Ontology-Based Recommender Systems

Stuart E. Middleton[1], David De Roure[2], and Nigel R. Shadbolt[2]

[1] IT Innovation Centre, University of Southampton, Southampton SO16 7NP, UK,
sem@it-innovation.soton.ac.uk

[2] Intelligence, Agents, Multimedia Group, Department of Electronics and
Computer Science, University of Southampton, Southampton SO17 1BJ, UK,
dder@ecs.soton.ac.uk, nrs@ecs.soton.ac.uk

Summary. We present an overview of the latest approaches to using ontologies in recommender systems and our work 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. 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.

In a specific case study we report results from 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. The overall performance of our ontological recommender systems are favourably compared to other systems in the literature.

1 Introduction

The mass of content available on the World-Wide Web raises important questions over its effective use. Information on the web is largely unstructured, with web pages authored by many people on a diverse range of topics. This often makes simple browsing too time consuming to be practical. The emergence of e-commerce sites means many vendors are offering potentially great deals on very similar products. Web information filtering has thus become necessary for most web users in order to find the things they really need.

Recommender systems have emerged as one successful approach that can help tackle the problem of information overload. They exploit patterns in item metadata and reviews posted by groups of people to find new items that might

S. Staab and R. Studer (eds.), *Handbook on Ontologies,* International Handbooks on Information Systems, DOI 10.1007/978-3-540-92673-3,

© Springer-Verlag Berlin Heidelberg 2009

be of interest to a user. Ontologies are increasingly being used within the field of recommender systems, allowing knowledge-based techniques to supplement classical machine learning and statistical approaches.

1.1 Recommender Systems

People find articulating exactly what they want difficult, but they are good at recognizing it when they see it. This insight has led to the utilization of relevance feedback, where people rate items as interesting or not interesting and the system tries to find items 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. Recommender systems can recommend many types of item, including web pages, new articles, music CDs and books.

Unobtrusive monitoring provides positive examples of what the user is looking for, without interfering with the user’s 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 [15].

Hybrid systems, attempting to combine the advantages of content-based and collaborative recommender systems, have also 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 behaviour-based. Knowledge-based approaches use static models of users and dynamically match users to the closest model. Questionnaires and interviews are often employed to obtain this user knowledge. Once a model is selected for a user, specialist domain knowledge for that user type can be applied to help the user. Behaviour-based approaches use the user’s behaviour as a model, commonly using machine-learning techniques to discover useful patterns in the behaviour. Behavioural logging is employed to obtain the data necessary from which to extract patterns. Kobsa [9] provides a good survey of user modelling techniques.

The user profiling approach used by most recommender systems is behavioural-based, commonly using a binary class model to represent what users find interesting and not interesting. Machine-learning techniques are then used to find potential items of interest in respect to the binary model, recommending items that match the positive examples and do not match the negative examples. 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 [19] 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 [8]. 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. Part III contains examples of the types of ontology that are in use today, such as chapter "COMM: A Core Ontology for Multimedia Annotation".

Ontologies help extend recommender systems to a multi-class environment, allowing knowledge-based approaches to be used alongside classical machine learning algorithms. Section 2 provides an in-depth overview of how ontologies are integrated into the techniques used for recommendation. Part IV of this book contains details on the current best practice for supporting infrastructures and for ontologies, especially chapters "Ontology Repositories" and "Ontology Mapping".

1.4 Chapter Structure

In this chapter we show how ontologies are used in recommender systems today, providing an overview of the technology space and some further reading on specific approaches. We then examine in some depth a case study of two recommender systems that were among the first to adopt ontological techniques. In these case studies the problem domain, algorithms and results are detailed along with a discussion that highlights some of the practical difficulties experienced running a recommender system for real.

2 Ontology Use in Recommender Systems

Ontologies are now used routinely in recommender systems in combination with machine learning, statistical correlations, user profiling and domain specific heuristics. Commercial recommender systems generally either maintain simple product ontologies (e.g. books) that they can then utilize via heuristics or have a large community of users actively rating content (e.g. movies) suitable for collaborative filtering. More research oriented recommender systems use a much wider variety of techniques that offer advantages such as improved accuracy coupled with constraints such as requiring explicit relevance feedback or intrusive monitoring of user behaviour over prolonged periods of time.

Recommendation of new items to users can be performed by looking at item to item similarity (content-based filtering), item reviews within a community of users (collaborative filtering), semantic relationships between items (heuristic-based recommendation) or a hybrid approach. In many cases the type of approach adopted will depend heavily on how much metadata is available about the items and how much user feedback is available, both implicit and explicit. Content-based techniques work well if training data is available in advance. Collaborative techniques work well when a system has a large community of users. There are, however, no definitive rules to decide on an approach and normally experience and expertise is required to pick the best approach for a given problem domain.

2.1 Content-Based Recommendation

Early recommender systems used content-based binary classification approaches looking at training sets of what was, and what was not interesting to a specific user. Machine learning techniques were employed to perform supervised learning based on sets of observed training examples that a user labelled either as “good” or “bad”. A classic example of a content-based recommender system is Fab [1], which uses a binary class k-Nearest Neighbour classifier. Other binary class examples include personal assistant agents such as NewsDude [2], using a naive Bayes classifier, and NewsWeeder [11], using a TF-IDF based classifier, which profile individual user interests and try to find items of interest.

To enhance binary classification domain ontologies were introduced allowing multi-class classification and hence multi-class recommendation. Typically the classes in a domain ontology, such as a product ontology defining all the products of an e-commerce website, would be used to classify the previously observed products / web pages a user had purchased / viewed. A good example of multi-class recommendation is RAAP [4], which uses a simple set of categories to represent individual user profiles.

Once a domain has been classified in terms of ontological concepts the relationships defined by the domain ontology can be used to infer interest and relevance of one concept from observed interest in another. A knowledge-based system can use expert system rules to infer probabilistic interest in classes of item with a semantic connection to an observed item of interest. Typically the semantic distance (number of relationships away one topic is from another) is used to calculate semantic similarity, and this is used to weight likely interest. Entre [3] is a restaurant recommender system that uses a knowledge-base and heuristic rules for recommendation. Where users articulate queries via a web interface the query criteria can drive a knowledge-based decision tree for advanced query refinement. The CWAdvisor [5] system is an example of such an approach where a finite state model is used to refine queries for available financial service products that match the user's stated requirements.

2.2 Clustering and Topic Diversification

Some domains do not have well identified classes of item from which content can be classified. In these cases recommender systems have employed clustering techniques to identify within groups of items potentially similar classes. Hierarchical clustering has been used to categorize document collections for recommender algorithms [18] and sub-divides into either distance-based clustering or concept-based clustering.

Distance-based clustering [21] takes either a top-down (partitioning) or bottom-up (agglomerative) approach to building a hierarchical class tree. A distance function is defined to compute similarity between documents, often based on the similarity of frequency of the words within the document. The clustering algorithm iterates, either dividing super-clusters or merging small clusters into larger ones, until the final concept tree is formed.

Concept-based clustering takes items represented as attribute-pairs and builds relationships based on the probability of occurrence of attribute-pairs within nodes. An early example of concept-based clustering is the COBWEB [6] algorithm. Nodes are created in a top-down approach where nodes are split or merged according to a category utility value; category utility is a measure of differentiation power of that node.

Often recommender systems will recommend clusters of items that are very similar, or variants of the same item (e.g. different formats of the film/DVD). To avoid this topic diversification [22] can be employed to ensure each recommendation is on a well defined concept, hopefully increasing the usefulness of a set of recommendations to the user. Algorithms to perform topic diversification will compute a dissimilarity ranking and merge this with the recommendation ranking. Semantic distance and super-class relationships can be used to compute dissimilarity between item sets.

2.3 Collaborative Filtering

Collaborative filtering works by using the ratings provided by a community of users to recommend items for a specific user. There are two complementary approaches available, user-based or item-based collaborative filtering. User-based collaborative filtering is where similar users are found and items recommended that these similar users also liked. Item-based collaborative filtering is where items are grouped if people rate them similarly.

In order to perform collaborative filtering a user profile must be created from the available historical records of what items people have reviewed and rated. Often a 5-point scale is used for ratings (very good to very bad). A common user profile representation is a weighted vector if interest with as many dimensions as the domain has classes. Vectors can also be used for item to item similarity. Domains where item metadata is not accessible as ontological terms will usually apply pre-processing techniques to compute word/document/metadata term frequencies, remove common words and merge similar words using a thesaurus like WordNet.

User-based collaborative filtering is the most popular recommendation algorithm due to its simplicity and excellent quality of recommendation. First neighbourhoods are formed using a similarity metric, such as a statistical correlation metric like Pearson-r correlation. Second a set of rating predictions are created using profiles that are within the same neighbourhood as the user's own profile. Recommendations are created from the top-N items. The GroupLens project [10] is an early exploiter of user-based collaborative filtering.

Item-based collaborative filtering has become popular in the last 5 years since it decouples the model computation from the prediction process; Amazon [13] have used this technique successfully. Just as in user-based similarity items are compared on the basis of how many users rank them similarly. The neighbourhoods computed are therefore collections of items that are similar. This technique scales well since new items will be added to neighbourhoods as users rate them without the need for explicit ontology maintenance.

Sometimes a recommender system will have to compare items from different domain ontologies, such as two product lists. In these cases an ontology can be created for both domains in a common language (such as OWL) and the mapping between them formulated, either manually or using a automated technique [12] such as a Bayesian belief network. Once concepts are successfully mapped the normal approaches for recommendation can be applied.

2.4 Use of the Semantic Web and Web 2.0 Approaches

Recent work has also used some of the emerging Web 2.0 resources from the Semantic Web to help identify classes of item. One such system [20] has used an internet movie database that contains extensive information about actors, movies, etc., and mapped this semantic information to user behaviour on a movie recommendation website. Tag clouds are created based on the keyword frequencies behind the items they have rated. Data mining techniques [4] can also be coupled with ontological knowledge to improve similarity matching and recommendation within historical usage data.

3 Case Study: Two Ontological Recommender Systems

For a case study two experimental recommender systems are presented, Quickstep and Foxtrot, that explored the novel idea of using an ontological approach to user profiling in the context of recommender systems. Representing user interests in ontological terms involves losing some of the fine grained information held in the raw examples of interest, but in turn allows inference to assist user profiling, communication with other external ontologies and visualization of the profiles using ontological terms understandable to users. Figure 1 shows the general approach taken by both our recommender systems. Quickstep implements only the basic recommendation interface, while Foxtrot implements all the shown features.

A research paper topic ontology is shared between all system processes, allowing both classifications and user profiles to use a common terminology. The ontology itself contains is-a relationships between appropriate topic classes; a section from the topic ontology is shown in Fig. 2. The Quickstep ontology was

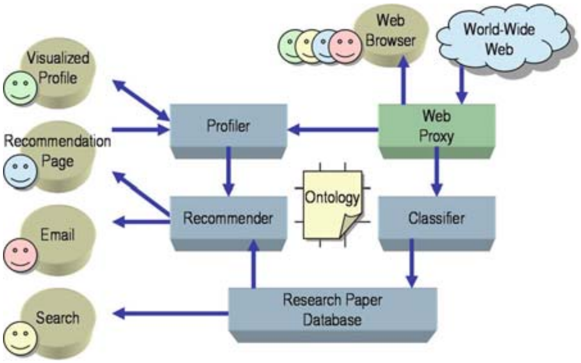


Fig. 1. Quickstep and Foxtrot recommender system data flow

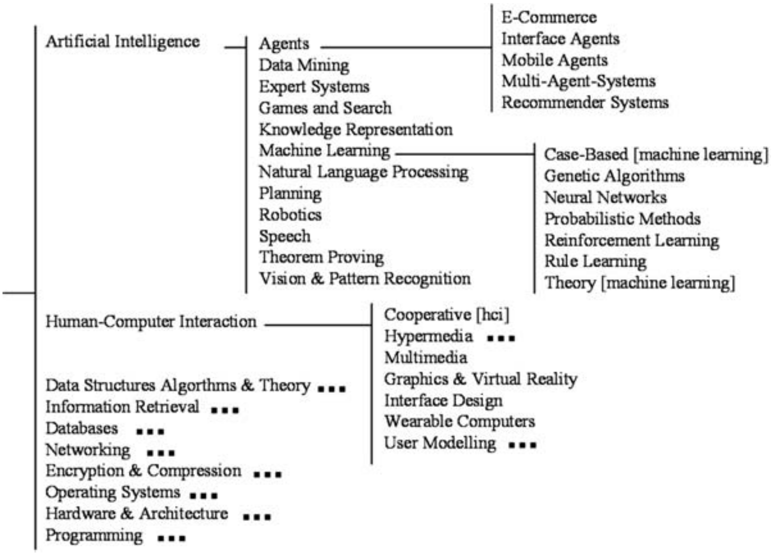


Fig. 2. Section from the Foxtrot research paper topic ontology

based on the open directory project's [7] computer science topic classification, while the Foxtrot ontology was based on the CORA [14] digital library paper classification; manual enhancements were made to each ontology to better reflect some of the more specialist sub-topics researchers required. Reusing existing classifications saves time and provides a source for training examples, especially with the CORA digital library, which contained many pre-classified research papers.

3.1 Classification Using a Research Paper Topic Ontology

Sharing training examples, within the structure of an ontology, allows for much larger training sets than would be possible if a single user just provided examples of personal interest. Larger training sets improve classifier accuracy. However, multi-class classification is inherently less accurate than binary class classification, so the increased training set size has to be weighed along with the reduction in accuracy that occurs with every extra class the system supports.

Both the Quickstep and Foxtrot recommender systems use the research paper topic ontology to base paper classifications upon. A set of labelled example papers is manually provided for each class within the ontology, and then used by the classifier as a labelled training set. In the Quickstep system users can add new examples of papers as time goes by, allowing the training set to reflect the continually changing needs of the users.

In addition to larger training sets, having users share a common ontology enforces a consistent conceptual model, which removes some of the subjective nature of selecting categories for research papers. A common conceptual model also helps users to understand how the recommender system works, which helps form reasonable user expectations and assists in building trust and a feeling of control over what the system is doing.

3.2 Ontological Inference to Assist User Profiling

Ontological inference is a powerful tool to assist user profiling. An ontology could contain all sorts of useful knowledge about users and their interests, such as related research subjects, technologies behind each subject area, projects people are working on, etc. This knowledge can be used to infer more interests than can be seen by just observation.

Our two experimental recommender systems both use ontological inference to enhance user profiles. Is-a relationships within the research paper topic ontology are used to infer interest in more general, super-class topics. We add 50% of the interest in a specific class to the super-class. This inference has the effect of rounding out profiles, making them more inclusive and attuning them to the broad interests of a user.

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

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. 3. Profiling algorithm

The profiling algorithm used is shown in Fig. 3. A time-decay function is applied to the observed behaviour events to form the basic profile. Inference is then used to enhance the interest profile, with the 50% inference rule applied to all ontological is-a relationships, up to the root class, for each observed event.

The event interest values were chosen to balance the feedback in favour of explicitly provided feedback, which is likely to be the most reliable. The 50% inference value was chosen to reflect the reduction in certainty you get the further away from the observed behaviour you move. Determining optimal values for these parameters would require further empirical evaluation.

3.3 Bootstrapping with an External Ontology

Recommender systems suffer from the cold-start problem [15], where the lack of initial behavioural information significantly reduces the accuracy of user profiles, and hence recommendations. This poor performance can deter users from adopting the system, which of course prevents the system from acquiring more behaviour data; it is possible that a recommender system will never be used enough to overcome its cold-start.

In one of our experiments we take an external ontology containing publication and personnel data about academic researchers and integrate it with the Quickstep recommender system. The knowledge held within the external ontology is used to bootstrap initial user profiles, with the aim of reducing the cold-start effect. The external ontology uses the same research topic ontology as the Quickstep system, providing a firm basis for communication. The external ontology contains publications and authorship relationships, projects and project membership, staff and their roles and other such knowledge. Knowledge of publications held within the external ontology is used to infer historical interests for new users, and network analysis of ontological relationships is used to discover similar users whose own interests might be used to bootstrap a new user's profile.

The two bootstrapping algorithms used in our experiment are shown in Figs. 4 and 5. The new-system initial profile algorithm takes all the publications of a user and creates a profile of historical interests. The assumption is that a user's previous publications indicate that user's interests. The new-user initial profile algorithm takes a set of similar users, obtained via network analysis of the external ontologies project membership and inter-staff relationships, and includes these users interests into the bootstrap profile. Historical publication interests from the new user are also added as before. The parametric values shown in Figs. 4 and 5 were empirically determined after several experimental runs using test data.

In addition to using the ontology to bootstrap the recommender system, our experiment uses the interest profiles held within the recommender system to continually update the external ontology. Interest acquisition is a problematic task for ontologies that are based on static knowledge sources, and this synergistic relationship provides a useful source of personal knowledge about individual researchers.

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

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

t = <research paper topic>

Interest value for super-class per topic = 50% of sub-class

Fig. 4. New-system initial profile algorithm

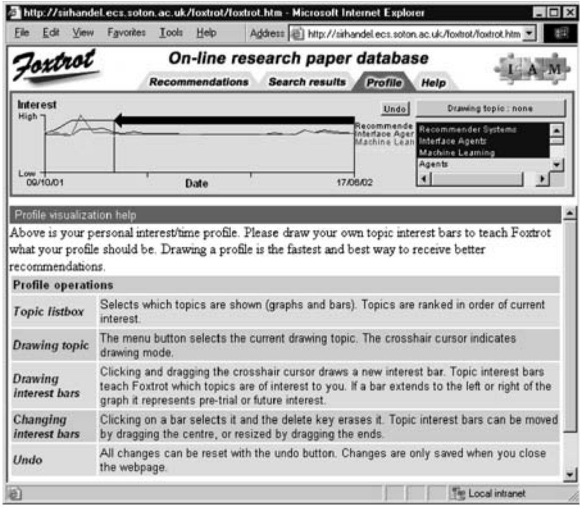
$$\text{topic interest}(t) = \frac{\gamma}{N_{\text{similar}}} \sum_{1..\,N_{\text{similar}}}^{u} \text{profile interest}(u,t)$$
$$+ \sum_{1..\,N_{\text{pubs}\,t}}^{n} 1 / \text{publication age}(n)$$

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 similarity of user

Interest value for super-class per topic = 50% of sub-class

Fig. 5. New-user initial profile algorithm



Profile visualization help

Above is your personal interest/time profile. Please draw your own topic interest bars to teach Foxtrot what your profile should be. Drawing a profile is the fastest and best way to receive better recommendations.

Profile operations	
Topic listbox	Selects which topics are shown (graphs and bars). Topics are ranked in order of current interest.
Drawing topic	The menu button selects the current drawing topic. The crosshair cursor indicates drawing mode.
Drawing interest bars	Clicking and dragging the crosshair cursor draws a new interest bar. Topic interest bars teach Foxtrot which topics are of interest to you. If a bar extends to the left or right of the graph it represents pre-trial or future interest.
Changing interest bars	Clicking on a bar selects it and the delete key erases it. Topic interest bars can be moved by dragging the centre, or resized by dragging the ends.
Undo	All changes can be reset with the undo button. Changes are only saved when you close the webpage.

Fig. 6. Foxtrot profile visualization interface

3.4 Profile Visualization Using Ontological Concepts

Since users can understand the topics held within the ontology, the user profiles can be visualized. These visualizations allow users to see what the system thinks they are interested in and hence allow them to gain an insight into how the system works. Profile visualization thus provides users with a conceptual model of how the profiling algorithm works, allowing users to gain trust in the system and providing users with a feeling of control over what's going on. With a better conceptual model user expectations should be more realistic.

The Foxtrot recommender system visualizes profiles using a time/interest graph. In addition to simply visualizing profiles, a drawing package metaphor is used to allow users to draw interest bars directly onto the time/interest graph. This allows the system to acquire direct profile feedback, which can be used by the profiler to improve profile accuracy and hence recommendation accuracy. Figure 6 shows the profile visualization interface.

4 Case Study: Experimentation Results

We have conducted three experiments with our two recommender systems. The Quickstep recommender system is used to measure the performance gain seen when using profile inference, and the reduction in the cold-start seen when an external ontology is used for bootstrapping. The Foxtrot recommender system is used to measure the effect profile visualization has on profile accuracy, and to perform a large-scale assessment of our overall ontological approach to recommender systems.

A more in-depth statistical investigation of this approach has been performed using the datasets gathered in our user trials (260 subjects, 15,792 documents) and is published in [17].

4.1 Using Ontological Inference to Improve Recommendation Accuracy

Our first experiment used the Quickstep recommender system to compare subjects whose profiles were computed using ontological inference with subjects whose profiles did not use ontological inference. 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. Subjects were taken from researchers in a computer science laboratory and split into two groups; one group used a topic ontology and profile inference while the other group used an unstructured flat list of topics with no profile inference. An overall evaluation of the Quickstep recommender system was also performed. This experiment is published in more detail in [16].

This experiment found that ontological profile users provided more favourable feedback and had superior recommendation accuracy. Figures 7 and 8 shows these results. Users provide feedback on their individual recommendations, rating them as "interesting", "uninteresting" or "no comment". Good topics are defined as those not rated as "uninteresting" by users. 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

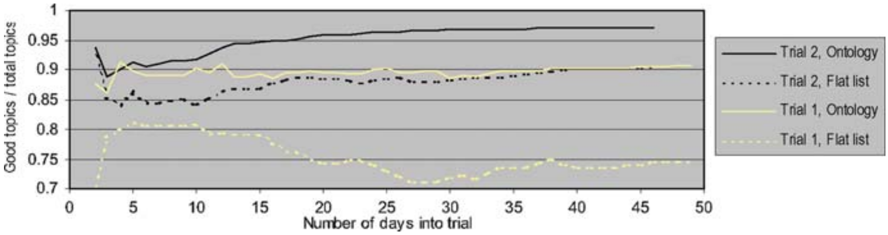


Fig. 7. Good topics to total topics ratio

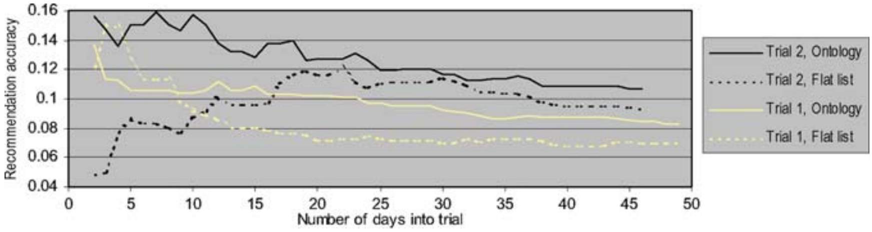


Fig. 8. Recommendation accuracy

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.

The ontology groups from the two trials 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, probably as a result of the interface improvements that made the feedback options less confusing. There is a small 1% improvement in recommendation accuracy by the ontology group. Both trials show between 7% and 11% recommendation accuracy.

Since 10 recommendations were provided at a time, a recommendation accuracy of 10% means that on average there was one good recommendation in each set presented to the user. We regard providing one good recommendation upon each visit to the recommendation web site as demonstrating significant utility.

While not statistically significant due to sample size, 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.

4.2 Ontological Bootstrapping to Reduce the Cold-Start Problem

Our second experiment integrated 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 based on a publication database and personnel database, coupled with a tool for performing network analysis of ontological relationships to discover similar users. The behavioural 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. This experiment is published in more detail in [15].

Subjects were selected from those in the previous experiment who had entries within the external ontology. We selected nine subjects in total and their URL browsing logs were broken up into weekly log entries. Seven weeks of browsing behaviour were taken from the start of the Quickstep trials, and an empty log created to simulate the very start of the trial where no behaviour has yet been recorded.

Two bootstrapping algorithms were tested, the new-system and new-user initial profile algorithms described earlier. As the new-system algorithm bootstraps a completely cold-start we tested from week 0 to week 7. The new-user algorithm requires the system to have been running for a while, so we added the new user on week 7, after the new-system cold-start was over.

Two measurements were made to measure the reduction in the cold-start. The first, profile precision, measures how many topics were mentioned in both the bootstrapped profile and benchmark profile. Profile precision is an indication of how quickly the profile is converging to the final state, and thus

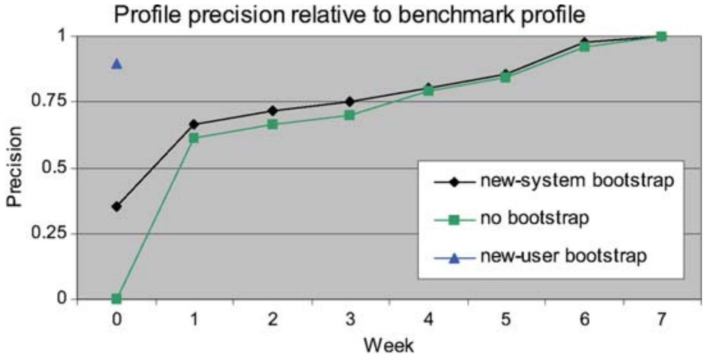


Fig. 9. Bootstrapping algorithm performance

how quickly the effects of the cold-start are overcome. The second, profile error rate, measures how many topics appeared in the bootstrapped profile that did not appear within the benchmark profile. Profile error rate is an indication of the errors introduced by the two bootstrapping algorithms. Figure 9 shows the precision results. The new-user result appears on week 0 to indicate the first week for the new-user, even though the system itself had been running for 7 weeks.

The new-system algorithm produced profiles with a low error rate of 0.06 and a reasonable precision of 0.35. This reflects that previous publications are a good indication of users current interests, and so can produce a good starting point for a bootstrap profile. The new-user algorithm achieved good precision of 0.84 at the expense of a significant 0.55 error rate.

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 and hence a profile precision of zero.

4.3 Visualizing Profiles to Improve Profile Accuracy

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. An overall evaluation of the Foxtrot recommender system was also performed.

This 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. All 260 subjects used the web proxy and hence their browsing was recorded and daily profiles built. 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. A total of nine subjects provided profile feedback.

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

Recommendation accuracy, profile accuracy and profile predictive accuracy was measured. Profile accuracy measures the number of papers jumped to or browsed that match the top three profile topics each 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 three profile topics in a 4-week period after the day the profile was created. This measures the ability of a profile to predict subject interests. Figures 10 and 11 show these results.

The "profile feedback" group outperformed the "relevance feedback" group for most of the metrics, and the experimental data revealed several trends. Email 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 email 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 third experiment shows that both profile visualization and profile feedback can significantly improve the profiling accuracy and the

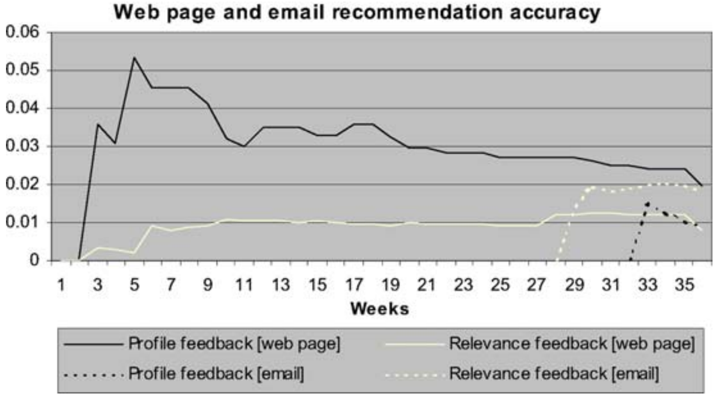


Fig. 10. Recommendation accuracy

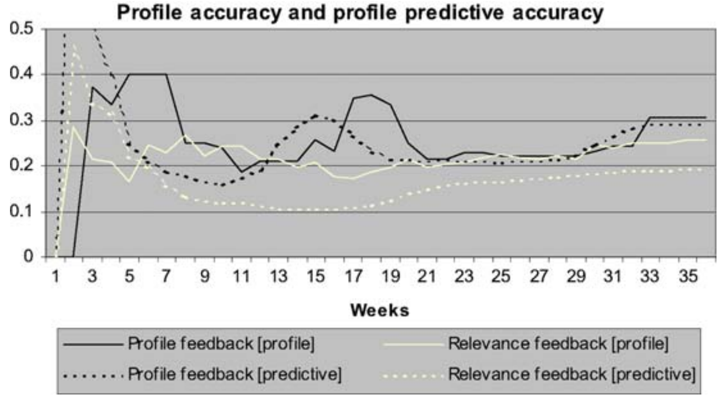


Fig. 11. Profile accuracy and predictive profile accuracy

recommendation process. Our ontological approach makes this possible because user profiles are represented in terms the users can understand.

5 Case Study: Conclusions

Through our three experiments we have demonstrated that using an ontological approach to user profiling offers significant benefits to recommender systems.

Ontological inference, even simple inference such as using is-a relationships to infer general interests, can improve profiling process and hence the recommendation accuracy of a recommender system. We achieve a 7–15% increase recommendation accuracy using just is-a relationships, and we feel it is clear that a more complete domain ontology, with more informative relationships, could perform significantly better.

External ontologies can be used to reduce significantly the cold-start problem recommender systems face. We have shown that a bootstrap profile precision of 35% is achievable given the right ontological knowledge to drawn upon. While further experimentation is required to determine exactly how good a bootstrap profile needs to be before a cold-start is avoided, it is clear that external knowledge sources offer a practical way to achieve this.

Most recommender systems hold user profiles in cryptic formats generated by techniques such as neural networks or Bayesian learners. Using an ontological approach to user profiling allows the visualization of user profiles using ontological terms users understand, and hence a way to elicit feedback on the profiles themselves. This profile feedback can be used to adjust profiles, improving their accuracy significantly. We have demonstrated increases in profiling accuracy of up to 50% of that which is achievable by traditional relevance feedback.

These three features are implemented in our two experimental recommender systems. Overall recommendation accuracy, for individual recommendations, of 7–11% for a laboratory based subject group and 2–5% recommendation accuracy for a larger department based group is demonstrated. This gives an average of one good recommendation per set of recommendations provided for the small group of about 20 users, and one every other set for the larger group of about 200 users. Both these systems compare favourably with other systems in the literature when the problem domains are taken into account.

Acknowledgements

This work was funded by EPSRC studentship award number 99308831 and the Interdisciplinary Research Collaboration In Advanced Knowledge Technologies (AKT) project GR/N15764/01.

References

1. Balabanovic M, Shoham Y (1997) Fab: Content-based, collaborative recommendation. Communications of the ACM 40(3):67–72.
2. Billsus D, Pazzani MJ (1998) A personal news agent that talks, learns and explains. In Autonomous Agents 98, Minneapolis MN, USA.
3. Burke R (2000) Knowledge-based Recommender Systems. In: Kent A (ed.) Encyclopedia of Library and Information Systems, vol. 69, supplement 32. Marcel Dekker, New York.
4. Eirinaki M, Lampos C, Paulakis S, Vazirgiannis M (2004) Web personalization integrating content semantics and navigational patterns. In Proceedings of the 6th annual ACM international workshop on Web information and data management, Washington DC, USA.
5. Felfernig A, Friedrich G, Jannach D, Zanker M (2006) An integrated environment for the development of knowledge-based recommender applications. International Journal of Electronic Commerce 11(2):11–34.
6. Fisher DH (1987) Knowledge acquisition via incremental concept clustering. Machine Learning 2(2):139–172.
7. Gerhart A (2002) Open directory project search results and ODP status. Search Engine Guide.
8. Guarino N, Giaretta P (1995) Ontologies and knowledge bases: towards a terminological clarification. In Mars N (ed.) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing. IOS Press, Amsterdam, pp. 25–32.
9. Kobsa A (1993) User modeling: recent work, prospects and hazards. In Schneider-Hufschmidt M, Khme T, Malinowski U (ed.) Adaptive User Interfaces: Principles and Practice. Elsevier Amsterdam.
10. Konstan JA, Miller BN, Maltz D, Herlocker JL, Gordon LR, Riedl J (1997) GroupLens: applying collaborative filtering to usenet news. Communications of the ACM 40(3):77–87.
11. Lang K (1995) NewsWeeder: learning to filter NetNews. In ICML95 Conference Proceedings, pp. 331–339.

12. Lee T, Chun J, Shim J, Lee S (2006) An ontology-based product recommender system for B2B marketplaces. International Journal of Electronic Commerce 11(2):125–155.
13. Linden G, Smith B, York J (2003) Amazon.com recommendations: Item-to-Item collaborative filtering. IEEE Internet Computing 7(1):76–80.
14. Mccallum AK, Nigam K, Rennie J, Seymore K (2000) Automating the construction of internet portals with machine learning. Information Retrieval 3(2):127–163.
15. Middleton SE, Alani H, Shadbolt NR, De Roure DC (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, USA.
16. Middleton SE, De Roure DC, Shadbolt NR (2001) Capturing knowledge of user preferences: ontologies on recommender systems. In Proceedings of the First International Conference on Knowledge Capture K-CAP 2001, Victoria, BC, Canada.
17. Middleton SE, Shadbolt NR, De Roure DC (2004) Ontological user profiling in recommender systems. ACM Transactions on Information Systems (TOIS) 22(1):54–88, ACM Press, New York.
18. Schickel-Zuber V, Faltings B (2007) Using hierarchical clustering for learning the ontologies used in recommendation systems. In KDD 2007, California, USA.
19. Sebastiani F (2002) Machine learning in automated text categorization. ACM Computing Surveys 34(1):1–47.
20. Szomszor M, Cattuto C, Alani H, O'Hara K, Baldassarri A, Loreto V, Servedio VDP (2007) Folksonomies, the Semantic Web, and Movie Recommendation. In Proceedings of 4th European Semantic Web Conference, Bridging the Gap between Semantic Web and Web 2.0 (in press), Innsbruck, Austria.
21. Zhao Y, Karypis G (2005) Hierarchical clustering algorithms for document datasets. Data Mining and Knowledge Discovery 10:141–168.
22. Ziegler C, McNee SM, Konstan JA, Lausen G (2005) Improving recommendation lists through topic diversification. In Proceedings of the 14th international conference on World Wide Web, Chiba, Japan.