Ontological User Profiling in Recommender Systems

STUART E. MIDDLETON, NIGEL R. SHADBOLT, and DAVID C. DE ROURE
University of Southampton

We explore a novel ontological approach to user profiling within recommender systems, working on the problem of recommending on-line academic research papers. Our two experimental systems, Quickstep and Foxtrot, create user profiles from unobtrusively monitored behaviour and relevance feedback, representing the profiles in terms of a research paper topic ontology. A novel profile visualization approach is taken to acquire profile feedback. Research papers are classified using ontological classes and collaborative recommendation algorithms used to recommend papers seen by similar people on their current topics of interest. Two small-scale experiments, with 24 subjects over 3 months, and a large-scale experiment, with 260 subjects over an academic year, are conducted to evaluate different aspects of our approach. Ontological inference is shown to improve user profiling, external ontological knowledge used to successfully bootstrap a recommender system and profile visualization employed to improve profiling accuracy. The overall performance of our ontological recommender systems are also presented and favourably compared to other systems in the literature.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*information filtering, relevance feedback*; I.2.6 [**Artificial Intelligence**]: Learning—*knowledge acquisition*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*intelligent agents*

General Terms: Algorithms, Measurement, Design, Experimentation

Additional Key Words and Phrases: Agent, machine learning, ontology, personalization, recommender systems, user profiling, user modelling

1. INTRODUCTION

The mass of content available on the World-Wide Web raises important questions over its effective use. The web is largely unstructured, with pages authored by many people on a diverse range of topics, making simple browsing too time consuming to be practical. Web page filtering has thus become necessary for most web users.

This research was supported by EPSRC Studentship award number 99308831 and the Interdisciplinary Research Collaboration in Advanced Knowledge Technologies (AKT) Project GR/N15746/01.
Authors' address: Intelligence, Agents, Multimedia Group, Department of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK; email: {sem99r,nrs,dder}@ecs.soton.ac.uk.

Search engines are effective at filtering pages that match explicit queries. Unfortunately, people find articulating what they want explicitly difficult, especially if forced to use a limited vocabulary such as keywords. As such, search queries are often poorly formulated, and result in large lists of search results that contain only a handful of useful pages.

The semantic web offers the potential for help, allowing more intelligent search queries based on web pages marked up with semantic metadata. Semantic web technology is, however, very dependent on the degree to which authors annotate their web pages, and automatic web page annotation is still in its infancy. Annotation requires selflessness in authors because the annotations provided will only help other people searching their web pages. Because of this, the vast majority of web pages are not annotated, and in the foreseeable future will remain so. The semantic web can thus only be of limited benefit to the problem of effective searching.

Recommender systems go some way to addressing these issues. We present a novel ontological approach to user profiling within recommender systems. Two recommender systems are build, called Quickstep and Foxtrot, and three experiments conducted to evaluate different aspects of their performance. Quickstep uses ontological inference to improve profiling accuracy and integrates an external ontology for profile bootstrapping. Foxtrot enhances the Quickstep system by employing the novel idea of visualizing user profiles to acquire direct profile feedback.

This section discusses our chosen problem domain and our general approach to ontological recommendation, along with related work. In Section 2, we describe the Quickstep recommender system and an experiment to show how inference can improve user profiling and hence recommendation accuracy. Section 3 details an integration between the Quickstep recommender system and an external ontology, along with an experiment to demonstrate its effectiveness at bootstrapping profiles. In Section 4 the Foxtrot recommender system is described, with an experiment to demonstrate how profile visualization can be used to acquire feedback and hence improve profile accuracy. Lastly, in Section 5, we bring this work together, collating the evidence found to support ontological to user profiling within recommender systems, and discuss future work.

1.1 Recommender Systems

People find articulating what they want hard, but they are very good at recognizing it when they see it. This insight has led to the utilization of relevance feedback, where people rate web pages as “interesting” or “not interesting” and the system tries to find pages that match the “interesting”, “positive examples” and do not match the ‘not interesting’, negative examples. With sufficient positive and negative examples, modern machine learning techniques can classify new pages with impressive accuracy; in some cases, text classification accuracy exceeding human capability has been demonstrated [Larkey 1998].

Obtaining sufficient examples is difficult however, especially when trying to obtain negative examples. The problem with asking people for examples is

that the cost, in terms of time and effort, of providing the examples generally outweighs the reward people will eventually receive. Negative examples are particularly unrewarding, since there could be many irrelevant items to any typical query.

Unobtrusive monitoring provides positive examples of what the user is looking for, without interfering with the users normal work activity. Heuristics can also be applied to infer negative examples from observed behaviour, although generally with less confidence. This idea has led to content-based recommender systems, which unobtrusively watch user behaviour and recommend new items that correlate with a user's profile.

Another way to recommend items is based on the ratings provided by other people who have liked the item before. Collaborative recommender systems do this by asking people to rate items explicitly and then recommend new items that similar users have rated highly. An issue with collaborative filtering is that there is no direct reward for providing examples since they only help other people. This leads to initial difficulties in obtaining a sufficient number of ratings for the system to be useful, a problem known as the cold-start problem [Maltz and Ehrlich 1995].

Hybrid systems, attempting to combine the advantages of content-based and collaborative recommender systems, have proved popular to-date. The feedback required for content-based recommendation is shared, allowing collaborative recommendation as well.

1.2 User Profiling

User profiling is typically either knowledge-based or behavior-based. Knowledge-based approaches engineer static models of users and dynamically match users to the closest model. Questionnaires and interviews are often employed to obtain this user knowledge. Behavior-based approaches use the user's behavior as a model, commonly using machine-learning techniques to discover useful patterns in the behavior. Behavioral logging is employed to obtain the data necessary from which to extract patterns. Kobsa [1993] provides a good survey of user modelling techniques.

The user profiling approach used by most recommender systems is behavior-based, commonly using a binary class model to represent what users find interesting and uninteresting. Machine-learning techniques are then used to find potential items of interest in respect to the binary model. There are a lot of effective machine learning algorithms based on two classes. A binary profile does not, however, lend itself to sharing examples of interest or integrating any domain knowledge that might be available. Sebastiani [2002] provides a good survey of current machine learning techniques.

1.3 Ontologies

An ontology is a conceptualisation of a domain into a human-understandable, but machine-readable format consisting of entities, attributes, relationships, and axioms [Guarino and Giaretta 1995]. Ontologies can provide a rich conceptualisation of the working domain of an organisation, representing the main

concepts and relationships of the work activities. These relationships could represent isolated information such as an employee's home phone number, or they could represent an activity such as authoring a document, or attending a conference.

We use the term, *ontology*, to refer to the classification structure and instances within a knowledge base.

1.4 Problem Domain

The web is increasingly becoming the primary source of research papers to the modern researcher. With millions of research papers available over the web from thousands of websites, finding the right papers and being informed of newly available papers is a problematic task. Browsing this many websites is too time consuming and search queries are only fully effective if an explicit search query can be formulated for what you need. All too often papers are missed.

We address the problem of recommending on-line research papers to the academic staff and students at the University of Southampton. Academics need to search for explicit research papers and be kept up-to-date on their own research areas when new papers are published. We examine an ontological recommender system approach to support these two activities. Unobtrusive monitoring methods are preferred because researchers have their normal work to perform and would not welcome interruptions from a new system. Very high accuracy on recommendations is not required since users will have the option to simply ignore poor recommendations.

Real world knowledge acquisition systems are both tricky and complex to evaluate [Shadbolt et al. 1999]. A lot of evaluations are performed with user log data, simulating real user activity, or with standard benchmark collections, such as newspaper articles over a period of one year, that provide a basis for comparison with other systems. Although these evaluations are useful, especially for technique comparison, it is important to back them up with real world studies so we can see how the benchmark tests generalize to the real world setting. Similar problems are seen in the agent domain where, as Nwana [1996] argues, it has yet to be conclusively demonstrated that people really benefit from agent-based information systems.

This is why a real problem has been chosen upon which to evaluate our work.

1.5 Related Work

Group Lens [Konstan et al. 1997] is an example of a collaborative filter, recommending newsgroup articles based on a Pearson-r correlation of other users' ratings. Fab [Balabanović and Shoham 1997] is a content-based recommender, recommending web pages based on a nearest-neighbor algorithm working with each individual user's set of positive examples. The Quickstep and Foxtrot systems are hybrid recommender systems, combining both these types of approach.

Personal web-based agents such as NewsDude and Daily Learner [Billsus and Pazzani 2000], Personal WebWatcher [Mladenić 1996] and NewsWeeder [Lang 1995] build profiles from observed user behavior. These systems filter

news stories/web pages and recommend unseen ones based on content, using *k*-Nearest Neighbor, naïve Bayes and TF-IDF machine learning techniques. Individual sets of positive and negative examples are maintained for each user's profile. In contrast, by using an ontology to represent user profiles, we pool these limited training examples, sharing between users examples of each class.

Ontologies are used to improve content-based search, as seen in OntoSeek [Guarino et al. 1999]. Users of OntoSeek navigate the ontology in order to formulate queries. Ontologies are also used to automatically construct knowledge bases from web pages, such as in Web-KB [Craven et al. 1998]. Web-KB takes manually labelled examples of domain concepts and applies machine-learning techniques to classify new web pages. Both systems do not, however, capture dynamic information such as user interests.

Digital libraries classify and store research papers, such as CiteSeer [Bollacker et al. 1998]. Typically such libraries are manually created and manually categorized. While our systems are digital libraries, the content is dynamically and autonomously updated from the browsing behavior of its users.

Mladenić and Stefan [1999] provides a good survey of text-learning and agent systems, including content-based and collaborative approaches. The systems most related to Quickstep and Foxtrot are Entrée [Burke 2000], which uses a knowledge base and case-based reasoning to recommend restaurant data, and RAAP [Delgado et al. 1998] that uses simple categories to represent user profiles with unshared individual training sets for each user. None of these systems use an ontology to explicitly represent user profiles.

Of note is that very few systems in the recommender system literature perform user trials using real users. To test classifier accuracy, most use either labelled benchmark document collections, such as Reuters news feed collection, or logged user data, such as Usenet logs.

1.6 Overview of Approach

Our ontological approach to recommender systems uses a hybrid recommender system, employing both collaborative and content-based recommendation techniques and representing user profiles in ontological terms. Two experimental systems have been built that follow this approach, called Quickstep and Foxtrot. Quickstep is a recommender system for a set of researchers within a computer science laboratory, while Foxtrot is a searchable database and recommender system for a computer science department. Figure 1 shows the generic structure of our ontological recommender systems.

A web proxy is used to unobtrusively monitor each user's web browsing, adding new research papers to the central database as users discover them. The research paper database thus acts as a pool of shared knowledge, available to all users via search and recommendation. The database of research papers is classified using a research paper topic ontology and a set of training examples.

Recorded web browsing and relevance feedback elicited from users is used to compute daily profiles of user's research interests. Interest profiles are represented in ontological terms, allowing other interests to be inferred that go beyond that just seen from directly observed behavior. The interest profiles are

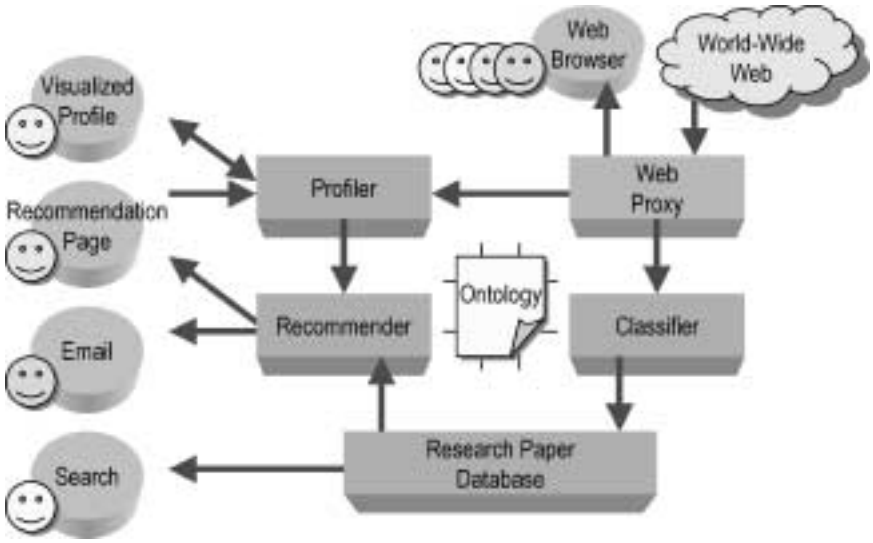


Fig. 1. Our ontological approach to recommender systems.

visualized to allow elicitation of direct profile feedback, providing an additional source of information from which profiles can be computed.

Recommendations are compiled daily using collaborative filtering techniques to find sets of interesting papers. These papers are then constrained to match the top topics of interest within the content-based profiles. The papers left are used to create the recommendations.

Users can view their recommendations via a web page or weekly email message, look at and comment on visualizations of their profile via a web page or just search the research paper database for specific papers of interest. Quickstep, the earlier system, supports only web page recommendation while Foxtrot supports all the interface features.

1.7 Empirical Evaluation

This paper describes three experiments performed using our two recommender systems. The first uses the Quickstep system to measure the effectiveness of using ontological inference in user profiling. Two 1.5 month trials were run using 24 members from the IAM research laboratory, comparing use of ontological profiles and inference to that of using unstructured profiles.

The second experiment integrated the Quickstep system with an external personnel and publication ontology. This experiment measured how effectively an external ontology can bootstrap a recommender system to reduce the recommender system cold-start problem. Behavior logs from the previous experiment were used as the basis for this evaluation.

The third experiment took the Foxtrot recommender system and measures its overall effectiveness and the performance increase obtained when profiles are visualized and profile feedback acquired. A trial was run using 260 staff and students from the computer science department of the University of Southampton

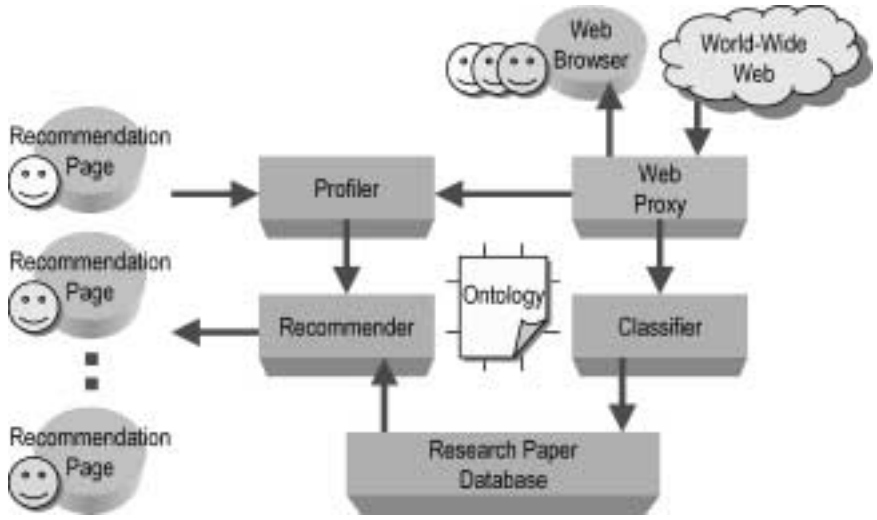


Fig. 2. The Quickstep system.

for an academic year, comparing performance of those subjects who provided profile feedback to those who did not.

2. ONTOLOGICAL USER PROFILING AND PROFILE INFERENCE

Our ontological approach to recommender systems, shown in Figure 2, involves various sub-processes. Our first experimental recommender system, called Quickstep [Middleton et al. 2001], implements all these processes but with just a web page interface. Quickstep is thus just a recommender system, without any search, email or visualization facilities. It was built to help researchers in a computer science laboratory setting, representing user profiling with a research topic ontology and using ontological inference to assist the profiling process. An experiment was run to compare the recommendation performance for subjects whose profiler used ontological inference with those whose profiler did not.

2.1 Overview of the Quickstep Recommender System

Quickstep unobtrusively monitors user browsing behavior via a web proxy, logging each URL browsed during normal work activity. A machine-learning algorithm classifies browsed URLs overnight, using classes within a research paper topic ontology, and saves each classified paper in a central paper store. Explicit relevance feedback and browsed topics form the basis of the interest profile for each user. Is-a relationships within the research paper topic ontology are also exploited to infer general interests when specific topics are observed.

Each day a set of recommendations is computed, based on correlations between user interest profiles and classified paper topics. These recommendations are accessible to users via a web page. Any feedback offered on these recommendations is recorded when the user looks at them. Users can provide new

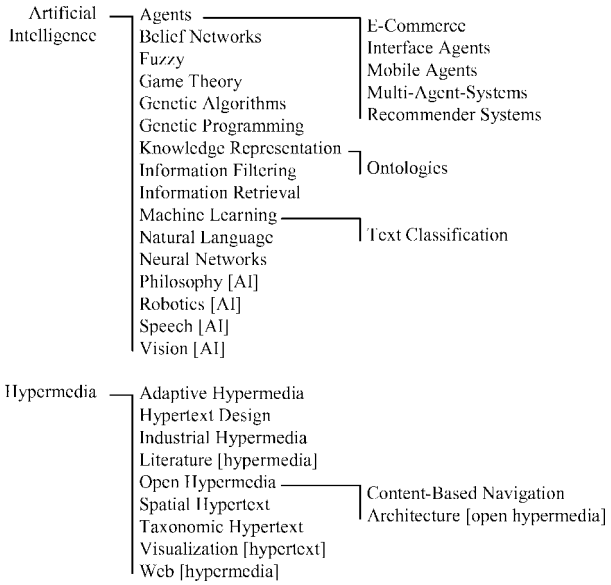


Fig. 3. Section from the Quickstep research paper topic ontology.

examples of topics and correct paper classifications where wrong. In this way, the training set improves over time as well as the profiles.

2.2 Approach of the Quickstep Recommender System

The Quickstep system uses a java-based web proxy, which records time-stamped URLs for each user. This proxy could handle about 30 users. The system ran on a Solaris platform and was mostly written in Java.

2.2.1 *Ontology.* The research paper topic ontology is based on the computer science classifications made by the dmoz open directory project [dmoz] and some minor customisations. We chose to re-use an existing taxonomy to speed development time and provide a potential route for system integration with other external ontologies in the future. Our simple ontology holds is-a relationships between research paper topics, and has 27 classes; for the second trial this ontology was extended to 32 classes. Figure 3 shows a section from the ontology. Pre-trial interviews formed the basis of which additional topics would be added to the ontology to customize it for the target researchers. An expert review by two domain experts validated the ontology for correctness before use in our experiment.

2.2.2 *Research Paper Representation.* Research papers are represented using term vectors. We use "term" to mean a single word within the text of a paper, thus all words that appear in the training set of example papers add one dimension to our term vectors. Term vector weights are computed from the term frequency (TF) divided by total number of terms, representing the normalized frequency in which a word appears within a research paper. Since many words

$$w(d_a,d_b) = \sqrt{\sum_{j=1..T} (t_{ja} - t_{jb})^2}$$

$w(d_a,d_b)$	kNN distance between document a and b
d_a,d_b	document vectors
T	number of terms in document set
t_{ja}	weight of term j document a

Fig. 4. *k*-Nearest Neighbor algorithm.

are either too common or too rare to have useful discriminating power to a classifier, we use a few dimensionality reduction techniques to reduce the number of dimensions of the term vectors. Porter stemming [Porter 1980] is used to remove term suffixes and the SMART [SMART Staff 1974] stop list is used to remove very common words like "the" and "or". Term frequencies below 2 are removed since they have little discriminating power. Dimensionality reduction is common in information system; Sebastiani [2002] provides a good discussion of the issues.

Most on-line research papers are in HTML, PS or PDF formats, with many papers being compressed. We support all these formats for maximum coverage in our problem domain, converting the papers to plain text and using this text to create the term vectors. Unusual or corrupt formats are ignored. Several heuristics are used to determine if the research papers are converted to text correctly and look like a typical research paper with terms such as "abstract" and "references". In the later experiments, term vectors for papers had around 15,000 dimensions after dimensionality reduction.

2.2.3 *Classifier.* Research papers in the central database are classified by an IBk [Aha et al. 1991] classifier, which is boosted by the AdaBoostM1 [Freund and Schapire 1996] algorithm. The IBk classifier is a *k*-Nearest Neighbor type classifier that uses example documents, called a training set, added to a term-vector space. Example documents in the training set are manually labelled using the class names within the research paper topic ontology. Figure 4 shows the basic *k*-Nearest Neighbor algorithm. The closeness of an unclassified vector to its neighbour vectors within the term-vector space determines its classification.

Classifiers like *k*-Nearest Neighbor allow more training examples to be added to their term-vector space without the need to rebuild the entire classifier. They also degrade well, so even when incorrect the class returned is normally in the right "neighborhood" and so at least partially relevant. This makes *k*-Nearest Neighbor a robust choice of algorithm for research paper classification.

Boosting works by repeatedly running a weak learning algorithm on various distributions of the training set, and then combining the specialist classifiers produced by the weak learner into a single composite classifier. The "weak" learning algorithm here is the IBk classifier. Figure 5 shows the AdaBoostM1 algorithm.

AdaBoostM1 has been shown to improve the performance of weak learner algorithms [Freund and Schapire 1996], particularly for the stronger learning

Initialise all values of D to 1/N
Do for t=1..T
call weak-learn(D_t)
calculate error e_t
calculate $\beta_t = e_t/(1-e_t)$
calculate D_{t+1}

$$\text{classifier} = \underset{c \in C}{\text{argmax}} \sum_{\substack{t - \text{all iterations} \\ \text{with result class } c}} \log \frac{1}{\beta_t}$$

D_t	class weight distribution on iteration t
N	number of classes
T	number of iterations
weak-learn(D_t)	weak learner with distribution D_t
e_t	weak_learn error on iteration t
β_t	error adjustment value on iteration t
classifier	final boosted classifier
C	all classes

Fig. 5. AdaBoostM1 boosting algorithm.

Fig. 6. Quickstep's web-based interface.

algorithms like *k*-Nearest Neighbor. It is thus a sensible choice to boost our IBk classifier.

Other types of classifier were considered, including the naïve Bayes classifier and the C4.5 decision tree, and informal tests run to evaluate their performance. The boosted IBk classifier was found to give superior performance for this domain.

2.2.4 *Web Page Interface.* Recommendations are presented to the user via a browser web page, shown in Figure 6. The web page applet loads the current recommendation set and records any feedback the user provides. Research papers can be jumped to, opening a new browser window to display

Fig. 7. Topic popup menus.

the paper URL. If the user likes or dislikes a paper topic, the interest feedback combo-box allows “interested” or “not interested” to replace the default “no comment”.

Clicking on the topic and selecting a new one from a popup menu can change the topic of each paper, should the user feel the classification is incorrect. Later in the experiment, the ontology group has a hierarchical popup menu, and the flat list group has a single level popup menu. Figure 7 shows the hierarchical popup menu.

New examples can be added via the interface, with users providing a paper URL and a topic label. These are added to the groups training set, allowing users to teach the system new topics or improve classification of old ones.

All feedback is stored in log files, ready for the profile builders run. The feedback logs are also used as the primary metric for evaluation. Interest feedback, topic corrections and jumps to recommended papers are all recorded.

2.2.5 *Profiler.* Interest profiles are computed daily by correlating previously browsed research papers with their classification. User profiles thus hold a set of topics and interest values in these topics for each day of the trial. User feedback also adjusts the interest of topics within the profile and a time decay function weights recently seen papers as being more important than older ones. Ontological relationships between topics of interest are used to infer other topics of interest, which might not have been browsed explicitly; an instance of an interest value for a specific class adds 50% of its value to the super-class. Figure 8 shows the profiling algorithm.

Profile feedback details a level of interest in a topic over a period of time. The user defines the exact level and duration of interests when they draw interest bars onto the time/interest graph via the profile interface. The profiling

$$\text{Topic interest} = \sum_{1..\text{no of instances}}^{n} \text{Interest value}(n) / \text{days old}(n)$$

Event interest values	Paper browsed = 1
	Recommendation followed = 2
	Topic rated interesting = 10
	Topic rated not interesting = -10
Interest value for super-class per instance	= 50% of sub-class

Fig. 8. Profiling algorithm.

$$\text{Recommendation confidence} = \text{classification confidence} * \text{topic interest value}$$

Fig. 9. Quickstep recommendation algorithm.

algorithm automatically adjusts the daily profiles to match any topic interest levels declared via profile feedback.

Event interest values were chosen to favour explicit feedback over implicit, and the 50% value used to represent the reduction in confidence you get the further from the direct observation you are.

Other profiling algorithms exist such as time-slicing and curve fitting, but the time-decay function appeared in informal tests to produce a good result; we found it to be a robust function for finding current interests.

2.2.6 *Recommender.* Recommendations are formulated from a correlation between the users' current topics of interest and papers classified as belonging to those topics, described in Figure 9. A paper is only recommended if it does not appear in the users browsed URL log, ensuring that recommendations have not been seen before. For each user, the top three interesting topics are selected with 10 recommendations made in total. Papers are ranked in order of the recommendation confidence before being presented to the user.

The classification confidence is computed from the AdaBoostM1 algorithm's class probability value for a paper, a value between 0 and 1.

2.3 Evaluation of Ontological Inference in User Profiling

We used the Quickstep recommender system to compare subjects whose profiles were computed using ontological inference with subjects whose profiles did not use ontological inference. The experiment took place over a 3-month period in the IAM laboratory using 24 computer science researchers. An overall evaluation of the Quickstep recommender system was also performed. The Quickstep recommender system and this experiment are published in more detail in Middleton et al. [2001].

2.3.1 *Experimental Design.* Two identical trials were conducted, the first with 14 subjects and the second with 24 subjects, both over 1.5 months. Some interface improvements were made for the second trial and 5 more ontological classes were added.

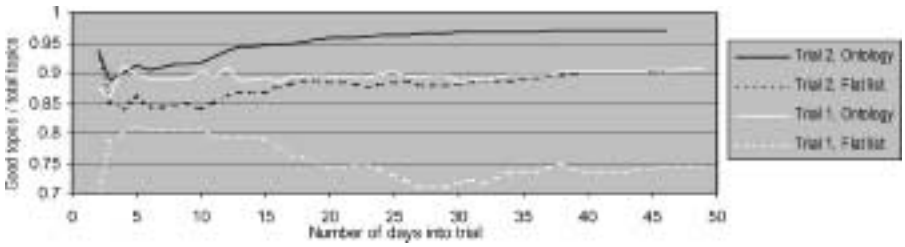


Fig. 10. Ratio of good topics/total topics.

Subjects were divided into two groups, one using an ontological approach to user profiling with a topic ontology and the other using a flat, unstructured list of topics. Both groups had their own separate training set of examples, which diverged from the bootstrap training set as the trial progressed when users corrected the classification of papers and hence provided new examples. The classifier algorithm was identical for both groups; only the training set changed.

The system interface used by both groups was identical, except for the popup menu for choosing paper topics. The ontology group had a hierarchical menu that used the topic ontology; the flat list group had a single level menu.

The system recorded each time the user declared an interest in a topic by selecting it "interesting" or "not interesting", jumped to a recommended paper or corrected the topic of a recommended paper. These feedback events were date stamped and recorded in a log file for later analysis, along with a log of all recommendations made.

2.3.2 *Experimental Results.* Topic interest feedback is where the user comments on a recommended topic, declaring it "interesting" or "not interesting", and is an indication of the accuracy of the current profile. When a recommended topic is correct for a period of time, a user will tend to become content with it and stop rating it as "interesting". On the other hand, an uninteresting topic is likely to always attract a "not interesting" rating. Good topics are defined as either "no comment" or "interesting" topics. The cumulative frequency figures for good topics are presented in Figure 10 as a ratio of the total number of topics recommended.

The two ontological groups have a 7% and 15% higher topic acceptance. In addition to this trend, the first trial ratios are about 10% lower than the second trial ratios.

A jump is where the user jumps to a recommended paper by opening it via the web browser. Jumps are correlated with topic interest feedback, so a good jump is a jump to a paper on a good topic. Recommendation accuracy is the ratio of good jumps to recommendations, and is an indication of the quality of the recommendations being made as well as the accuracy of the profile. Figure 11 shows the recommendation accuracy results.

There is a small 1% improvement in recommendation accuracy by the ontology group. Both trials show between 8–10% of recommendations leading to good jumps.

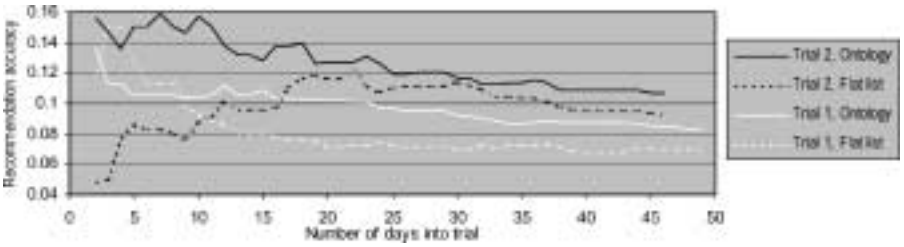


Fig. 11. Recommendation accuracy.

Table I. Quickstep Classifier Accuracy and Coverage

	Accuracy	Coverage	Classes	Examples	Terms
Trial 1 Ontology	0.48	0.90	27	157	15897
Trial 1 Flat list	0.52	1.00	25	162	16311
Trial 2 Ontology	0.46	0.89	32	208	17601
Trial 2 Flat list	0.46	0.97	32	212	16798

A cross-validation test was run on each group's final training set to assess the accuracy and coverage of the classifier. The results are shown in Table I. The accuracy value is a ratio of how many correctly classified papers there were over the number classified. The coverage value is a ratio of how many papers were classified over the total number of papers.

2.3.3 *Discussion.* From the experimental data of both trials, several suggestive trends are apparent. The initial ratios of good topics were lower than the final ratios, reflecting the time it takes for enough log information to be accumulated to let the profiles settle down. The ontology users were 7–15% happier overall with the topics suggested to them.

A hypothesis for the ontology group's apparently superior performance is that the is-a hierarchy produces a rounder, more complete profile by including general super class topics when a specific topic is browsed by a user. This in turn helps the profile to acquire a broad range of interests, rather than just latching onto one correct topic.

The first trial showed fewer good topics than the second trial with a 10% difference seen by both groups. This is probably because of interface improvements made for the second trial, where the topic feedback interface was made less confusing. Subjects were sometimes rating interesting topics as not interesting if the paper quality was poor. Since there are more poor quality papers than good quality ones, this introduced a bias to not interesting topic feedback resulting in a lower overall ratio.

About 10% of recommendations led to good jumps. Since 10 recommendations were given to the users at a time, on average one good jump was made from each set of recommendations received. As with the topic feedback, the ontology group again was marginally superior but only by a 1% margin; this trend is promising but not statistically significant even though it appears in both trials. This smaller difference is probably due to people having time to follow only one or two recommendations. Thus, although the ontology group had more good

topics, only the top topic of the three recommended was really be looked at; the result was a smaller difference between the good jumps made and the good topics seen.

These results are not statistically significant due to the sample size. Nevertheless, the trend in the data appears to be encouraging.

2.4 Conclusions

The results suggest how using ontological inference in the profiling process results in superior performance over using a flat list of unstructured topics. The ontology users tended to have more "rounder" profiles, including topics of interest that were not directly browsed. This increased the accuracy of the profiles, and hence usefulness of the recommendations.

Very few systems in the recommender system literature perform user trials using real users, making direct comparison difficult. Most use either labelled benchmark document collections to test classifier accuracy or logged user data taken from sources such as newsgroups. NewsWeeder reports a 40–60% classification accuracy with real users, while Personal WebWatcher [Mladenić 1996] reports a 60–90% classification accuracy using benchmark data. Quickstep's classifier is on the low side with 40–50% accuracy, but this appears much better when the number of classes used in classification is taken into account and the potential this allows for improving profiling via inference and profile feedback.

The Daily Learner [Billsus and Pazzani 2000] system recommends news stories via a wireless hand held receiver. They used a k-NN algorithm for short-term profiles and a naïve Bayes algorithm for longer-term profiles. Over a 10-day trial, involving 300 users, they reported a precision of 33%, recall 29%, for the top four recommendations based on recording which stories were selected by users to read. This provided a small increase in performance compared to a edited news service, Yahoo! news.

The Entrée [Burke 2000] restaurant recommender system, which uses a knowledge-based approach to recommendation, reports a recommendation accuracy of 38%, for 15 item profiles, to 70%, for 5 item profiles, based on analysis of historically logged user activity of a web site.

An informal result was seen in the nearest neighbour classifier's robustness. Even when it made a mistake, 50–60% of the time in fact, the class it chose was normally in the correct area. For example, for an "interface agent" paper, the classification would more likely be "agent" than "human computer interaction". The users liked this as it showed the system was at least making a reasonable attempt at classification, even if it was getting things wrong.

3. BOOTSTRAPPING USING AN EXTERNAL ONTOLOGY

Having a recommender system that represents user profiles in ontological terms offers the potential for communication and knowledge sharing with other, external ontologies. This section examines our integration of the Quickstep recommender system with an external ontology built from a publication database and personnel database. The integration is made possible because the external

ontology uses the same research paper topic ontology as the Quickstep system. The experiment we ran bootstrapped the Quickstep recommender system with knowledge about the researchers who wanted to use it, thus reducing the cold-start problem, a problem inherent to all recommender systems. Details of this work have been published in Middleton et al. [2002].

3.1 Integrating an External Ontology with the Quickstep Recommender System

One difficult problem commonly faced by recommender systems is the cold-start problem [Maltz and Ehrlich 1995], where recommendations are required for new items or users for whom little or no information has yet been acquired. Poor performance resulting from a cold-start can deter user uptake of a recommender system. This effect is thus self-destructive, as the recommender never achieves good performance since users never use it for long enough. We examine two types of cold-start problem.

The *new-system* cold-start problem is where there are no initial ratings by users, and hence no profiles of users. In this situation, most recommender systems have no basis on which to recommend, and hence perform very poorly.

The *new-user* cold-start problem is where the system has been running for a while and a set of user profiles and ratings exist, but no information is available about a new user. Most recommender systems perform poorly in this situation too.

Collaborative recommender systems fail to help in cold-start situations, since they cannot discover similar user behavior because there is not enough previously logged behavior data upon which to base any correlations. Content-based and hybrid recommender systems perform a little better since they need just a few examples of user interest in order to find similar items.

External ontological sources of knowledge complement well the behavioral information held within the recommender systems, by providing initial knowledge about users and their domains of interest. Of particular interest to our academic problem domain is knowledge about departmental publications, the projects a researcher has worked on and the position a researcher has within the department. It should thus be possible to bootstrap the initial learning phase of a recommender system with such knowledge, easing the cold-start problem.

In return for any bootstrap information, the recommender system could provide details of dynamic user interests to the ontology. This would reduce the effort involved in acquiring and maintaining knowledge of people's research interests. To this end, we investigate the integration of Quickstep, a web-based recommender system, and an external ontology built from a publication database and personnel database.

3.2 Approach to Integrating the External Ontology

We integrate the Quickstep recommender system and external ontology using a synergistic arrangement. The external ontology initially bootstraps the recommender system, and after the cold-start is over the user profiles held within the recommender system are sent to the external ontology as an additional knowledge source. Figure 12 shows the integrated system.

Fig. 12. Integration of Quickstep and an external ontology.

The synergy of this relationship is fully explored in Middleton et al. [2002]. A relationship analysis tool, Ontocopi [Alani et al. 2003], is used to uncover each user's communities of practice by applying a set of ontology-based network analysis techniques that examine the connectivity of instances in the knowledge base with respect to the type, density, and weight of these connections. Ontocopi applies an expansion algorithm that generates the community of practice for a selected instance by identifying the set of close instances and ranking them according to the weight of their relationships. It applies a breadth-first spreading activation search, traversing the semantic relationships between instances until a defined threshold is reached. Semantic distance is thus used to indicate similarity between users.

The external ontology used in this work was designed to represent the academic domain, and was developed by Southampton's Advanced Knowledge Technologies (AKT) project team [O'Hara et al. 2001]. It models people, projects, papers, events and research interests. The ontology itself is implemented in Protégé 2000 [Eriksson et al. 1999], a graphical tool for developing knowledge-based systems. It is populated with information extracted automatically from a departmental personnel database and publication database. The ontology consists of around 80 classes, 40 slots, over 13000 instances and is focused on people, projects, and publications.

3.2.1 *Bootstrapping Algorithms.* Upon start-up, the ontology provides the recommender system with an initial set of publications for each of its registered users. Each user's known publications are then correlated with the recommender systems classified paper database, and a set of historical interests compiled for that user. These historical interests form the basis of an initial profile, overcoming the new-system cold-start problem. Figure 13 details the new-system initial profile algorithm. As in the Quickstep profiling algorithm, fractional interest in a topic super-classes is inferred when a specific topic is added.

When the recommender system is up and running and a new user is added, the ontology provides the historical publication list for the new user and the relationship analysis tool provides a ranked list of similar users. The initial profile of the new user is formed from a correlation between historical publications and any similar user profiles. This algorithm is detailed in Figure 14, and addresses the new-user cold-start problem.

$$\text{topic interest}(t) = \sum_{\substack{1..\ \text{publications} \\ \text{belonging to class } t}}^{n} 1 \,/\, \text{publication age}(n)$$

new-system initial profile = (t, topic interest(t))*

t = <research paper topic>

Fig. 13. New-system initial profile algorithm.

$$\text{topic interest}(t) = \frac{\gamma}{N_{\text{similar}}} \sum_{1..\ N_{\text{similar}}}^{u} \text{profile interest}(u,t) + \sum_{1..\ N_{\text{pubs t}}}^{n} 1 \,/\, \text{publication age}(n)$$

profile interest(u,t) = interest of user u in topic t * confidence
new-user initial profile = (t, topic interest(t))*

t = research paper topic
u = user
γ = weighting constant >= 0
N_{similar} = number of similar users
$N_{\text{pubs t}}$ = number of publications belonging to class t
confidence = confidence in user similarity

Fig. 14. New-user initial profile algorithm.

3.3 Experiment to Evaluate Bootstrapping Performance

We used the integration of the Quickstep recommender system with an external ontology to evaluate how using ontological knowledge could reduce the cold-start problem. The external ontology used was the AKT ontology described earlier, based on a publication database and personnel database, coupled with a tool for performing relationship analysis of ontological relationships to discover similar users. The behavioral log data from the previous experiment was used to simulate the bootstrapping effect both the new-system and new-user initial profiling algorithms would have. Both the integration and experiment are published in more detail in Middleton et al. [2002].

3.3.1 *Experimental Design.* Subjects were selected from those who participated in the previous Quickstep experiment and had entries within the external ontology. We selected nine subjects in total, with each subject typically having one or two publications.

The URL browsing logs of these users, extracted from the 3 months of browsing behavior recorded during the Quickstep trials, were broken up into weekly log entries. Seven weeks of browsing behavior were taken from the start of the

$$\text{profile precision} = \frac{1}{N_{users}} \sum_{1..N_{users}}^{user} \frac{N_{correct}}{N_{correct} + N_{missing}}$$

$$\text{profile error rate} = \frac{1}{N_{users}} \sum_{1..N_{users}}^{user} \frac{N_{incorrect}}{N_{correct} + N_{incorrect} + N_{missing}}$$

$N_{correct}$	Number of user topics that appear in current profile and benchmark profile
$N_{missing}$	Number of user topics that appear in benchmark profile but not in current profile
$N_{incorrect}$	Number of user topics that appear in current profile but not in benchmark profile
N_{users}	Total number of users

Fig. 15. Bootstrapping evaluation metrics.

Quickstep trials, and an empty log created to simulate the very start of the trial where no behavior has yet been recorded.

Eight iterations of the integrated system were thus run, the first simulating the start of the trial and others simulating the following weeks 1 to 7. Interest profiles were recorded after each iteration. Two complete runs were made, one with the "new-system initial profiling" algorithm and a control run with no bootstrapping. The control run without the "new-system initial profiling" algorithm started with blank profiles for each of its users. An additional iteration was run to evaluate the effectiveness of the "new-user initial profile" algorithm.

In order to evaluate the algorithms effect on the cold-start problem, all recorded profiles were compared to the benchmark week 7 profile. This allowed measurement of how quickly profiles converge to the stable state existing after a reasonable amount of behavior data has been accumulated. The quicker the profiles move to this state the quicker they will have overcome the cold-start. Week 7 was chosen as the cut-off point of our analysis since after about 7 weeks of use the behavior data gathered by Quickstep dominated the user profiles and the cold-start was over.

3.3.2 *Experimental Results.* Two measurements were made when comparing profiles to the benchmark week 7 profile. The first, profile precision, measures how many topics were mentioned in both the current profile and benchmark profile. Profile precision is an indication of how quickly the profile is converging to the final state, and thus how quickly the effects of the cold-start are overcome. The second, profile error rate, measures how many topics appear in the current profile that do not appear within the benchmark profile. Profile error rate is an indication of the errors introduced by the two bootstrapping algorithms. Figure 15 describes these metrics.

It should be noted that the absolute precision and error rate of the profiles are not measured—only the relative precision and error rate compared to the week 7 steady state profiles. Absolute profile precision is a subjective measurement.

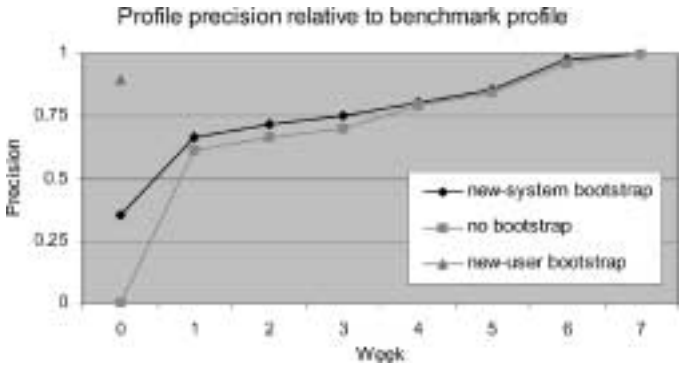


Fig. 16. Bootstrapping profile precision.

The results of our experiment are detailed in Figure 16. The new-user results consist of a single iteration, so appear on the graph as a single point.

At the start, week 0, no browsing behavior log data is available to the system so the profiles without bootstrapping are empty. The new-system algorithm, however, can bootstrap the initial user profiles and achieves a reasonable precision of 0.35 and a low error rate of 0.06. We found that the new-system profiles accurately captured interests users had a year or so ago, but tended to miss current interests. This is because publications are generally not available for up-to-date interests.

As expected, once the weekly behavior logs become available to the system the profiles adjust accordingly, moving away from the initial bootstrapping. On week 7, the profiles fully converge to the benchmark profile, the cold-start being over. This is the reason the precision graph converges at week 7. Bootstrapping is most effective during week one, where no information about users is available and when the users are most likely to stop using the system due to poor performance.

The new-user algorithm result show a more dramatic increase in precision to 0.84, but comes at the price of a significant error rate of 0.55. The profiles produced by the new-user algorithm tended to be very inclusive, taking the set of similar user interests and producing a union of these interests. While this captures many of the new users real interests, it also included a large number of interests not relevant to the new user but which were interesting to the people similar to the new user.

3.3.3 *Discussion.* The new-system algorithm produced profiles with a low error rate and a reasonable precision of 0.35. This reflects the fact that previous publications are a good indication of users current interests, and so can produce a good starting point for a bootstrap profile. Where the new-system algorithm fails is for more recent interests, which make up the remaining 65% of the topics in the final benchmark profile. To discover these more recent interests, it is possible that the new-system algorithm could be extended to take some of the other information available within the ontology into account, such as the projects a user is working on. To what degree these relationships will help

is difficult to predict however, since the ontology itself has great difficulty in acquiring knowledge of recent interests.

For the purposes of this experiment, those users who had some entries within the external ontology were selected. There were some users who had not entered their publications into the ontology or who had yet to publish their work. For these users there is little information within the ontology, making any new-system initial profiles of little use. In a larger scale system, more sources of information would be needed from the ontology to build the new-system profiles. This would allow some redundancy, and hence improve robustness in the realistic situation where information is sparsely available.

The new-user algorithm achieved good precision of 0.84 at the expense of a significant error rate. This was partly because the new-user algorithm included all interests from the similar users. An improvement would be to only use those interests held by the majority of the similar people. This would exclude some of the less common interests that would otherwise be included into the new-user profile.

3.4 Related Work

The cold-start problem is discussed by Claypool et al. [1999] who examines a hybrid approach to recommendation, using a content-based overlap coefficient technique coupled with a collaborative Pearson-r approach that takes over when sufficient ratings have been acquired. The hybrid concept is also adopted by Melville et al. [2002] with a content-boosted approach, where pseudo ratings are generated by a naïve Bayes classifier to supplement real ratings. There is still a need to acquire initial examines of content, however, before recommendation can occur.

Some metrics for measuring recommendation performance are suggested by Schein et al. [2002], along with a discussion of the cold-start problem. They apply a CROC curve based metric on a naïve Bayes classifier.

Several heuristic-based item selection strategies are compared by Rashid et al. [2002] using the MovieLens dataset. Focus is given to reducing user effort during sign-up as well as recommendation accuracy.

3.5 Conclusions

Cold-starts in recommender systems are a significant problem. If initial recommendations are inaccurate, user confidence in the recommender system may drop with the result that users give up on the system and thus not enough usage data is gathered to ever overcome the cold-start.

This experiment suggests that using an ontology to bootstrap user profiles can significantly reduce the impact of the recommender system cold-start problem. It is particularly useful for the new-system cold-start problem, where the alternative is to start with no information at all.

A question still remains as to just how good an initial profile must be to fully overcome the effects of the cold-start problem. If initial recommendations are poor users will not use the recommender system and hence it will never get a chance to improve. We have shown that improvements can be made to initial

profiles, but further empirical evaluation would be needed to evaluate exactly how much improvement is needed before the system is "good enough" for users to give it a chance.

4. THE FOXTROT RECOMMENDER SYSTEM

Our second experimental recommender system, called Foxtrot, extended the Quickstep system by implementing a research paper search interface, profile visualization interface and email notification in addition to the web page recommendation interface. Profile visualization is made possible because profiles are represented in ontological terms understandable to the users. Foxtrot was built to help researchers in a computer science department, allowing researchers to search the database of research papers in addition to receiving recommendations. A large-scale experiment was run to evaluate the overall approach, and to compare the recommendation performance of subjects who provided profile feedback to the performance of those subjects who just provided relevance feedback.

4.1 Overview of the Foxtrot Recommender System

Foxtrot is an evolution of the Quickstep system, and fully follows the architecture shown earlier in Figure 1. Foxtrot differs from Quickstep in the following ways. The number of interfaces supported is increased, providing a research paper search interface, profile visualization and feedback facility and e-mail notification support. A static research paper ontology with many more classes is used, along with increased dimensionality reduction to cope with the increase in classes and hence term dimensions. The profiler takes profile feedback into account allowing users control over their own profiles. Lastly, a more collaborative recommendation algorithm is employed, talking into account the profiles of other similar users when deciding what to recommend.

4.2 Approach of the Foxtrot Recommender System

The Foxtrot system used a Squid proxy, which can handle hundreds of users across the whole computer science department at the University of Southampton. Each user ran an Identd server so the Squid proxy could identify their usernames, and a modified Squid logging function was used to record time-stamped URLs for each user. The system ran on a dedicated LINUX platform and like Quickstep was mostly written in Java.

Since Foxtrot is an evolution of the Quickstep system, only the processes that changed are detailed. All other processes, for example, the classifier, work in the way described for the Quickstep system.

4.2.1 *Research Paper Topic Ontology.* Foxtrot uses a research paper topic ontology to represent the research interests of its users. A class is defined for each research topic and is-a relationships defined where appropriate. The ontology is based on the CORA [McCallum et al. 2000] digital library, since it classifies computers science topics and has example papers for each class. Figure 17 shows some of the classes within the ontology. An existing ontology

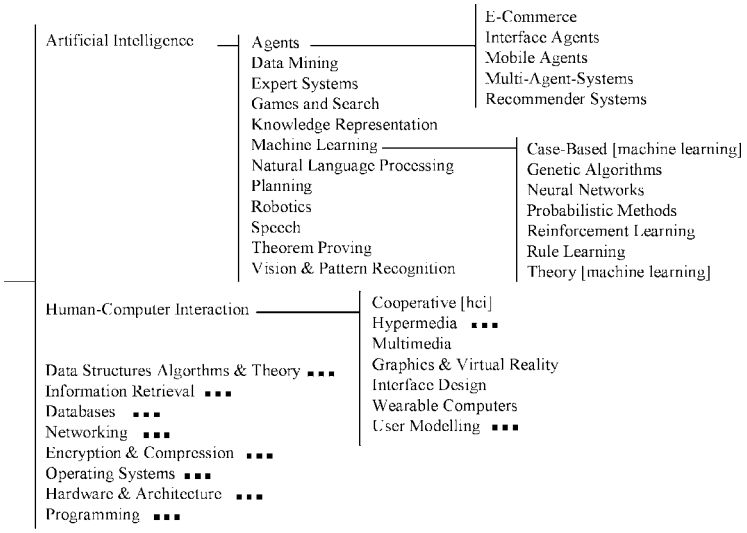


Fig. 17. Section from the Foxtrot research paper topic ontology.

was chosen for reuse to speed development time and provide a source of example papers.

Labelled examples of research papers were manually added to the classifier training set, many taken from the results of the Quickstep trial and papers downloaded from the CORA system. There were a total of 97 classes and 714 training examples. The ontology remained fixed throughout the Foxtrot trial, but could in theory be updated as time goes on to reflect changes in the research domain. We decided not to allow users to add classes to the ontology since with a large number of users there are bound to be classification mistakes, and these could adversely effect the classification accuracy. For every ontological class, a set of 5–10 example papers was provided at the start of the trial.

4.2.2 *Research Paper Representation.* Research papers are represented as term vectors, just as in the Quickstep system. Because of the increased number of classes an additional dimensionality reduction technique is employed to keep the number of terms within the vectors manageable. Each class was constrained to have only its top 50 terms, ranked by document frequency; the union of each classes most discriminating terms was thus used for term vectors. In the Foxtrot experiment, term vectors had 1152 dimensions.

4.2.3 *Interface.* Users primarily interact with Foxtrot via a web page. The basic interface is shown in Figure 18. A web search engine metaphor, familiar to most computer scientists, was used for the interface design, allowing users to enter search queries via edit boxes and a search button used to initiate a search.

Search results are returned in the area below the edit boxes, showing the details of each research paper found. Two sets of radio buttons appear below each search result to allow users to provide relevance and quality feedback if they so desire. When users first go to the Foxtrot web page their daily recommendations

Fig. 18. Foxtrot's recommendation and search interface.

Fig. 19. Foxtrot's profile visualization interface.

are automatically presented in the search result area. In this way, users can choose to read the recommendations or just enter a search query and use the system normally.

Users who are in the profile group can visualize their interest profiles by clicking on a profile tab. Figure 19 shows the profile interface. Profiles are

Fig. 20. Foxtrot's email notification interface.

$$\text{Pearson r coefficient}_{ab} = \frac{\sum_{\text{topics}}^{t} (I_a(t) - \bar{I}_a) * (I_b(t) - \bar{I}_b)}{\sqrt{\sum_{\text{topics}}^{t} (I_a(t) - \bar{I}_a)^2 * \sum_{\text{topics}}^{t} (I_b(t) - \bar{I}_b)^2}}$$

$I_a(t)$ = User a's interest in topic t
$\bar{I}_a$ = User a's mean interest value over all topics

Pearson r coefficient$_{ab}$ = similarity of user a's profile to user b's profile
Recommended papers = papers on user's current interests $\cap$ papers read by similar user's

3 papers recommended on the 3 most interesting topics, totalling 9 papers per day
If more than 3 papers meet above criteria, papers ranked by quality rating

Fig. 21. Foxtrot's recommendation algorithm.

displayed as a time/interest graph, showing what the system thinks their top few interests are over the period of the trial. Direct profile feedback can be drawn onto this graph by using the controls to the side. A drawing package metaphor is used here, and users can draw colored horizontal bars to represent a level of interest in a topic over a period of time. In this way, users can literally draw their own profiles.

In addition to the Foxtrot web page, a weekly e-mail notification feature was added 3 months from the end of the trial. This provided a weekly e-mail stating the top three recommendations from the current set of nine recommendations. Users could then jump to these papers or load the Foxtrot web page and review all nine recommendations. Figure 20 shows the e-mail notification message.

4.2.4 *Recommendation Agent.* Daily recommendations are formulated by a hybrid recommendation approach. A list of similar people to a specific user is compiled, using a Pearson-r correlation on the content-based user profiles. Recommendations for a user are then taken from those papers on the current topics of interest, which have also been read by people similar to that user. Figure 21

shows the recommendation algorithm. During the Foxtrot trial, three papers were recommended each day on the three most interesting topics, making a total of nine recommended papers. Previously read papers were not recommended twice and if more than three papers were available for a topic they were ranked by average quality rating.

4.3 Experiment to Evaluate Profile Visualization and Feedback

Our third experiment used the Foxtrot recommender system to compare subjects who could visualize their profiles and provide profile feedback with subjects who could only use traditional relevance feedback. Profile visualization and feedback is only possible because profiles are represented using an ontology, which contains concepts users can understand. This experiment took place over an academic year with 260 staff and students of the computer science department at the University of Southampton. An overall evaluation of the Foxtrot recommender system was also performed.

4.3.1 *Experimental Design.* The experimental trial took place over the academic year 2002, starting in November and ending in July. Of the 260 subjects registered to use the system, 103 used the web page, and of these 37 subjects used the system three or more times; this makes the uptake rate 14%. All 260 subjects used the web proxy and hence their browsing was recorded and daily profiles built. As such, 260 subjects contributed, by way of the web proxy monitoring their web browsing, to the growth of the research paper database, but there were only 37 active users during the experiment. By the end of the trial the research paper database had grown from 6,000 to 15,792 documents as a result of subject web browsing.

Subjects were divided into two groups. The first "profile feedback" group had full access to the system and its profile visualization and profile feedback options; the second "relevance feedback" group were denied access to the profile interface. It was found that many in the "profile feedback" group did not provide any profile feedback at all, so in the later analysis these subjects are moved into the "relevance feedback" group. A total of nine subjects provided profile feedback.

Towards the end of the trial, an additional e-mail feature was added to the recommender system. This e-mail feature sent out weekly e-mails to all users who had used the system at least once, detailing the top three papers in their current recommendation set. E-mail notification was started in May and ran for the remaining 3 months of the trial.

The feedback data obtained from the trial occurs at irregular time intervals, based on when subjects looked at recommendations or browsed the web. For ease of analysis, data is collated into weekly figures by summing interactions throughout each week. Group data is computed by summing the weekly contribution of each subject within a group. Figure 22 shows the metrics measured.

4.3.2 *Experimental Results.* The recommendation accuracy metric takes explicit feedback and computes accuracy figures for both web page and email

<future papers> = browsed/jumped papers in the 4 weeks after profile
<papers> = browsed/jumped papers over duration of profile (normally 1 day)
<top topics> = top 3 topics of profile

$$\text{Predicted profile accuracy} = \frac{\text{No of <future papers> matching <top topics>}}{\text{No of <future papers>}}$$

$$\text{Profile accuracy} = \frac{\text{No of <papers> matching <top topics>}}{\text{No of <papers>}}$$

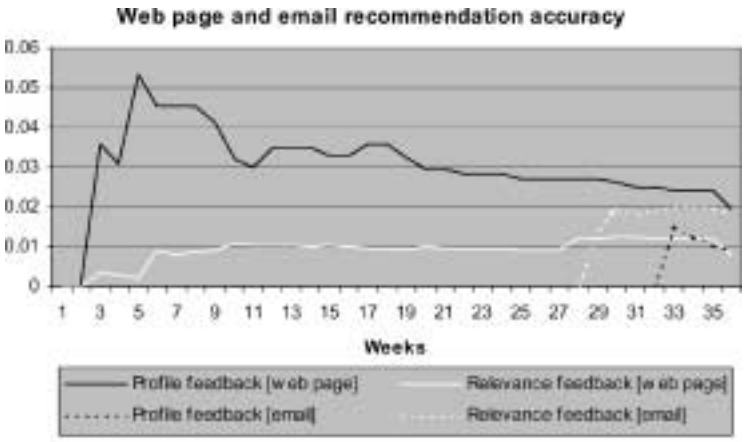
$$\text{Web page rec accuracy} = \frac{\text{No of recommended papers browsed or jumped to}}{\text{No of recommended papers}}$$

$$\text{Email rec accuracy} = \frac{\text{No of emailed papers browsed or jumped to}}{\text{No of emailed papers}}$$

$$\text{Jumps to recommendations} = \frac{\text{No of jumps to recommended papers}}{\text{No of jumps}}$$

$$\text{Jumps to profile topics} = \frac{\text{No of jumps to papers matching <top topics>}}{\text{No of jumps}}$$

Fig. 22. Measured metrics.



Final week 95% confidence intervals			
Profile [web page]	0.043	Relevance [web page]	0.0028
Profile [email]	0.018	Relevance [email]	0.011

Fig. 23. Web page and email recommendation accuracy.

recommendations. A simple ratio is used to obtain the number of recommendations followed as a fraction of the total number of recommendations; this provides a measure of the effectiveness of the recommendations. Figure 23 shows the recommendation accuracy for web page and email recommendations.

The small number of subjects within the 'profile feedback' group accounts for the larger confidence intervals. While not statistically significant, there is an apparent trend for more accurate recommendation when using profile feedback, especially in the early weeks. Email recommendations appeared to be preferred

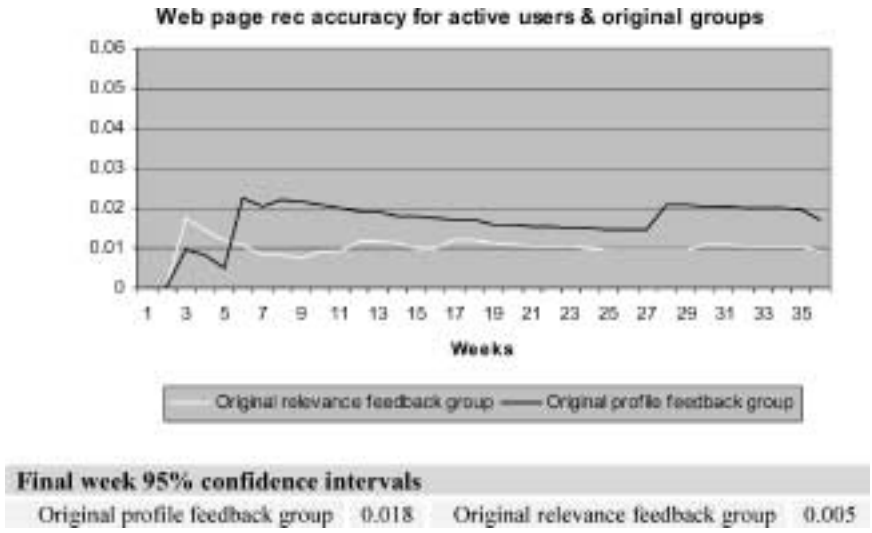


Fig. 24. Web page recommender accuracy for original group's active subjects.

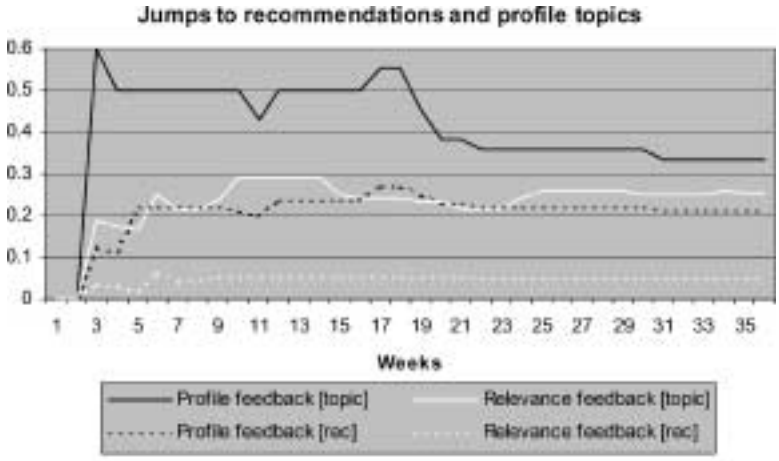
by the "relevance feedback" group, slightly out performing the "profile feedback" group.

The above results compare the profile feedback users with relevance feedback users. To show that the profile feedback user group was not "self selected", containing only the active users, Figure 24 shows the original grouping's accuracy figures for only active users; active users are defined as those who used the system three or more times. There were 16 active users in the relevance feedback group and 21 active users in profile feedback group. The profile feedback group's recommendation accuracy is still higher, though not by as much since the results are averaged in the profile feedback group between those who did provide a profile and those that only used relevance feedback.

In addition to recommendation accuracy, the proportion of jumps to recommendations and jumps to papers with a top 3 topic was computed. Jumps to recommendations measure the degree to which subjects use the recommendation facility as opposed to just using the search facility of the database. Jumps to papers with a top 3 profile topic measures how well the profiles fitted the subject's actual interests. Figure 25 shows these figures.

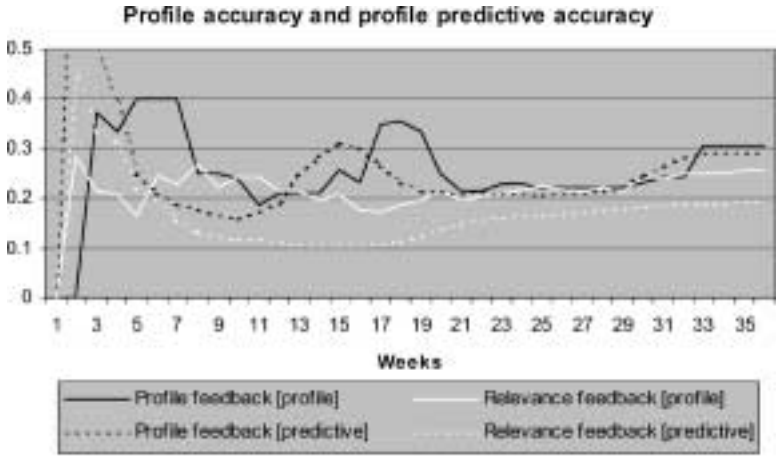
The "profile feedback" group made a greater proportion of jumps to recommendations than the "relevance feedback" group; this trend is statistically significant. A similar trend is seen in jumps to papers on top profile topics, but is less clear.

Since user browsing is recorded, both the profile accuracy and profile predictive accuracy can be measured. Profile accuracy measures the number of papers jumped to or browsed that match the top 3 profile topics for the duration of that profile; since profiles are updated daily, the average duration of a profile is one day. This is a good measure of the accuracy of the current interests within a profile at any given time. Profile predictive accuracy measures the number of papers jumped to or browsed that match the top 3 profile topics in a 4-week



Final week 95% confidence intervals			
Profile [topic]	0.27	Relevance [topic]	0.041
Profile [rec]	0.14	Relevance [rec]	0.039

Fig. 25. Jumps to recommendations and profile topics.



Final week 95% confidence intervals			
Profile [profile]	0.23	Relevance [profile]	0.036
Profile [predictive]	0.23	Relevance [predictive]	0.036

Fig. 26. Profile accuracy and profile predictive accuracy.

period after the profile was created. This measures the ability of a profile to predict subject interests. Metrics are measured for every profile computed over the period of the trial, providing a view on how the quality of the profiles varies over the length of the trial. Figure 26 shows the figures for the profile metrics.

While not statistically significant, there is a trend for the "profile feedback" group to have profiles that are better at predicting future browsing interests.

Table II. Classifier Accuracy and Coverage

	Accuracy	Coverage	Classes	Examples	Terms
Classifier	0.42	1.00	97	714	1152

Table III. Foxtrot Post Trial Questionnaire Results

Question	1	2	3	4	5	Mean
How useful did you find the Foxtrot database?	4	2	5	2		2.38
How much did you use the recommendation facility?	7	5		1		1.62
How accurate were the recommended topics?	3	3	2	3	1	2.67
How useful were the recommended papers?	4	2	4		2	2.5

This trend is not reflected in the daily profile accuracy figures however, where the two groups are similar. This would appear to show that the two groups are profiling slightly different interest sets, with the "profile feedback" interests of a longer-term nature.

In addition to measuring subject group interactions with the system, the AdaBoostM1 boosted IBk classifier performance was computed. A standard cross-validation test was applied to the classifier training set, to obtain the figures for accuracy and coverage. Table II shows the results. The accuracy value is a measure of how many correctly classified documents there were as a proportion of the number classified. The coverage value is a measure of how many documents were classified as a proportion of the total number of documents.

A post-trial questionnaire was sent out via e-mail to every subject who used the system at least once. Table III shows the results of this survey, completed by 13 subjects. It shows that the search facility was most useful to the subjects, with the recommendation facility being only partially used. This is borne out by the relatively small amount of feedback provided by users during the trial. The most positive comments were from those users who were interested in general papers in an area, such as Ph.D students performing a literature review. The more negative comments came from those subjects wanting papers on very specific topics of much finer granularity than the research topic ontology offered.

4.3.3 *Discussion.* The "profile feedback" group outperformed the "relevance feedback" group for most of the metrics, and the experimental data revealed several trends.

Web page recommendations, and jumps to those recommendations, were better for the "profile feedback" group, especially early on in the first few weeks after registering. This is probably because the "profile feedback" users tended to draw interest profiles when they first registered with the system, and only update them occasionally afterwards. This has the effect that the profiles are most accurate early on and become outdated as time goes by. This aging effect on the profile accuracy is shown by the "profile feedback" group performance gradually falling towards that of the "relevance feedback" group. One interesting observation is that the initial performance enhancement gained using profile feedback appears to help overcome the cold-start problem, a problem inherent to all recommender systems.

E-mail recommendation appeared to be preferred by the "relevance feedback" group, and especially by those users who did not regularly check their web page recommendations. A reason for this could be that since the "profile feedback" group used the web page recommendations more, they needed to use the e-mail recommendations less. There is certainly a limit to how many recommendations any user needs over a given time period; in our case nobody regularly checked for recommendations more than once a week.

The overall recommendation accuracy was about 1%, or 2–5% for the profile feedback group. This may appear low, especially when compared to other recommendation systems such as Quickstep, but it reflects the nature of the recommendation service offered. Users had the choice to simply ignore recommendations if they did not help to achieve their current work goal. This optional nature of the system assisted system uptake and acceptance on a wide scale.

The profile accuracy of both groups was similar, but there was a significant difference between the accuracy of profile predictions. This reflects the different types of interests held in the profiles of the two groups. The "profile feedback" group's profiles appeared to be longer term, based on knowledge of the users general research interests provided via the profile interface. The "relevance feedback" profiles were based solely on the browsing behavior of the users current task, hence contained shorter-term interests. Perhaps a combination of profile feedback-based longer-term profiles and behavior-based short-term profiles would be most successful.

The overall profile accuracy was around 30%, reflecting the difficulty of predicting user interests in a real multitask environment. Integrating some knowledge of which task the user is performing would allow access to some of the other 70% of their research interests. These interests were in the profile but did not make it to the top 3 topics of current interest.

Profile feedback users tended to regularly check recommendations for about a week or two after drawing a profile. This appeared to be because users had acquired a conceptual model of how the system worked, and wanted to keep checking to see if it had done what they expected. If a profile was required to be drawn before registering on the system, this behavior pattern could be exploited to increase system uptake and gain some early feedback. This may, in turn, increase initial profile accuracy and would certainly leave users with a better understanding of how the system worked, beneficial for both gaining user trust and encouraging effective use of the system.

In order to perform such a large trial, involving the monitoring of subject web-browsing behavior over a significant period of time, a number of things had to be done concerning subject privacy rights. First, every subject was informed of the trial, and what it involved, via e-mail and a website. All aspects of the profiling and monitoring process were explained in detail. User's names were encrypted using a one-way encryption algorithm so that, if someone were to examine the web-browsing logs, they would not be able to trace usernames to network account names, and hence real people. The key to this one-way encryption was destroyed after the trial finished. Finally, in accordance with the UK's Data Protection Act, the trial was for purely research purposes. A commercial system would likely need written consent from each subject under UK law.

A post-hoc power analysis was considered after the initial experimental analysis was completed, but not performed after consultation with a statistics expert due to reservations about its value. Post analysis of the data collected would also be problematic due to the encrypted nature of the user identifiers, and lack of easy correlations between the various logged data sources other than those that were pre-planned into the experimental design.

4.4 Conclusions

This experiment shows that profile visualization and profile feedback can significantly improve the profiling accuracy and the recommendation process. Our ontological approach makes this possible because user profiles are represented in terms the users can understand.

The previous section on Quickstep compared performance to reported systems in the literature, and points out the lack of published experimental results for systems with real users. As such, the Quickstep system is an ideal candidate for result comparison.

The Quickstep [Middleton et al. 2001] system had a recommendation accuracy of about 10% with real users, while Foxtrot manages a 2–5% recommendation accuracy, reflecting the different types of subjects involved in the two experiments. The Quickstep subjects were willing researchers taken from a computer science laboratory, while the Foxtrot subjects were staff and students of a large department who would only be willing to use the system if it was perceived to offer direct benefits to their work. A recommendation accuracy of 5% means that on average 1 in 2 sets of recommendations contained a paper that was downloaded, while 10% means on average every set of recommendations contains a downloaded paper. While initially appearing low, this result is good when the problem domain is taken into account; most systems in the literature do not attempt such a hard and realistic problem.

Individual aspects of the Foxtrot system could be enhanced further to gain a relatively small performance increase, such as increasing the training set size, fine tuning the ontological relationships and trying alternative classification algorithms. However, the main problem is that the system's profiler is not capturing about 70% of the user's interests. We expect major progress to come from expanding the ontology and using a task model for profiling, which are discussed in the next section.

5. CONCLUSIONS

Our ontological approach to recommender systems offers many advantages and a few disadvantages. The two experimental systems and three experiments conducted with them provide evidence for this. Due to the attenuating nature of real world trials with noisy data and varying levels of subject activity, some of the trends seen are not significant statistically. However, we do feel the power and consistency of the trends seen are significant, and it is our opinion that the advantages of our ontological approach clearly outweigh the disadvantages.

Ontological user profiles allow inference to be employed, allowing interests to be discovered that were not directly observed in the user's behavior.

Constraining examples of user interest to a common ontology also allows examples of ontological classes to be shared among all users, increasing the size of the classifiers training set. Multiclass classification is, however, inherently less accurate that binary class classification, which reduces classification accuracy. Our first experiment quantifies these effects and demonstrates that profile inference compensates for the lower classifier accuracy.

Once profiles are represented using an ontology, they can communicate with other ontologies that share similar concepts. This allows external knowledge bases to be employed to help bootstrap the recommender system and reduce the cold-start problem inherent to all recommender systems. Our second experiment demonstrates this, using a publication and personnel ontology to bootstrap our recommender system with significant success.

One last advantage of using an ontological user profile is that the profiles themselves can be visualized. Since our research paper ontology contains terms understandable to users, the profile visualizations are understandable too. Traditional binary profiles are often represented as term vector spaces, neural network patterns etc. that are difficult to understand by users. The ontological representation allows users to provide feedback on their own profiles, which is used to significantly improve profile accuracy. Our third experiment demonstrates this.

There is a lack of experimental results in the literature for systems using real people. This is a failing of this research field, and it makes direct comparison of systems that address real problems hard. Our final experiment is particularly valuable in that it shows how a recommender system performs in a large scale, realistic situation. We feel that more large-scale trials are needed in the literature so that the utility of the recommender system paradigm can be quantified for a variety of work domains.

5.1 Future Work

Expanding the ontology to include more relationships than just is-a links between topics would allow much more powerful inference, and thus give a significant boost to profiling accuracy. Knowledge of the projects people are working on, common technologies in research areas and linked research areas would all help. This technology could also help the cold-start problem.

Knowledge of a user's current task would allow the profiler to distinguish between short and long-term tasks, separate concurrently running tasks and adjust recommendations accordingly. While 70% of users' browsing interests were not in the current profile's top 3 topics, they were in the profile somewhere at a lower level of relevance. Having separate profiles for each user task would allow a finer grained profiling approach, significantly improving performance. This is far from easy to achieve in practice, but it appears to be an important aspect of user profiling and one that future versions of this system may well investigate. Papers such as Budzik et al. [2001] examine the use of contextual information in task modelling.

An agent-based metaphor can easily be applied to our ontological recommender system and would allow extra information to come from external

agents, via free exchange or trading. It is easy to see a situation where external agents, with ontologies containing personal information, interact with profile agents to share knowledge about specific interests with the goal of improving each other's profiles.

REFERENCES

ALANI, H., DASMAHAPATRA, S., O'HARA, K., AND SHADBOLT, N. 2003. ONTOCOPI—Using ontology-based network analysis to identify communities of practice. *IEEE Intell. Syst. 18*, 2, 18–25.

AHA, D., KIBLER, D., AND ALBERT, M. 1991. Instance-based learning algorithms. *Mach. Learn. 6*, 37–66.

BALABANOVIĆ, M. AND SHOHAM, Y. 1997. Fab: Content-based, collaborative recommendation. *Commun. ACM 40*, 3, 67–72.

BILLSUS, D. AND PAZZANI, M. J. 2000. User modelling for Adaptive News Access. In *User Model. User-Adapt. Interact. 10*, 147–180.

BOLLACKER, K. D., LAWRENCE, S., AND GILES, C. L. 1998. CiteSeer: An autonomous web agent for automatic retrieval and identification of interesting publications. In *Autonomous Agents 98* (Minneapolis, Minn.).

BUDZIK, J., HAMMOND, K., AND BIRNBAUM, L. 2001. Information access in context. *Knowl. Based Syst. 14* (1–2), 37–53.

BURKE, R. 2000. Knowledge-based recommender systems. In *Encyclopaedia of Library and Information Systems*, vol. 69, Supplement 32. A. Kent, Ed.

CLAYPOOL, M., GOKHALE, A., AND MIRANDA, T. 1999. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)* (Berkeley, Calif.). ACM, New York.

CRAVEN, M., DIPASQUO, D., FREITAG, D., MCCALLUM, A., MITCHELL, T., NIGAM K., AND SLATTERY, S. 1998. Learning to extract symbolic knowledge from the world wide web. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*.

DELGADO, J., ISHII, N., AND URA, T. 1998. Intelligent collaborative information retrieval. In *Proceedings of Artificial Intelligence (IBERAMIA'98)*. Lecture Notes in Artificial Intelligence Series No. 1484.

ERIKSSON, H., FERGESON, R., SHAHR, Y., AND MUSEN, M. 1999. Automatic generation of ontology editors. In *Proceedings of the 12th Workshop on Knowledge Acquisition, Modelling, and Management (KAW'99)* (Ban, Alberta, Canada).

FREUND, Y. AND SCHAPIRE, R. E. 1996. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*.

GUARINO, N. AND GIARETTA, P. 1995. Ontologies and knowledge bases: Towards a terminological clarification. In *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, N. Mars, Ed. IOS Press, 25–32.

GUARINO, N., MASOLO, C., AND VETERE, G. 1999. OntoSeek: Content-based access to the web. *IEEE Intell. Syst. 14*, 3.

KOBSA, A. 1993. User modeling: Recent work, prospects and Hazards. In *Adaptive User Interfaces: Principles and Practice*, M. Schneider-Hufschmidt, T. Kühme, and U. Malinowski, Eds. North-Holland, Amsterdam, The Netherlands.

KONSTAN, J. A., MILLER, B. N., MALTZ, D., HERLOCKER, J. L., GORDON, L. R., AND RIEDL, J. 1997. GroupLens: Applying collaborative filtering to usenet news. *Commun. ACM 40*, 3, 77–87.

LANG, K. 1995. NewsWeeder: Learning to filter NetNews. In *ICML95 Conference Proceedings*, 331–339.

LARKEY, L. S. 1998. Automatic essay grading using text categorization techniques. In *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval* (Melbourne, Australia).

MALTZ, D. AND EHRLICH, E. 1995. Pointing the way: Active collaborative filtering. In *Proceedings of the CHI'95 Human Factors in Computing Systems*. ACM. New York.

McCallum, A. K., Nigam, K., Rennie, J., and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Inf. Retri. 3*, 2, 127–163.

Melville, P., Mooney, R. J., and Nagarajan, R. 2002. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-2002)* (Edmonton, Ont., Canada).

Middleton, S. E., Alani, H., Shadbolt, N. R., and de Roure, D. C. 2002. Exploiting synergy between ontologies and recommender systems. In *International Workshop on the Semantic Web, Proceedings of the 11th International World Wide Web Conference WWW-2002* (Hawaii).

Middleton, S. E., de Roure, D. C., and Shadbolt, N. R. 2001. Capturing knowledge of user preferences: Ontologies on recommender systems. In *Proceedings of the 1st International Conference on Knowledge Capture (K-CAP 2001)* (Victoria, B.C., Canada).

Mladenić, D. 1996. Personal WebWatcher: Design and implementation. Tech. Rep. IJS-DP-7472, Department for Intelligent Systems, J. Stefan Institute.

Mladenić, D. and Stefan, J. 1999. Text-learning and related intelligent agents: A survey. *IEEE Intell. Syst*. 44–54.

Nwana, H. 1996. Software agents: An overview. *The Knowl. Eng. Rev. 11*, 3, 205–244.

O'hara, K., Shadbolt, N., and Buckingham Shum, S. 2001. *The AKT Manifesto*. http: www.aktors.org/publications/manifesto.pds.

Porter, M. 1980. An algorithm for suffix stripping. *Program. 14*, 3, 130–137.

Rashid, A., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., and Riedl, J. 2002. Getting to know you: Learning new user preferences in recommender systems. In *Proceedings of the IUI'02* (San Francisco, Calif.).

Schein, A. L., Popescul, A., and Ungar, L. H. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the SIGIR'02* (Tampere, Finland).

Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Comput. Surv*.

Shadbolt, N., O'hara, K., and Crow, L. 1999. The experimental evaluation of knowledge acquisition techniques and methods: history, problems and new directions. *Int. J. Hum.-Comput. Stud. 51*, 729–755.

Smart Staff 1974. User's Manual for the SMART Information Retrieval System. Tech. Rep. 71–95, Cornell University.

Received September 2002; revised April 2003; accepted September 2003