Alipes: A Swift Messenger in Cyberspace

Dwi H. Widyantoro, Jianwen Yin, Magy Seif El Nasr
Linyu Yang, Anna Zacchi and John Yen
Department of Computer Science
Texas A&M University
College Station, TX 77844-3112

Abstract

Finding relevant information effectively on the Internet is a challenging task. Although the information is widely available, exploring Web sites and selecting the right document are still considered a time consuming and tedious task. As a result, many software agents have been launched in cyberspace to perform autonomous information gathering and filtering on behalf of its users. One of the important issues for these agents is to adapt to the changing interests of the users. In this paper, we present *Alipes*, an Information Brokering Agent that learns user's interests and provides personalized news articles retrieved from the Internet.

Introduction

The number of information sources available on the Web is growing by the minute. The more the Internet grows, the more difficult it is to find the information we are looking for. Although search engines facilitate finding Web sites related to given keywords, it cannot proactively deliver information to users on a regular basis. A major challenge in cyberspace is to automate delivering relevant information to individual users.

In such a system, a capability to model and learn user profile is an essential feature. The capability is embedded into a software agent that allows the agent to perform its task on behalf of its users. The agent then learns the user interests through user explicit and implicit feedback. The explicit feedback from the user provides the agent with an accurate information about the user's needs and interests, while the implicit feedback alleviates the user's burden on such a task. Additionally, the learning capability should be able to address the nature and the dynamics of the user's interests.

A number of software agents have been developed to tackle this very important point. We will review these agents in detail in the next section. Through the third and fourth section, we will present our proposed model of a new agent named *Alipes* [1]. *Alipes* was developed to model the user using an evolving user profile. A list of news articles is recommended to the user each time he/she signs on. An evaluation technique and its results will be presented in the fifth section. Before conclusion, a summary of current and future work will be provided.

[1] *Alipes* is an alias of Hermes, a God of commerce and invention in Greek mythology, who also served as messenger, scribe, and herald for the other Gods.

Related Work

Intelligent agents or personal assistants have been developed at MIT's Media Lab. For instance, *NewT* uses keyword-based filtering and machine learning (relevance feedback and genetic algorithms) to personalize the presentation of Usenet news (Sheth 1993). *Amalthea* is an artificial ecosystem of evolving information-filtering and discovery agents that cooperate and compete in a market-like environment (Moukas and Zacharia 1997). Additionally, *Letizia* is a user interface agent that assists a user browsing the World Wide Web by learning the user's interests and scouting ahead from the user's current position to find Web pages of possible interests (Lieberman 1995). There are much other relevant efforts within the distributed AI and the multi-systems community.

Another similar system is *INFOS* (Intelligent News Filtering Organizational System) which is developed by Mock. INFOS used feature vectors to represent user's interests (Mock 1996). The feature vectors are manipulated using keyword-based and knowledge-based techniques. Chen developed *WebMate*, an agent that helps users to effectively browse and search on the web (Chen 1998). *WebMate* keeps track of user interests in different domains through multiple TF-IDF vectors (Salton 1993). The domains are learned automatically as users give positive examples. A more similar system, *Fab*, was also developed by Balabanović. *Fab* is an adaptive system for Web page recommendation (Balabanović 1997). The initial version of this system uses single feature vector weighted using TF-IDF method to represent its user's profile. In the more recent version, the system has been extended to allow multi-topic user interests (Balabanović 1998).

Most of the above works use feature vectors as the representation of user interests and represent an interest category with a single-descriptor approach. The advantage of using single-descriptor model for interest cat-

egory representation is a simple update scheme. However, it also has some disadvantages. In learning a user's feedback, the scheme can either easily forget the previous learned interests or be hard to learn a new interest. Consequently, long-term and short-term learning cannot be handled simultaneously. Furthermore, due to the simple structure, it fails to represent different levels of interest within a broader scope of interest category. A new scheme incorporated in *Alipes* addresses the problem by introducing 3-descriptor interest category representation for modeling and learning user profile.

Alipes: An Information Broker Agent

The goals of *Alipes* are as follows:

- Providing personalized information (news, articles, etc.) for each user.
- Searching proactively and collecting relevant information from the Internet.
- Keeping track of the users' interests actively.

As a software agent, *Alipes* is supposed to recommend information timely and accurately to its users. It learns the users' information needs and interests from users' explicit and implicit feedback.

Design

Alipes is a multi-agent system. The learning and problem solving capability should be taken into consideration. Based on the requirements of the system we considered several design alternatives.

First, the system could be designed as a fully distributed or a centralized controlled system. If the system is designed fully distributed, the communication among agents will be very complicated. Therefore, a coordinator agent can be designed as the control *facilitator* among agents, which simplifies agent communication and facilitates on-line update of the Web interface. However, the coordinator can become a bottleneck of the whole system, which will reduce the reliability of the system. Multiple coordinator agents can potentially alleviate this problem.

Second, the learning capability can be either embedded into each agent or designed as a learning agent. Embedded learning capability in each agent reduces the communication overhead, and makes the learning capability specific to a particular learning task, thus make it easy to implement. However, that could duplicate the learning capability among agents. On the contrary, a centralized learning agent reduces the learning requirements of all the other agents. However, the communication overhead is increased and the learning task becomes more complicated.

Third, the problem solving capability can be either distributed among each agent or designed as a function of coordinator agent. Distributed problem solving can be tailored to each agent's need, but it will cause the redundancy of problem solving capability among agents.

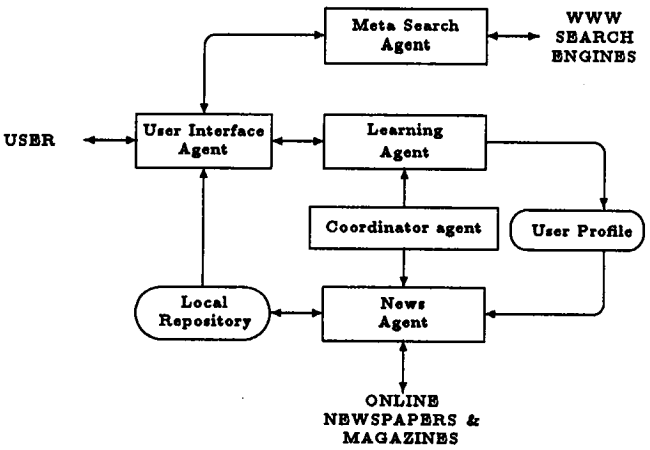

Figure 1: The *Alipes* Architecture

On the contrary, centralized problem solving capability in the coordinator agent will simplify the design of the other agents, but make the coordinator agent much more complicated.

In our design, we use two coordinator agents for the coordination and synchronization of work among agents in the system. Additionally, we use a dedicated learning agent that is centralized in the system. To reduce the potential problem in a centralized system, communication among agents is performed in a distributed manner.

Multi-Agent Architecture of Alipes

Figure 1 shows the multi-agent architecture of *Alipes*. It consists of five different agents: the news agent, the learning agent, the meta search agent, the coordinator agent, and the user interface agent. All agents reside on the Web server except the user interface agent that is partly on the client site. Communication among agents is handled in a distributed manner. However, two agents serve as the coordinators for the synchronization of agents activities in the whole system. One agent is an explicit coordinator agent, and the other is embedded in the user interface agent.

The news agent explores and collects Web pages from online newspapers or magazines. The activities of this agent are controlled by the coordinator agent. The coordinator agent determines when to start collecting and from which online newspaper Web pages should be retrieved. After all pages have been retrieved, those pages are filtered out and new user's news articles ranking are rebuilt for each user.

The learning agent builds and maintains a user model. Most of the activities of the learning agent are triggered by the user through the user interface agent as the user provides a feedback to the system. Some others are triggered by the coordinator agent when the system needs to learn implicit feedback. For the learning agent, the user interface agent serves as the sec-

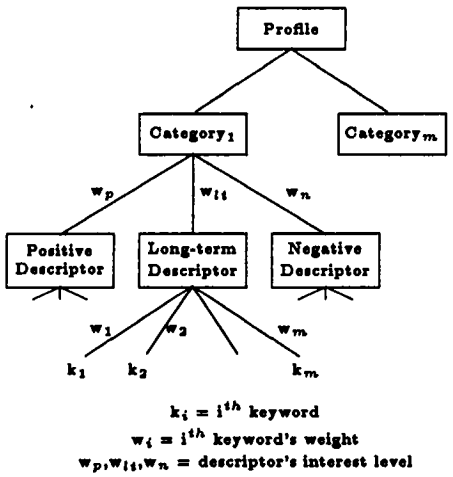

Figure 2: A 3-Descriptor Representation of User Profile

ond coordinator so that the communication overhead on the explicit coordinator agent is reduced. The learning agent communicates with the news agent through shared user profiles.

The meta search agent provided in this architecture allows its users explore new interests. It uses several existing search engines (e.g. Yahoo, Lycos, HotBot, Altavista and Excite) to search. The user can open the Web page provided by the meta search agent and give explicit feedback to the system to learn the content of the page.

The user interface agent serves as the interface between the user and the system. It also serves as the coordinator for the collaboration among the learning agent, the meta search agent and the news agent, which reduces the communication overhead of the whole system.

Learning the Dynamic of User Interests

Learning user interests in *Alipes* is derived intuitively by imitating human personal assistants doing the same task. Users may have several interest categories with different level of interests at the same time. Some of these interests can change dynamically in a short time while some others are long-term interests that are reluctant to change. Furthermore, in a broader scope of interest category, exceptions sometimes occur.

We used a 3-descriptor scheme for the representation of an interest category (see Figure 2). With this schema, an interest category is described by three descriptors, each for long-term, positive and negative descriptor respectively. Long-term descriptor maintains interests built in the long run, while other two descriptors are used to handle short-term interests. Positive and negative descriptors are formed by learning from positive and negative feedback respectively. Long-term descriptor learns from both types of feedback. Each descriptor in the interest category is composed of a feature vector that describes the area of interest, and a descriptor's weight that describes the level of interest.

Learning is performed by manipulating the feature vector and the interest level of each descriptor in an interest category. Given a positive feedback, the feature vector of positive descriptor is modified by incorporating the contribution of the feature vector of the document to be learned. Further, the interest level of the positive descriptor of the corresponding interest category will be increased according to the learning rate. At the same time, the interest level of the negative descriptor will be decreased. However, in addition to the learning rate, the amount of decreasing in the interest level depends on the similarity of the document to be learned and the feature vector of the negative descriptor. On a negative feedback, similar process will be performed but in the opposite direction. For both positive and negative feedback, the interest level of long-term descriptor will be increased or decreased according to the type of feedback.

The updating of the descriptor feature vector is a linear combination of the descriptor feature vector and the feature vector of the document to be learned. For the positive and negative descriptor, the contribution of each feature vector is determined by the learning rate factor that accompanies the feedback. Higher learning rate will cause more contribution from the feature vector of the learned document, and vice versa. In the long-term descriptor, however, the contribution of each feature vector is determined by the inverse of the number of documents to be learned so far.

The interest level of a descriptor is modified by employing an appropriate function. In the long-term descriptor, bipolar sigmoid logistic function is used to update its interest level. The function is chosen to govern the behavior of interests that is reluctant to change in the long run. Meanwhile, increasing the interest level on the positive or negative descriptor is linear based on its current value and the learning rate factor. Decreasing the interest level of those descriptors also incorporates the similarity of the feature vector of those descriptors with the feature vector of document to be learned.

Implementation

This section discusses the functionality of the system in general and details the implementation of each agent in the system. A brief discussion on the security issues will also be described in the last subsection.

System's Functionalities

On the main entry as shown in Figure 3, *Alipes* has a set of news topics (e.g. technology, sport, weather etc) for general users. Each topic contains a list of related articles ranked based on the similarity of the article with the topic assigned. These topics are system's predefined profiles whose structure is the same as the one on the system's user profile. On this page, users cannot

Figure 3: The Main Entry of *Alipes*

Figure 4: The Example of a Personalized Page

give feedback about the news presented because these are provided for general users with various interests. However, registered users may use the articles to provide feedback to the system so that the system is able to learn their interests. Another functionality on the main entry is the meta search engine which searches information from World Wide Web through different types of search engines available on the Internet. The rest area of the main entry is used for customizable interesting related links. Additionally, new users through this page can register to obtain an account; and registered users can sign on to open their personalized news page.

Once a registered user has logged on, a personalized user Web page is presented to him/her as shown in Figure 4. The Web page contains a list of titles along with sample sentences extracted from earlier paragraphs of the corresponding article. The article titles are ordered according to the similarity of the article with the user profile. Right next to the title is a number representing the systems prediction about the interest level of its user if given the article. The top of the page contains pointers to other news articles list of pre-defined topics as well as some utilities. The utility allows users to display their profile or ask the system to rebuild their personalized page based on the latest profile. Just before the first article title, an input form is provided to search Web pages using meta search engines. In this prototype, the existing search engines used are Lycos, Excite, HotBot, Yahoo, and Altavista. Users can use either one of these search engines or all of them.

When a user reads an article, a list of pre-defined user comments are provided on top of user window where the user can give explicit feedback about the article they read. Options for explicit user feedback are as follows:

1. Strong positive feedback ("I like it, always show me article like this")
2. Moderate positive feedback ("Interesting")
3. Weak positive feedback ("Not Bad")
4. Moderate negative feedback ("Not interesting")
5. Strong negative feedback ("Never show me article like this")

In addition to the above feedback, a negative implicit feedback will be automatically sent by the system for a category that its corresponding articles are not read by the user for a period of 24 hours.

Agents Implementation

News Agent The news agent retrieves periodically news articles from online newspapers and magazines. It runs as a background process and the information gathering process is initiated by the coordinator agent. The coordinator agent provides the information about the sites that need to be explored.

Given an entry site of an online newspaper or a magazine, the document retrieval starts from depth level zero and follows document links. Link following process is complicated by the various types of link protocols such as Html document, image, mail, ftp, Java script, etc. In this case, only a link to an Html document is extracted from a document. Once a document has been retrieved and the links on the document have been extracted, a quick check is performed to determine whether the document contains a news article. A simple heuristic is applied by detecting the existence of a somewhat long paragraph in the document. If the document does not contain news article, it will be removed. The document retrieval process is then proceeded in depth first search with limited depth until no more sites can be explored.

After all documents from an online source have been retrieved, the next step is to extract the content of each retrieved document. First, the Html tag, Java script as well as the information within a link are removed. The document's title and sample paragraph are extracted, and the rest are converted into plain text file format. The next stage is to convert the plain text news article into a document feature vector. In this step, common words (e.g. and, or, thus etc.) are removed using a stop list. We use 293 words in the stop list. Words are stemmed [2] and their frequency are counted. In this

[2] We use Porter's stemming algorithm for this purpose

stage, the occurrence of two-word phrases is identified by detecting the occurrence of the same two successive words at least twice during word counting. The feature vector of the document is then obtained by computing the keywords' weights using the TF-IDF method (Salton 1993) and normalizing the weights. The feature vector, document's title, sample content from the first somewhat long paragraph, and document's URL address is then saved as a meta document.

The last stage performed by the news agent is to score and to rank each document according to the profile of its users. For each user, the score of each document is calculated based on its profile. The top 30 documents are kept and ordered in decreasing document score value. This information will be used to generate personalized news article as the corresponding user signs on to open his/her personalized page.

Learning Agent The main task of the learning agent is to learn the user profile and to adapt to the changes of user interests. Explicit positive or negative feedback is obtained from user's rating on the document. Implicit positive feedback is sent by the user's browser as the users open a document on their personalized page. Learning negative implicit feedback is performed once a day at the midnight and is initiated by coordinator agent.

For a new user, the initial profile of the user is created based on the information provided during registration process. In the registration form, users are given a list of interests to be chosen as an option. The options of interest include various types of sport, money related topics, technology, weather, health, and world news. For each of the interest categories, the system has the corresponding feature vector that is obtained by averaging the feature vector of documents in the category. After the form has been submitted by the user, his/her personal information is saved in Oracle database and an initial profile is created in a separate file.

On learning explicit feedback, the learning rate of strong positive and negative feedback is set to 0.9. This value will ensure that after learning the feedback, similar documents on the next information filtering session will be scored high for positive feedback or low for negative feedback. The learning rate on a moderate positive and negative feedback is set 0.5 to allow moderate changes to the user profile. A learning rate of 0.2 is set for weak positive feedback and a very slight change will be affected by this feedback.

Meta Search Agent Searching through a meta search engine is carried out by sending search commands to a search engine available on the Internet and parsing the search results. Five different search engines are used for the construction of meta search engines including Yahoo, Altavista, Lycos, Excite and Hotbot.

Each of the search engines has different syntax on its search query and different format on its search results. Consequently, each of them requires separate routine to contact and to parse the search output. After the title and the URL's address are extracted from the search engine's output, a customized search result is created. A small code to activate feedback generation is inserted on every meta search agent's result.

Through the user interface agent, users are given choices to use one of the provided search engines or all of them. If only one search engine is used, the corresponding search engine will be contacted and the results are presented to the users. If the users choose to use all search engines, processing a query on each search engine is performed and the results are merged.

User Interface Agent On the server site, it creates an Html interface on the fly to convey information and messages from other agents to its user and vice versa through the Internet browser (e.g. Netscape, Internet Explore, Mosaic etc). It also redirects messages sent through the Internet from its user to an appropriate agent in the system. On the client site (e.g. user's desktop), the user interface agent provides a means for its user to send a message for a particular actions (e.g. giving feedback, reading news etc.).

Database and Security Issues

All of the users' profiles are stored in an Oracle database. We also create the internal representation of the profiles in text files for learning and information-filtering. Since someone could invade Oracle database and destroy the entire database, it is important to consider the security issues. Therefore, we provide user's login and password as described earlier to protect our database. More solutions will be implemented in the future to provide even high-level of security in our system.

- Introducing the time out mechanism. Whenever a user is on-line for some time which is longer than a threshold and he/she doesn't do anything, we force the system to log him/her out.
- DES will be used as the standard for encryption to prevent the disclosure of information.
- There is a privacy issue when a user try to retrieve something interesting. Sometimes, users do not want other people (including the system administrator) to know what he/she is interested in. To provide this kind of privacy service, we are planning to localize the privacy related user profile by moving them into user's own machine. This information will be transferred to the server whenever it is necessary and will be deleted right after that. In this case, the communication overhead will be improved to certain degree, which we still do not know yet.

System Evaluation

Experimental evaluation has been conducted to measure the performance of our agent to model and to learn user interests. In this experiment, artificial users are used instead of human users. Rather than retrieving the documents directly from the Internet that can

take a long time for each of the evaluation, we build our own document collection. We also use an experiment scenario that simulates the document collection as information sources on the Internet. The documents used in the experiment are articles in Html format retrieved from 12 different online newspapers and magazines (e.g. USATODAY, USNEWS etc.) at different times. The collection contains 1427 documents comprising of 6 different general topics: world news, money, health, weather, technology, and sports.

We have observed the accuracy of our agent at various threshold values. The threshold is a system parameter that determines the proliferation of a new interest category when the system learns from feedback. Measuring accuracy involves filtering a set of documents using a learned profile and then comparing to those selected by the target profile. By setting the threshold to 0.05, 0.25, 0.45 and 0.65, and using 90 highest weighted keywords, the system's accuracy are 44.7%, 53.4%, 52.8% and 49.9% respectively. The results are averaged over 28 trial representing 28 users with different profiles. The same pattern is obtained when we use 40 keywords, where the performance peak is at a threshold of 0.25. These results are surprising because the system's accuracy tends to degrade at higher threshold values.

Future Work

The main feature of *Alipes* is its capability to model and to learn changing user interests. Our current evaluation indicates that the approach is feasible. We plan to extend our work in two directions in the future.

First, We are currently extending the agent's capability to perform collaborative filtering. In addition to the use of content-based filtering in current system, social filtering will be incorporated in this system to benefit the experience of other users. Various existing techniques on this type of filtering will be investigated for the potential adoption for use in our system. Another alternative to develop a new scheme of social filtering will also be considered for this purpose.

Second, We plan to extend the capability of the system to learn and to explore information sources so that the retrieval process can be done effectively. With this capability, the agent is expected to be able to identify useful sites for future use and to avoid exploiting the ones that could not provide interesting information based on its user profile. This is a very challenging issue due to the fact that the information sources on the World Wide Web can be heterogeneous in their domain expertise, unstructured and are subject to change over time.

Conclusion

In conclusion, the results shown from the experimental evaluation demonstrate that our scheme on dynamic modeling and learning user profile is feasible to model the evolving user interests through learning and feedback. However, as it was pointed out in the future work section, much work still needs to be done to realize our vision on this research. Although *Alipes* may have many limitations, it is a good starting point to step ahead. We think it may lead the way to a greater success, as it had done through its history as a Greek God.

Acknowledgement

We are in debt to Mr. Bobby Duncan for providing his fund donation to Center for Fuzzy Logic, Robotics, and Intelligent Systems at Texas A&M University. We also would like to thank Dr. James Wall at The Texas Center for Applied Technology and LTC Robert J. Hammell at Army Research Laboratory (ARL) for fruitful discussions. This research was in part supported by ARL project under contract number DAAL01-97-M-0235. The views contained in this paper are those of the authors and should not be interpreted as representing the official policies of the sponsoring organizations or agencies.

References

Balabanović, M. 1997. An Adaptive Web Page Recommendation Service. In *Proceedings of the First International Conference on Autonomous Agents 1997*, 378-385. New York. N.Y.: ACM.

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

Chen, L., and Sycara, K. 1998. WebMate: Personal Agent for Browsing and Searching. In *Proceedings of the Second International Conference on Autonomous Agents 1998*, 132-139. New York. N.Y.: ACM.

Lieberman, H. 1995. Letizia: An Agent That Assists Web Browsing. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence 1995*, 924-928. San Mateo, Calif.: M. Kaufmann.

Mock, K. J. 1996. Hibrid-Hill-Climbing and Knowledge-based Techniques for Intelligent News Filtering. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, 48-53. Menlo Park, Calif.: AAAI Press.

Moukas, A. and Zacharia G. 1997. Evolving a Multiagent Information Filtering Solution in Amalthea. In *Proceedings of the First International Conference on Autonomous Agents*, 394-403. New York, N.Y.: ACM.

Salton, G., and McGill, M. J. 1993. *Introduction to Modern Information Retrieval.* New York. N. Y.: McGraw-Hill.

Sheth, B. D. 1993. A learning Approach to Personalized Information Filtering. Master thesis., Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.