A Multi-Agent System that Facilitates Scientific Publications Search

Aliaksandr Birukou *
Department of Information and Communication Technology
University of Trento - Italy
birukou@dit.unitn.it

Enrico Blanzieri
Department of Information and Communication Technology
University of Trento - Italy
enrico.blanzieri@unitn.it

Paolo Giorgini
Department of Information and Communication Technology
University of Trento - Italy
paolo.giorgini@unitn.it

ABSTRACT

It is very difficult for beginners to define and find the most relevant literature in a research field. They can search on the web or look at the most important journals and conference proceedings, but it should be much better to receive suggestions directly from experts of the field. Unfortunately, this is not always possible and systems like CiteSeer and GoogleScholar become extremely useful for beginners (and not only). In this paper, we present an agent-based system that facilitates scientific publications search. Users interacting with their personal agents produce a transfer of knowledge about relevant publications from experts to beginners. Each personal agent observes how publications are used and induces behavioral patterns that are used to create more effective recommendations. Feedback exchange allows agents to share their knowledge and virtual communities of cloned experts can be created to support novice users. We present also a set of experimental results obtained using CiteSeer as a source of information, that show the effectiveness of our approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering, Relevance feedback, Search process*; I.2.6 [**Artificial Intelligence**]: Learning—*Knowledge acquisition*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*

General Terms

Design, Experimentation

Keywords

Multi-agent system, learning agents, personal agents, publication search, CiteSeer, learning from observations

*The primary author of the paper is a PhD student

AAMAS '06 May, 2006, Hakodate, Japan.

1. INTRODUCTION

The Web is a huge source of information where it is possible to find publications and scientific literature. However, the majority of papers are usually irrelevant to individual researchers. Beyond conventional search engines, users need specific tools and methods for an effective use of all the available scientific resources. For example, special search engines for scholar articles, like [7], have recently gain a lot of popularity among professionals and specialists. Online libraries like CiteSeer [9] and ACM digital library [16] for computer science are more and more used by researchers to search and download papers that are relevant to their scientific area. These services provide search and browsing facilities based on the list of references of the papers. However, for a novice researcher it is still very hard to determine which papers are really significant among those available. It should be very nice for a beginner to receive suggestions directly from experts of the field. An expert knows which papers are more influential, which papers were published at top conferences and in prestigious journals, what is the state-of-the-art and so on. There are a number of implicit factors an expert takes into account when determines if a paper is relevant or not. But, unfortunately, we cannot assign a personal expert for each area of interests to every novice researcher. This calls for systems able to facilitate scientific publication search.

Among the proposed solutions for the scientific publication search, personalization systems [4, 8] focus on the need of a certain user, maintaining his/her profile and notifying him/her periodically or on-demand about papers related to the user's interests. The main goal of these systems is to filter information flow delivering to the user only items related to his/her interests. Differently, recommendation systems [1, 12] usually exploit the knowledge obtained from different users of the system to generate useful recommendations. Many recommendation systems [11, 12] use collaborative filtering algorithms [14] to compare similarity between users in order to recommend to the target user previously unseen items contained in the profiles of similar users. To some extent, currently developed systems mitigate search of the relevant literature for the person that is a novice in the field. However, they have several shortcomings. For instance, there exists a lack of expert knowledge, in case all the users are novices. Pro-active recommendations sometimes turn into a heavy flow of potentially interesting items. Finally, most of the systems require explicit feedback, which is not always convenient and often discouraged by users.

In this paper we present a multi-agent system that facilitates scientific publications search. The architecture of the

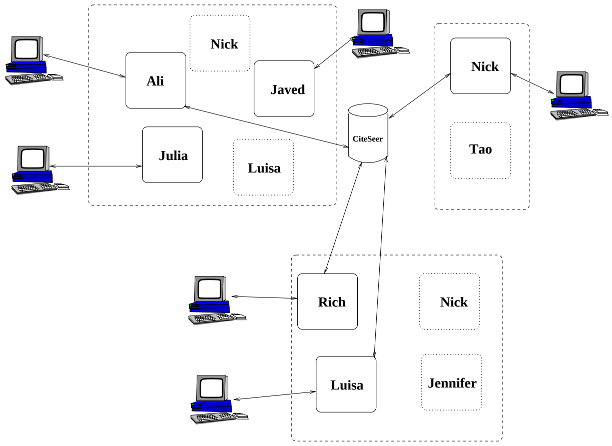

Figure 1: An example of several platforms with three ad hoc communities. Cloned expert agents are depicted with dotted squares. Julia, Ali and Javed also form an organizational community.

system is distributed and includes several multi-agent platforms. Each platform represents an organizational community of users with similar research interests. Alternatively, virtual communities of cloned experts can be created to facilitate the search of the papers relevant to a specific topic. As said above, we cannot assign an expert to each novice, but we can assign to a novice user an agent which has access to the knowledge of an expert, represented in other agents. Users interact with their personal agents which cooperate and produce a transfer of knowledge about relevant publications from experts to beginners. Agents in our system are based on Systems for Implicit Culture Support [3] to learn behavioral patterns by observing how publications are used, namely which papers users cite, download, etc. Agents use learnt patterns and exchange feedback to produce personal recommendations more effectively. The system also provides an opportunity to share opinions and information contained in single bibliographies. It is possible to have an external information source, e.g. CiteSeer, to obtain citation graph and to download desired documents.

Although personal internet assistant agents have been developed for more then ten years [6, 10], the idea of using agents to support search of scientific publications is relatively new (see e.g. [8, 12]). Unlike most currently developed systems we use implicit feedback, avoiding extra work of the users. The framework used for the recommendation creation is general, being able to recommend not only papers, but expert in the research area as well.

The paper has the following structure. Section 2 motivates our approach by examples. In Section 3 we describe the general architecture of the system, whereas experimental results are presented in Section 4. We discuss related work in Section 5 and conclude in Section 6.

2. MOTIVATION

In this section we illustrate the possible use of the system and show the role of agents in the system.

2.1 Motivating Example

Ali is a first year PhD student who studies Data Mining. His tutor Nick has recommended him to have a look on the paper “Privacy-preserving data mining” by Agrawal and Srikant. Ali submits the title and the authors of the paper to the system and his personal agent asks personal agents of his colleagues whether they have the paper in their bibliographies. In our case, the other PhD students, who are working in the same research group with Ali, are Javed and Julia. The platform containing the personal agents of these students is depicted in the upper left part of Figure 1. Julia is the PhD student of the second year and she already has the paper requested by Ali in her bibliography. Her personal agent provides this information in response to the query of Ali's personal agent. Ali receives complete information about the paper, including its location on the Internet, or in the local intranet, and comments from Julia, if of course she wants to give them. Now Ali can download the paper and then cite it in his paper. If the information about the target paper has not been found on the platform, it is possible to contact other platforms or to query different information sources, e.g. CiteSeer.

To describe virtual community creation we introduce Luisa who is Ali's co-tutor, and works on Security. Ali wants to find relevant papers on the intersection of these two topics (Data Mining and Security), but his tutors are very busy. However, they agreed that Ali clones their personal agents. Ali's personal agent creates a virtual community cloning the personal agents of Nick and Luisa. Now, the corresponding expert agents run on the platform with Ali's personal agent. Each agent on the platform contains the information regarding behavior of its user, in particular Luisa's and Nick's citing, reading and browsing behavior. In our case, to find relevant papers on the intersection of Data Mining and Security, Ali's personal agent queries expert agents (Nick's and Luisa's) and shows to Ali the results ranking the papers recommended by both experts higher. The idea of combining Nick's and Luisa's results is to find papers that are relevant for them both. Moreover, being cloned and placed on another platform, the agents of Nick and Luisa adapt themselves to the local community, namely changing their recommendations towards the local common interests, or “culture”.

In Figure 1 three ad hoc communities are depicted. For instance, Luisa and Rich work in the same research laboratory and consider research of Nick and Jennifer to be relevant to their current interests. Therefore cloned agents of Nick and Jennifer are running on the platform. Nick works with Tao and has its expert agent on the platform.

2.2 The Role of Agents

In our architecture, agents do the “dirty” work for their users. They perform bibliography maintenance, answer queries asked by other agents, and provide links where to download the desired papers. Among all this, they also represent the means of knowledge transfer from more experienced users to beginners.

Autonomy and mobility are capabilities that we exploit when we clone agents. Running in a new environment (the ad hoc community), the cloned agent uses the past experience of its user and its original community to support and interact with the new community. The main idea here is that all the agents learn the "culture" of the community — i.e., knowledge about typical behavioral patterns of the members of the community — that can be used to select the

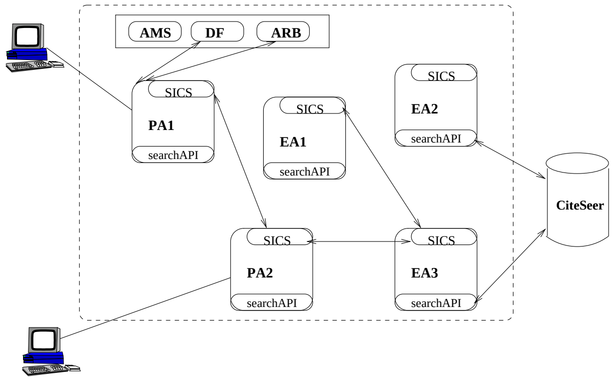

Figure 2: The architecture of the single system node. ***Users*** **of the system node query their** ***personal agents*** **on the platform. Personal agents contact one another to find papers on a specified topic. They also use a search API to contact different information providers, e.g. CiteSeer. SICS module inside the personal agent filters and re-ranks collected information. There are also** ***Expert agents*** **running on the platform, which are the clones of the personal agents of the experts on the topic.**

relevant literature.

3. SYSTEM INTERNAL ARCHITECTURE

The system consists of several multi-agent platforms, or *system nodes*, which are able to communicate with one another. A single platform runs on a single computer and comprises one or several personal agents and several expert agents. It represents a community of users with similar research interests, e.g. personnel of a research lab or PhD students working in the same research field. Each user is assisted by a personal agent that helps to find relevant publications. The number of personal agents on the platform corresponds to the number of users of a system node. An expert agent is a clone of the personal agent of an expert user. Its task is to share its experience with other agents supporting them in their search.

The presence of several platforms where users are grouped by research interests helps us to handle scalability issues. Moreover, personal and expert agents running on the same platform and assisting users with similar interests are able to adapt to the specific needs of the community and to produce more valuable suggestions.

The system nodes are multi-agent systems implemented with Java Agent DEvelopment framework (JADE). The overall architecture of such a platform is depicted in Figure 2. It is composed of users, personal agents, expert agents, and external information sources like CiteSeer. There are also several auxiliary JADE agents. We describe system actors in more details in the next subsection.

3.1 System Participants

- A *user* accesses the system using an html/php interface on the client side. It submits a query to the system, which can be either a *paper request* or an *information request*. In the case of the paper request, the user specifies the author(s) and the title of the paper. For the information request the user has to specify the topic he/she is interested in.

- A *personal agent* is a software agent running on the server side. The task of the personal agent is to assist the user in searching scientific publications. In case of a paper request it searches for the desired paper on the platform or queries an external information source using searchAPI. If the user submits an information request about a specific topic, then the personal agent starts the interaction with other personal and expert agents on the platform. The SICS module of the agent processes past actions, e.g. reading and citing, of its user and, partially (using feedback exchange), the actions of other users in order to find papers relevant to the given topic. The next subsection describes SICS module in more detail.

- An *expert agent* is is the clone of an expert's personal agent. It can represent, for instance, an author that is considered to be one of the top researchers in a certain area. In the best case, it represents a person and has information about how such a person usually browse the web in searching for papers and which papers are eventually downloaded and read. In the worst (but anyway, quite good) case, the agent has knowledge about the use of citations by its user. This knowledge is extracted from the list of publications of the author. The papers that the expert agent suggests are considered to be of extreme importance and to form the state-of-the-art on the topic. It is also possible for the cloned agent to update its knowledge base, contacting the original personal agent.

- By an *external information source* we mean any database containing scientific publications and offering some kind of API to access them. For the experiments we have used data from CiteSeer, but we could have chosen Google Scholar or any university or public library index.

- A *wrapper* is an agent which deals with those information sources that have no API. Typically, each information source requires a tailored wrapper dealing only with this particular source. The presence of wrapper(s) on the platform is optional.

- An *Agent Management System (AMS)* exerts supervisory control over the platform. It provides agent registration, search, deletion and other services. It is an internal JADE agent running on every platform.

- A *Directory Facilitator (DF)* holds the list of services that are provided by an agent and provides a set of agents that offer a specific service. It provides agents with other personal/expert agents' IDs. In our case both personal and expert agents have the service "PaperRecommender" that defines the agent's ability to answer queries about scientific articles.

- An *Agent Resource Broker (ARB)* provides a link between the current platform and other platforms. Using this link, the agents can propagate requests of their users to different platforms. Contacting different system nodes that correspond to different research groups

Table 1: The actions that can be observed by the system

action	object	attributes
participate	conference_name	paper, session_name, topic
publish	journal_name	paper, special_issue_title, topic
cite	paper	paper, topic
download	paper	topic
view_details	paper	topic
reject	paper	topic

it is possible to collect responses that represent different points of view.

3.2 Internal Agent Structure

Since expert agents are cloned personal agents, they have exactly the same structure. The agents have certain beliefs, e.g. database of past user actions or bibliography, and capabilities, in particular the searchAPI and the SICS module. A *bibliography* is collection of bibitems, each representing a single publication. It is stored as .bib file and can be used in LaTeX. A *searchAPI* is an API that agent uses to query an information source. Basically, it is a library that contains a set of functions to query the information source. It is dedicated to the specific source and is usually provided by the source owner.

3.2.1 SICS

SICS stands for System for Implicit Culture Support. It is responsible for recommendation creation process. It consists of the three modules: the *observer* that records users' actions, the *inductive module* that applies data mining techniques in order to induce a theory about these actions, and the *composer* that produces answers to the query by using the information saved by the observer and produced by the inductive module. The observer can partially collect information about actions of other users. It is possible due to an exchange of feedback that we describe in the next subsection. We refer the reader interested in more details of SICS module to the paper of Blanzieri et al. [3].

Every *action* in our terminology has an *agent*, which is the person that executes the action, e.g. name of the user executing the action; an *object* which participates in the action, e.g. paper; and a set of *attributes*, which are the features of the actions that can be useful for its analysis, e.g. session name for the action of participating to a conference. In our application, SICS module analyzes the actions presented in Table 1. An *agent* is always an author; therefore we omit it in the table. Every action is recorded being executed with respect to a certain topic. To determine topics related to an article, we can use the list of keywords given in the article itself. Furthermore, the keywords that were used to retrieve the article from an information source could be also considered as a list of topics, to which document is relevant.

We explain the information contained in the table in detail:

- *Participate.* An author can participate in a conference, publishing a paper in the conference proceedings. In order to observe this action it is just necessary to know the information about where the paper is published. We use a name of the conference session as an additional indicator of the topic of the paper.

- *Publish.* A paper can be published in a journal, related to a certain research field. Again, we observe this action if we have the information about where the paper is published. In case of publication in a special issue, the name of the special issue is an additional indicator of the topic of the paper

- *Cite.* This action indicates the papers cited by an author in his/her paper. This information is extracted from citation graph.

- *Download.* This action can be performed when an author looks through the list of papers provided by his/her personal agent in response to a query.

- *View.* It indicates that an author clicked on the "view details" link corresponding to one of the papers provided by his/her personal agent.

- *Reject.* It comes with those papers that were recommended by a personal agent, but did not attract any attention (like view or download actions) from the user.

We would like to stress that we use keywords, or topics to separate papers of authors who have several research interests. This solves the problem mentioned by McNee et al. [11] of the irrelevant recommendations received by users who have only one topic in common with an author that has a wide range of research interests.

There is also a problem of interpretation of reject action. For, example, reject from a novice user is more informative, since it can mean that the paper is rejected because it is irrelevant. An expert user can reject the paper just because he/she saw it many times already. We use the past history to analyze these details.

3.2.2 Implicit Culture

The goal of SICS in general is to infer a cultural theory about users' actions and to encourage a novice to behave according to this theory. The cultural theory contains information implicitly hidden in the actions, like "if author A has papers on *agents*, it publishes on $AAMAS$ conference and cites papers of author B". The theory is induced by the inductive module of the SICS. The composer module produces recommendations taking into account the cultural theory and user preferences, encouraging the user to behave consistently with the theory. We would like to achieve consistence here because it is natural to assume that the way the experienced researchers select relevant papers is much more close to the optimal one than the novice's actions. Therefore, if the novice follows the recommendations, his/her behavior will converge to one nearer to optimal. The relationship characterized by this knowledge transfer is called "Implicit Culture".

3.3 Search in the System

There are two types of user requests: a *paper request* and an *information request.*

3.3.1 Paper Request

When a user submits a paper request, he/she knows the title and the list of the authors. This is reasonable, because he/she could have found this information in another article, or on a conference website. The personal agent of the user checks if user's bibliography contains this item. In case it

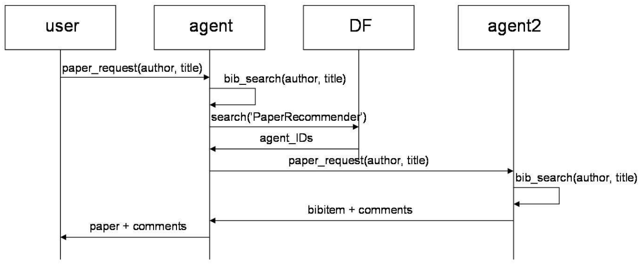

Figure 3: Sequence diagram in response to a paper request.

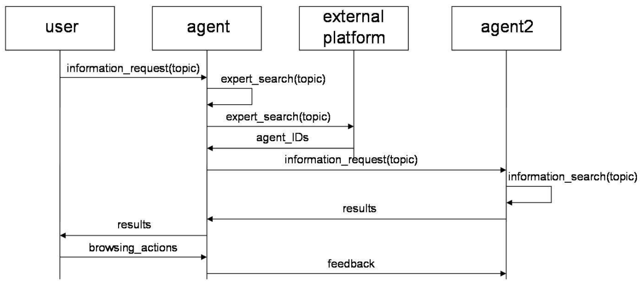

Figure 4: Sequence diagram in response to an information request.

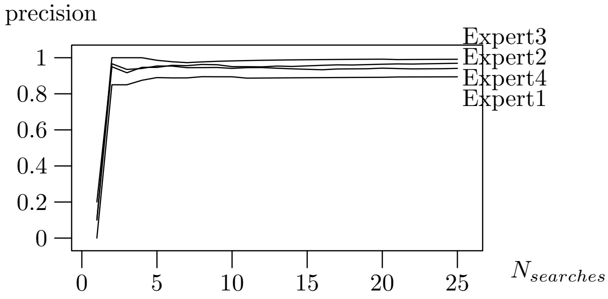

Figure 5: Average precision of 10 simulations with different number of searches.

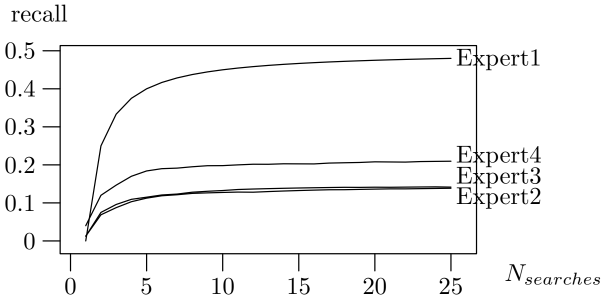

Figure 6: Average recall of 10 simulations with different number of searches.

is present in the bibliography, the system locates the paper and shows it to the user. Otherwise, personal agent uses its previous experience to fill in a list of agents to ask. In case the agent cannot fill the list itself, it contacts Directory Facilitator (DF) which returns IDs of agents that offer service "PaperRecommender" on the platform. The agent contacts every agent in the list, propagating user request. The contacted agents respond with bibitem that contains information and comments of their users. Using the bibitem, the agent locates the paper and shows it and the comments to the user. The sequence diagram corresponding to system actions during the search is given in Figure 3.

However, the requested paper can be absent in the bibliographies of the other users. In this case, user's personal agent contacts different data repositories, like CiteSeer which can be queried from the system. Once the paper is found, the personal agent offers the user either to read the abstract or to download the paper.

3.3.2 Information Request

If a user submits an information request, he/she has to specify the topic he/she is interested in. The personal agent checks if agents of the experts in this topic are present on the platform. If they are on the platform, the agent propagates the user's information request to them. The goal of the virtual community of expert agents is to recommend items that are considered relevant to the topic and to include them in so-called "reading list". To fill it with the items relevant to the topic the agents analyze citation, search and reading behaviors of the community members. Then, the personal agent shows the obtained results to the user. The results of the information request can be supplemented with the results obtained from CiteSeer or from another community.

If there are no topic experts on the platform, the personal agent contacts other agents on the platform, or external platforms in order to find experts in the topic. Note, that here the SICS module of queried agent simulates some kind of reputation scheme by discovering which agent can be an expert. To determine the level of user's expertise it is possible to analyze how often his/her paper is cited. This information can be obtained using CiteSeer. After the list of top experts in the field is filled, the personal agent clones experts' personal agents thus creating an ad hoc community running on the user PC and answering user's queries.

The user browses through a list of results being able to execute a set of basic actions that can be applied to a single item: it is possible to view the details (e.g. abstract) of a paper; a paper can be downloaded or rejected. Information about the actions that user executes (view, download, reject) is recorded by the observer module of the SICS and is also sent to the agents which participated in the search. They use this *feedback* to produce more effective recommendations next time. The personal agent of the user uses this information in further searches too. The sequence diagram of system actions during an information search is depicted in Figure 4.

4. EVALUATION

In this section we present the experimental results obtained using the platform. We also define the measures we have used to estimate the quality of suggestions produced by the SICS.

The aim of the experiment is to check the hypothesis that agents with SICS are able to produce recommendations in accordance with the "community culture" and adapting their suggestions to the needs of a single community. We built a community of several experts, and we built their

Table 2: Profiles of the experts

	Common rank of the paper									
Expert name	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Expert1	0	0	0	0	0	0	0	0.5	0	0.5
Expert2	0.23	0.18	0.14	0.14	0.14	0.07	0.05	0	0.05	0
Expert3	0.19	0.14	0.14	0.14	0.1	0.14	0.05	0	0.1	0
Expert4	0.29	0.29	0.14	0	0.14	0	0.14	0	0	0

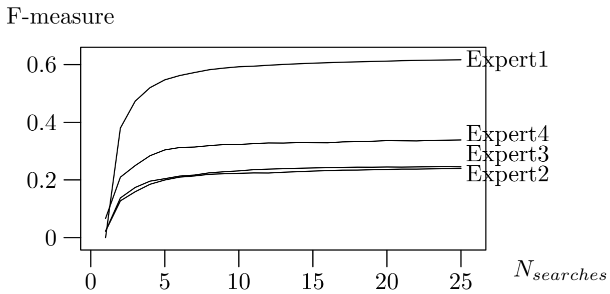

Figure 7: Average F-measure of 10 simulations with different number of searches.

preferences. We expect that personal agent will be able to adapt suggestions to expert preferences.

In our experiment we add in each expert agent a piece of code that replaces behavior of the real user. The main function of this replacement is to generate pseudo-user response to the recommendations. The responses are generated according to a user profile. The user profile determines click-through ratio of the acceptance of the possible results. The dataset contains m results each corresponding to a particular paper. The dataset is used during simulations instead of contacting CiteSeer. The profile determines the probabilities $p(j)$ of choosing the j-th paper, $j \in \{1, \dots, m\}$ while searching for a specified topic. We assume that the user accepts one and only one paper during search, so $\sum_{j=1}^{m} p(j) = 1$. In our experiment, the number of papers m is equal to 10.

We use the following measures in order to evaluate the quality of suggestions:

- We call a paper **relevant** to a topic if the probability of its acceptance, as specified in the user profile, is non-zero.
- **Precision** is the ratio of the number of suggested relevant papers to the total number of suggested papers, relevant and irrelevant.
- **Recall** is the ratio of the number of proposed relevant papers to the total number of relevant papers.
- **F-measure** is a kind of tradeoff between precision and recall. It is calculated as follows:

$$\text{F-measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}.$$

In order to build a small community of expert agents we generate four different profiles, each representing one expert agent. As experts we select the most cited authors, namely authors of the most cited papers that CiteSeer returns for the keyword "Data Mining". To determine papers the expert consider to be relevant, we take the first five of the most cited papers by each expert and look at the papers they cite. The top ten papers they cite all together are considered to be relevant to the selected group of authors. Then, we create a personal profile for every expert. The profile contains the probability that the expert considers the paper from the top ten selected to be relevant. These probabilities are calculated from the expert's citations, as follows:

$$p(j) = \frac{c_j}{\sum\limits_{i=1}^{m} c_i}, \quad j = 1 \dots m.$$

Here c_j is the number of times an expert cites the j-th paper in his/her five most cited papers that we select for the experiment. Of course, some of the selected papers are not relevant to a particular expert, being relevant to a group as a whole. The profiles are given in Figure 2. The information that we need to determine experts and the information about citations is taken from CiteSeer using a specially developed wrapper.

We repeat searches for the keyword "Data Mining" 25 times, measuring precision, recall and F-measure of the recommendations after every search. We model user acceptance behavior as follows: given a keyword, accepted result is generated randomly according to the distribution that is specified in the profile. Other papers obtained from the SICS and agents are marked as rejected. At the end of 25-searches' session the observation data are deleted. We repeat the search sessions several times in order to control the effect of the order of paper acceptance.

The results contain precision, recall and F-measure of the papers recommended by the SICS module and by the agents queried by expert's personal agent. Lines in Figure 5 represent precision of the recommendations that are produced by the four personal agents for the experts, which names are shown on the right part of the figure. In Figure 6 and Figure 7 we have analogous curves for recall and F-measure, correspondingly.

From these figures we can note that the SICS modules of the personal agents are able to track users' interest, adapting suggestions to the needs of the community. The precision keeps on a very high level (due to the high number of relevant papers in our experiments) while recall and, as a result, F-measure tend to increase over time, reaching adequate values just after five searches.

5. RELATED WORK

In a previous paper [1], we applied Implicit Culture concepts to the development of a multi-agent recommendation system for web search. Although also this paper is about web search, the context of application is completely different. The two architecture are similar, but there are some significant differences. Firstly, here we consider a possibility of ad hoc community creation for information filtering, while in the previous work we were more focused on an organizational community. Secondly, we consider a rather wide set of actions here, while in the previous work it was reduced to *request, accept,* and *reject*. Finally, in the case of web search a lot of repetitions, namely searches for the same keyword several times, occur. In the case of scientific papers, the actions tend to transform from several actions of search type to the actions of citation type, which can be repeated many times for good papers or can be performed only once or even never for bad papers.

A CiteSeer extension aimed at personalization of the search is presented in [4]. This paper describes the mechanism of user profiles creation and maintenance. The information in the profiles indicates user interests and can be used to generate a list of potentially interesting papers. The main difference between our architecture and the one presented in the paper is that we propose collaboration of the users that should lead to the more qualitative search results for all the collaborators. Unlike their approach, we do not require any explicit feedback from the user, collecting information by means of observations and implicit inferences.

There are a number of papers presenting different approaches to facilitating scientific literature search. Bradshaw et al. [5] describe Rosetta, a system that indexes papers based on the way they have been described when cited in other articles. Using this information they find the items that correspond to the different variations of the user query, grouping them by topics. GroupLens group in [11] explores the use of collaborative filtering to recommend research papers. Torres et al. [15] combine content-based and collaborative filtering approaches. They have also shown that different algorithms are more suitable for recommending different kinds of papers. Middleton et al. [12] propose an ontological approach to build a recommendation system for scientific articles. The research interests are organized in an ontology according to which papers are classified. The papers corresponding to the classes of the ontology that represents user's interests are recommended to the user. The system records relevance feedback to compute users' interests and to recommend papers using collaborative filtering.

The system proposed in this paper has an architecture that permits to deal with several information sources. We have used the notion of Implicit Culture that is more general than collaborative filtering [2]. Moreover, the use of agents in our approach provides several advantages, such as possibility of sharing bibliographies and creation of virtual communities. Finally, with our framework it is possible to recommend not only papers, but also the experts.

6. CONCLUSION AND FUTURE WORK

In this paper, we have presented a multi-agent system that facilitates scientific publication search and helps in finding relevant papers on a specified topic. Agents applying Implicit Culture concepts observe user behaviors and infer knowledge about users' actions. The system finds relevant papers analyzing citation graph and action history of the user. Personal agents produce via interaction a transfer of knowledge about relevant publications from experts to beginners. The experimental results have proved that SICS is able to adapt suggestions to the user preferences. However, the results are still preliminary and as part of our future work we plan to complete the evaluation.

As a future work we would like to extend the inductive module. For instance, association rules can be used to find papers usually cited together with other papers. Clustering techniques can be applied to distinguish between several topics of the same author. Clusters should contain the papers that deal with similar topics. We also plan to use CiteSeer API [13] for conducting further experiments.

7. ACKNOWLEDGEMENTS

We would like to thank the participants of the poster session at 2K* 2005 symposium for their valuable comments. This work is supported by grants RP COFIN "Artificial Intelligence Techniques for the Retrieval of High Quality Information on the Web" and RP Quiew "Quality-based Indexing of Web Information".

8. REFERENCES

[1] A. Birukov, E. Blanzieri, and P. Giorgini. Implicit: An agent-based recommendation system for web search. In *Proceedings of the 4th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 618–624. ACM Press, 2005.

[2] E. Blanzieri and P. Giorgini. From collaborative filtering to implicit culture: a general agent-based framework. In *Proceedings of the Workshop on Agents and Recommender Systems*, Barcellona, 2000.

[3] E. Blanzieri, P. Giorgini, P. Massa, and S. Recla. Implicit culture for multi-agent interaction support. In *CooplS '01: Proceedings of the 9th International Conference on Cooperative Information Systems*, pages 27–39, London, UK, 2001. Springer-Verlag.

[4] K. D. Bollacker, S. Lawrence, and C. L. Giles. Discovering relevant scientific literature on the web. *IEEE Intelligent Systems*, 15(2):42–47, 2000.

[5] S. Bradshaw, A. Scheinkman, and K. Hammond. Guiding people to information: providing an interface to a digital library using reference as a basis for indexing. In *IUI '00: Proceedings of the 5th international conference on Intelligent user interfaces*, pages 37–43, New York, NY, USA, 2000. ACM Press.

[6] L. Chen and K. Sycara. Webmate: a personal agent for browsing and searching. In *AGENTS '98: Proceedings of the second international conference on Autonomous agents*, pages 132–139, New York, NY, USA, 1998. ACM Press.

[7] Google scholar: http://scholar.google.com/.

[8] W. C. Janssen and K. Popat. Uplib: a universal personal digital library system. In *DocEng '03: Proceedings of the 2003 ACM symposium on Document engineering*, pages 234–242, New York, NY, USA, 2003. ACM Press.

[9] S. Lawrence, C. L. Giles, and K. Bollacker. Digital libraries and autonomous citation indexing. *Computer*, 32(6):67–71, 1999.

[10] H. Lieberman. Letizia: An agent that assists web browsing. In C. S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 924–929, Montreal, Quebec, Canada, 1995. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[11] S. M. McNee, I. Albert, D. Cosley, P. Gopalkrishnan, S. K. Lam, A. M. Rashid, J. A. Konstan, and J. Riedl. On the recommending of citations for research papers. In *CSCW '02: Proceedings of the 2002 ACM*

conference on Computer supported cooperative work, pages 116–125, New York, NY, USA, 2002. ACM Press.

[12] S. E. Middleton, N. R. Shadbolt, and D. C. D. Roure. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88, 2004.

[13] Y. Petinot, C. L. Giles, V. Bhatnagar, P. B. Teregowda, H. Han, and I. Councill. Citeseer-api: towards seamless resource location and interlinking for digital libraries. In *CIKM '04: Proceedings of the thirteenth ACM conference on Information and knowledge management*, pages 553–561, New York, NY, USA, 2004. ACM Press.

[14] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, Chapel Hill, North Carolina, United States, 1994. ACM Press.

[15] R. Torres, S. M. McNee, M. Abel, J. A. Konstan, and J. Riedl. Enhancing digital libraries with techlens+. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 228–236, New York, NY, USA, 2004. ACM Press.

[16] J. White. Acm opens portal to computing literature. *Communications of the ACM*, 44(7):14–16, 2001.