2nd Int'l Conf on Autonomous Agents*, 132-139.

[6] Deerwester, S. et.al 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41:391-407.

[7] Lang, K. 1995. NewsWeeder: Learning to Filter Netnews. In *Proceedings of Machine Learning Conference*, 331-339.

[8] Lam, W., Mukhopadhyay, S., Mostafa J. and Palakal, M. 1996. Detection of Shifts in User Interests for Personalized Information Filtering. In *Proc of the 19th Int'l ACM-SIGIR Conf on Research and Development in Information Retrieval*, 317-325.

[9] Moukas, A. and Zacharia G. 1997. Evolving a Multi-agent Information Filtering Solution in Amalthea. In *Proc of the 1st Int'l Conf on Autonomous Agents*, 394-403.

[10] Pazzani, M. and Billsus, D. 1997. Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*, 27:313-331.

[11] Salton, G., and McGill, M. J. 1983. *Intr to Modern Information Retrieval.* New York: McGraw-Hill.

[12] Sheth, B. D. 1993. A learning Approach to Personalized Information Filtering. *Master thesis*, Dept. of Electrical Eng. and Computer Science, MIT.

[13] Tan, A., and Teo, C. 1998. Learing User Profile for Personalized Information Dissemination. In *Proc of Int'l Joint Conf on Neural Network 1998*, 183-188.

[14] Widmer, G., Kubat, M. 1996. Learning in the Presence of Concept Drift and Hidden Contexts. *Machine Learning*, 23(1):69-101, Kluwer Acad. Publ.

[15] Widmer, G. 1997 Tracking Context Changes through Meta-Learning. *Machine Learning*, 27(3):259-286, Kluwer Academic Publisher.

[16] Widyantoro, D.H, Yin, J., Seif El-Nasr, M., Yang, L., Zacchi, A. and Yen, J. 1999. Alipes: A Swift Messenger in Cyberspace. In *AAAI'99 Spring Symp on Intelligent Agent in Cyberspace*, 62-67.

[17] Widyantoro, D.H. 1999 Learning User Profile in Personalized News Agent. *Master Thesis*, Dept. of Computer Science, Texas A&M University.

[18] Wiener, E., Pederson, J. and Weigend, A. 1995. A NN Approach to Topic Spotting. In *4th Symp on Doc Analysis and Inf Retrievel*, Las Vegas, NV.