Conceptual Recommender System for CiteSeer^x

Ajith Kodakateri Pudhiyaveetil
University of Arkansas
Fayetteville, AR 72701, USA
akodakat@uark.edu

Susan Gauch
University of Arkansas
Fayetteville, AR 72701, USA
sgauch@uark.edu

Hiep Luong
University of Arkansas
Fayetteville, AR 72701, USA
hluong@uark.edu

Joshua Eno
University of Arkansas
Fayetteville, AR 72734, USA
jeno@uark.edu

ABSTRACT
Short search engine queries do not provide contextual information, making it difficult for traditional search engines to understand what users are really requesting. One approach to this problem is to use recommender systems that identify user interests through various methods in order to provide information specific to the user's needs. However, many current recommender systems use a collaborative model based on a network of users to provide the recommendations, leading to problems in environments where network relationships are sparse or unknown. Content-based recommenders can avoid the sparsity problem but they may be inefficient for large document collections. In this paper, we propose a concept-based recommender system that recommends papers to general users of the CiteSeer^x[1] digital library of Computer Science research publications. We also represent a novel way of classifying documents and creating user profiles based on the ACM (Association for Computer Machinery) classification tree. Based on these user profiles which are built using past click histories, relevant papers in the domain are recommended to users. Experiments with a set of users on the CiteSeerX database show that our concept-based method provides accurate recommendations even with limited user profile histories.

Concepts and Subject Descriptors
H.3.3 **[Information Storage and Retrieval]**: Information Search and Retrieval

General Terms
Algorithms, Performance, Design, Economics, Reliability, Experimentation.

Keywords
Recommender system, Conceptual Recommendation, CiteSeerX, Information Retrieval

1 INTRODUCTION
With the enormous growth and complexity of information on the Web, it has been a challenging task for users to find exactly what they are looking for or find relevant, interesting information. This has lead to the development of recommender systems which recommend items to the users by capturing their interests and needs [8]. Recommender systems in real-world applications have been designed to acquire information by explicitly asking users to rate a set of titles or by implicitly watching the browsing or purchasing behavior of users [10]. For example, Netflix asks users to rate movies they have seen in the past in order to determine their likes and dislikes. Amazon uses both active ratings and passive behavior tracking to for its item recommender system.

However, the current generation of recommender systems requires further improvements to make recommendation methods more effective and applicable to an even broader range of real-life applications [1]. Many existing systems suffer from the cold-start problem of handling new items or new users [5]. In a collaborative system, for example, new items cannot be reliably recommended until they have been rated by several users. In some cases the number of users might be small compared to the number of items, which causes sparcity problems in many algorithms. Recommendations for items that are new to the catalog are therefore considerably weaker than more widely rated products. To avoid this cold start problem, content-based recommendation systems recommend items based on item content and a profile of the user's interests.

In this paper, we focus on document recommendation services, which are valuable to users of digital libraries [9]. We propose a concept-based recommender system that recommends papers to general users of CiteSeer^x digital library of Computer Science research publications. CiteSeer^x is the successor of the CiteSeer [4], which was the first digital library of technical papers and included a search engine to provide citation indexing and citation linking using the method of autonomous citation indexing [7].

Traditional content-based recommender systems use a TF-IDF method to find the top "n" words from each document a user visits and then uses these keywords to locate recommendations. For example, [11] performs its recommendations by using the top keywords as a query to a search engine. However, this model relies heavily on the exact keyword match and does not consider ambiguities present in natural language such as synonyms and polysemy. It is also inscrutable to users, who may have a hard time determining which words in their profiles are important and which may be skewing their results to irrelevant recommendations. In contrast, our method creates a concept-based user profile for each user, based on visited documents, based on the ACM Computing Classification System (CCS) [12]. The categorization process, used to categorize both visited documents and documents in the collection, can serve to disambiguate keyword terms. By summarizing the profile using just a few concepts, it is also easier for a user to manage and understand. Our recommender system makes use of this user profile to suggest documents the user might be interested in.

[1] http://CiteSeerx.ist.psu.edu/

RecSys'09, October 23–25, 2009, New York, New York, USA.

In this paper, we report on our initial work on the content-based recommender system for users of the CiteSeer^x digital library. We have found that our system provides good accuracy while avoiding many of the problems of both collaborative and keyword approaches.

2 RELATED WORK

Traditional recommender systems are usually classified based on what kind of information they use and on how they use that information [1]. The kind of information used can be content-based, collaborative, or a hybrid approach. Content-based approaches compare the contents of the item to the contents of items the user has shown interest in previously. Collaborative systems determine similarity based on collective user-item interactions, rather than on any explicit content of the items.

The information in from either content or collaborative approaches can be used for either memory-based or model-based algorithms. Memory-based systems calculate recommendations on-the-fly based on previous user activities. Model-based systems use training set to train a model, and then use the trained model to make recommendations. Memory-based methods are simpler, seem to work reasonably well in practice, and new data can be added easily and incrementally. However, this approach can become computationally expensive, in terms of both time and space complexity, as the size of the database grows. For model-based algorithms, the model itself may offer added value beyond its predictive capabilities by highlighting certain correlations in the data, offering an intuitive rationale for recommendations, or simply making assumptions more explicit [10].

Our system is content-based with both memory-based and model-based attributes. Accordingly, we focus here on content-based and hybrid approaches that use either memory- or model-based systems.

Adomavicius et al. have presented an interesting survey of recent existing recommender systems. They found that since content-based systems are designed mostly to recommend text-based items, the content in these systems is usually described with keywords [1]. For example, a content-based component of the Fab system [2], which recommends Web pages to users, represents Web page content with the 100 most important words. Basu et al [3] model the task of assigning technical papers to conference reviewers as a problem of recommending technical papers to the authors based on their interests and background. They showed that their content-based retrieval methods can exceed the performance of collaborative methods within the context of peer reviewing of papers. In contrast, our work uses far fewer distinct concepts for each profile, and the concepts correspond to the already-familiar ACM concepts.

Chandrasekan et al. presented work similar to our concept categorization approach in [6]. The system builds profiles for authors based on their authored papers in the CiteSeer database. Based on similarities between the author concept profile and concepts for documents in the collection, additional papers are recommended to the author. Our work differs in creating user profiles which is based on using past documents viewed rather than on authored papers, extending the recommendations to CiteSeer^x users as well as authors.

3 APPROACH

The conceptual recommender system creates a user profile for each of the users visiting our CiteSeer^x mirror and makes use of this user profile to identify the interests of the user. To train the concept classifier, all documents in the collection were parsed and those documents that contain standard ACM tags were considered as training documents. Remaining documents in the collection were categorized into a predefined set of concepts according to the ACM's Computing Classification system taxonomy [12] using the model created from the training set. The ACM's taxonomy is 3 levels deep with 368 concepts. Thus, each document in the CiteSeer^x collection has an author-assigned or system-assigned set of concepts. Next, we built a login system for our mirror. This login system has three functions.

1. Allow a registered user to access CiteSeer^x using his login details.
2. Allow a non-registered user to register with CiteSeer^x.
3. Allow non-registered users who do not want to login to access CiteSeer^x without affecting our experiment.

We keep track of the documents visited by registered users for a particular query entered and then make use of this data to create a user profile for the user. The user profile contains the ACM concepts the user is interested in ranked in descending order. This is a dynamic user profile that is updated each time the user visits a new document. The concepts in the user profile are then used by the recommender system to recommend documents for the user.

3.1 System Architecture

Figure 1 shows an architectural diagram for our Conceptual Recommender system for CiteSeer^x.

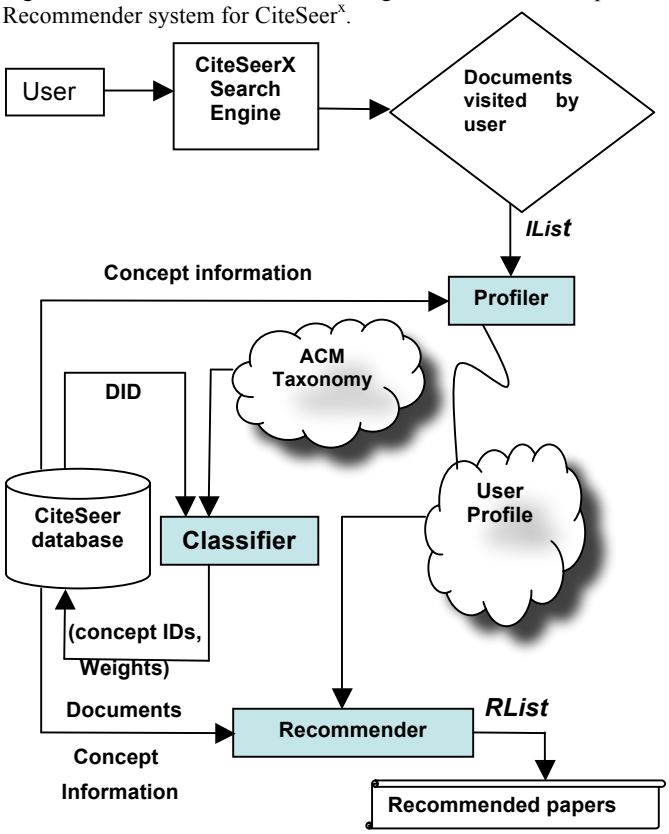

Figure 1: System Architecture

The system consists of three modules:

1. Classifier
2. Profiler
3. Recommender

Each of these modules is explained in the following sections.

3.2 Classifier

To begin with, all the documents in the CiteSeer database are classified into a set of predefined concepts in the ACM's Computing Classification System (CCS). This 3-level deep hierarchical set of concepts contains 369 total concepts. The classification is done in two stages.

1. Training stage: We parsed through the 1164939 text documents in the collection and found 31121 documents with author-assigned ACM tags. These documents were used as training data for a kNN classifier. For each concept in the CCS, we randomly selected 10 documents tagged by authors as belonging to the concept. 101 concepts had fewer than 10 candidate documents so they were ignored by the training algorithm, leaving us with a classifier trained on 268 total concepts.
2. Classification stage: The non-tagged documents were then classified using the kNN classifier trained as described above. The top 3 concept matches (concept identifier and match weight) returned by the classifier for each document in the collection were stored in a database for use by the conceptual recommendation system.

3.3 Profiler

The main objective of a profiler module is to create a user profile for the users of CiteSeer^x for whom we are trying to recommend papers. We track the documents visited by a particular user by storing the click track of the documents visited by the user. This visit history is input to the profiler module as a sequence of document IDs. We now retrieve the top three concepts or concepts and their corresponding weights *wt* for each of the documents visited by the user. Here, wt refers to the degree of association between a document and an associated concept as calculated by the categorizer.

We then create the user profile using the concept and the weight information. The concept and weight pairs are initially sorted according to the concepts. If we have more than one instance of the same concept with different weights, these weights are added to compute the final weight associated with that particular concept in the user profile.

$$wt(cp, j) = \sum wt(cp, i), \text{ for all documents } i \text{ for user } j$$

where:

wt (cp, j) = weight of concept cp in user profile j
wt (cp, i) = weight of concept cp in document i

Thus the final output of the profiler module is a list of ACM concepts and their corresponding weights the user might be interested in arranged in decreasing order.

Let us now consider an example to explain the profiler module. Suppose that we have 2 documents that the user has visited, D1 and D2. The top three concepts and their corresponding weights for each document are retrieved from the database. Let ((A, 0.1), (B, 0.2), (C, 0.3)) and ((D, 0.1), (E, 0.4), (A, 0.5)) be the set of (concept, wt) pairs for the documents D1 and D2 respectively.. The weights for the concepts associated with the user's visited documents are accumulated to produce a profile. For example, the profile constructed from D1 and D2 would contain, in decreasing order of concept weight:

A 0.6 = (0.1 + 0.5)
E 0.4
C 0.3
B 0.2
D 0.1

Thus, the user profile or a list of concept-weight pairs, where the weights represent the amount of interest a user might have in a particular concept.

3.3 Recommender

The recommender module uses the output of the user profile as the input. The final output of a recommender module is a set of recommended papers for a user. For each concept *cp* in the user profile, the recommender module retrieves the documents in CiteSeer^x that have concept *cp* as one of their top three concepts, as determined by the classifier. These documents are added to the list of possible recommendations. The weight of that document is calculated as weight associated with the concept *cp* in the user profile multiplied by the weight associated with the concept *cp* for the document. So the final weight of the document would be

$$wt(i, j) = wt(cp, j) * wt(cp, i)$$

where

wt (i, j) = weight of document i for user j
wt (cp, j) = weight of concept cp in user profile j
wt (cp, i) = weight of concept cp in document i

4 EXPERIMENTAL EVALUATION

We conduct several experiments in this section to evaluate the accuracy of our method. Additionally, we varied the number of concepts considered to determine the optimal number that removes ambiguity while avoiding spurious or irrelevant concepts.

4.1 Subjects and Dataset

The dataset for our study is a subset of the CiteSeer^x collection. CiteSeer^x is a search engine and digital repository of scientific and academic papers with major focus in the field of computer and information science. The collection contains over 5,000,000 documents. Our experiments were conducted over a subset of 1,000,000 documents from the collection.

We conducted a user study to determine the quality of our recommendations. Our study included 7 volunteer professors and graduate students. Each user entered 3 queries and we tracked the first 10 documents visited by each. These documents are then input to the profiler module. The output of the profiler module, a user profile in the form of a set of weighted concepts, is the input to the recommender module.

4.2 Evaluation Method

We evaluated the recommender's accuracy as the av it produced based while varying the number of concepts extracted from the user profile. We evaluated the recommendations produced from the top weighted 3, 6, 9, and 12 concepts contained in each user profile. For each method, we collected the top five recommended documents. The documents from all methods were then merged

and randomized and presented to the user for their feedback. Users were asked to judge them as very relevant (2), relevant (1), or irrelevant (0).

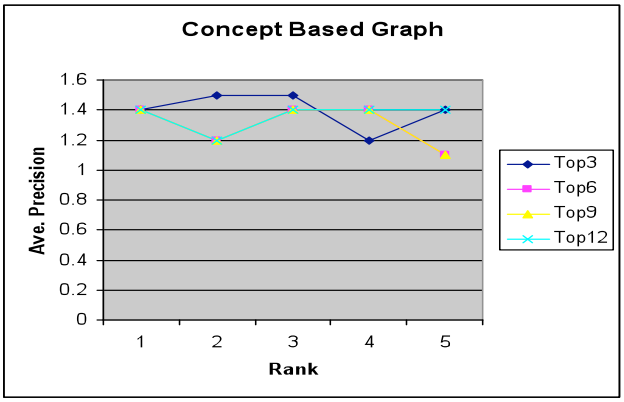

Figure 2: Effect of number of profile concepts on precision.

Figure 2 shows the results obtained by collecting user feedback and averaging it over the users in the study. The x axis shows the rank order of the recommendation and the y axis the average user judgment for documents at that rank. Looking only at the top recommendation (rank 1), all methods were equivalent with an average user judgment of 1.4 (midway between relevant and very relevant). However, the second and third ranked recommendations are higher when only the top 3 concepts from the user profile are used as the basis for the recommendations. Overall, these results indicate that recommending documents based on the top 3 concepts from the user profiles performed the best, compared with the top 6, 9, or 12 concepts.

5 CONCLUSION AND FUTURE WORK

In this paper, we describe a novel way of recommending documents to users of CiteSeer^x. We used the ACM classification tree and the documents visited by the user to create user profiles. In order to evaluate our method, we included 7 subjects in a user study including the professors and graduate students in Computer Science & Computer Engineering at the University of Arkansas. User profiles were created for these users and recommendations were made from the CiteSeer^x collection. We draw the following conclusions from this initial experiment:

1) Our concept-based method provides accurate recommendations without requiring extensive user histories or explicit ratings.

2) The method provides accurate recommendations even if only top 3 concepts were chosen from the user profile.

Currently, we are working to compare our method with traditional keyword-based recommenders and also conducting experiments to determine the ideal number of concepts and number of user clicks needed to get accurate recommendations. We are also integrating our system with the core CiteSeer^x project.

6 ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation grant number 0454121: CRI: Collaborative: Next Generation CiteSeer.

7 REFERENCES

[1] Adomavicius, G., and Tuzhilin, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. On Knowledge and Data Engineering 17 (6), pp. 734-749. June (2005).

[2] Balabanovic, M., and Shoham, Y. 1997. Fab: Content-Based, Collaborative Recommendation. Comm. ACM, vol. 40, no. 3, pp. 66-72, 1997.

[3] Basu, C., Hirsh, H., Cohen, W., and Nevill-Manning, C. 2001. Technical paper recommendation: a study in combining multiple information sources. Journal of Artificial Intelligence Research, (14) pp. 231-252. (2001).

[4] Bollacker, K., Lawrence, S., and Giles, C.L. 1998. Citeseer: an autonomous web agent for automatic retrieval and identification of interesting publications. Agents'98, 2nd International ACM Conference On Autonomous Agents, pp. 116-123 (1998).

[5] Burke, R. 2007. Hybrid Web Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007), pp.377-408.

[6] Chandrasekan, K., Gauch, S., Lakkaraju, P., and Luong, P-H. 2008. Concept-Based Document Recommendations for CiteSeer Authors. The 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2008), Hannover, Germany, July 28 - August 1, 2008, pp. 83-92.

[7] Councill, I.G., Giles, C.L., Iorio, E.D., Gori, M., Maggini, M., and Pucci, A. 2006. Towards Next Generation CiteSeer: A Flexible Architecture for Digital Library Deployment. Research and Advanced Technology for Digital Libraries, 10th European Conference, (ECDL 2006): 111-122, 2006.

[8] Gauch, S., Speretta, M., Chandramouli, A., and Micarelli, A. 2007. User Profiles for Personalized Information Access. Brusilovsky P., Kobsa A., and Nejdl W.(Eds.): The adaptive web, LNCS 4321, pp. 54 – 89. (2007).

[9] Geisler, G., McArthur, D., and Giersch, S. 2001. Developing recommendation services for a digital library with uncertain and changing data. In Proceedings of the 1st ACM/IEEE-CS. JCDL '01. ACM Press, pp. 199-200. New York, (2001).

[10] Pennock, D.M., Horvitz, E., Lawrence, S., and Giles, C.L. 2000. Collaborative Filtering by Personality Diagnosis: A Hybird Memory-and-Model-Based Approach. In Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-2000), pp. 473-480, Morgan Kaufmann, San Francisco, 2000.

[11] Salton, G., and Buckley, C. 1988. Term weighting approaches in Automatic Text Retrieval. Information Processing and Management, 24(5), pp. 513-523 (1988).

[12] The ACM Computing Classification System, http://acm.org/class/1998/