Enhancing Digital Libraries with TechLens$^+$

Roberto Torres[1,2], Sean M. McNee[2], Mara Abel[1], Joseph A. Konstan[2], John Riedl[2]

Grupo de Banco de Dados Inteligentes[1]
Universidade Federal do Rio Grande do Sul
Instituto de Informática
Porto Alegre, RS - Brazil
{rtorres, marabel}@inf.ufrgs.br

GroupLens Research[2]
Dept. of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455 - USA
{mcnee, konstan, riedl}@cs.umn.edu

ABSTRACT

The number of research papers available is growing at a staggering rate. Researchers need tools to help them find the papers they should read among all the papers published each year. In this paper, we present and experiment with hybrid recommender algorithms that combine Collaborative Filtering and Content-based Filtering to recommend research papers to users. Our hybrid algorithms combine the strengths of each filtering approach to address their individual weaknesses. We evaluated our algorithms through offline experiments on a database of 102,000 research papers, and through an online experiment with 110 users. For both experiments we used a dataset created from the CiteSeer repository of computer science research papers. We developed separate English and Portuguese versions of the interface and specifically recruited American and Brazilian users to test for cross-cultural effects. Our results show that users value paper recommendations, that the hybrid algorithms can be successfully combined, that different algorithms are more suitable for recommending different kinds of papers, and that users with different levels of experience perceive recommendations differently. These results can be applied to develop recommender systems for other types of digital libraries.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *information filtering, retrieval models, search process.*
H.3.7 [**Information Storage and Retrieval**]: Digital Libraries – *systems issues, user issues.*

General Terms

Algorithms, Experimentation, Human Factors

Keywords

Collaborative Filtering, Content-Based Filtering, Digital Libraries, Hybrid Recommender Systems

1. INTRODUCTION

According to the United States' National Science Foundation, more than half a million research papers were published in more than 1,900 journals worldwide in 1999. Since 1986, the number of papers published each year has increased at a rate of 1% per year [18]. If this trend continues, more than 10 million papers will be published in the next 20 years. Researchers find selecting the papers they should read difficult from among this growing abundance. Even within fairly narrow areas of interest such as data mining and digital libraries, it is still impossible to cope with all of the paper published each year. The problem is even more challenging for interdisciplinary research, since the resulting papers are published in a wide variety of venues.

One solution is search engines like Google, which make it easy to find papers by author, title, or keyword. However, in order to find a paper with a search engine the researcher has to know or guess appropriate search keywords. How can users discover the existence of previously unknown papers that they would find valuable?

Our previous research showed that collaborative filtering-based recommender systems successfully recommended research papers to users [16]. Here we introduce a new experimental system, called TechLens$^+$, to explore how two well-known techniques of recommender systems, namely collaborative filtering and content-based filtering can help users sort through the abundance of available research and find papers of interest. The TechLens$^+$ algorithms explore both the content of the paper and the social context of the paper, through the analysis of its citations to other papers. We believe that the algorithms we developed can be a significant aid in both current and emergent digital libraries.

We carry out this research in the context of CiteSeer, a well-known public repository of computer science research papers [3]. CiteSeer works by crawling the Web looking for research papers in a wide variety of formats, and parsing the research papers to determine the title, authors and other information. CiteSeer uses a process called "automatic citation indexing" to automatically build a citation graph among the papers it discovers.

CiteSeer is a digital library in which the content is not created, selected, or edited by professional staff, but by automatic algorithms. We call a digital library of this type an *emergent digital library*. Emergent digital libraries are similar to other digital libraries in many respects: they have a well-defined corpus of material; that corpus is digital in nature, and hence may be distributed by digital means; and they have powerful cataloging and search technologies appropriate to their content. However, emergent digital libraries have a key difference: the quality of their content varies more widely than the quality of content in most digital libraries, because professional staff is not involved in collection and appraisal. Therefore, exploration and search techniques are needed that can seek quality and relevance of results beyond what keyword similarity can provide. Our research seeks to explore such techniques.

JCDL'04, June 7–11, 2004, Tucson, Arizona, USA.
Copyright 2004 ACM 1-58113-832-6/04/0006…$5.00.

1.1 Contributions

This research presents unique approaches to recommend research papers. In addition to our results, we provide two key contributions to the fields of recommender systems and digital libraries.

First, we provide a set of algorithms that can be used for recommending research papers in many scientific domains. The algorithms were tested through both offline and online experiments and can be easily incorporated into existing digital libraries.

Second, we provide an online experimental evaluation to assess users' perceptions about paper recommendations. The experiment surveyed users across many dimensions and our results be used by other research projects.

The outline of this paper is as follows. We first discuss the related work both in digital libraries and recommender systems. We then talk about how to combine collaborative and content-based filtering in the domain of research papers. Next, we explain our algorithms and how they were tested in our experiments. Finally, we discuss the results and draw some conclusions.

2. RELATED WORK

2.1 Collaborative Filtering

Collaborative Filtering (CF) is one of the most successful techniques used in recommender systems. It has been used to recommend Usenet news [21], audio CDs [23], and research papers [16], among others. CF works by recommending items to people based on what other similar people have previously liked. CF creates neighborhoods of "similar" users (neighbors) for each user in the system and recommends an item to one user if her neighbors have rated it highly. CF is "domain independent" in that it performs no content analysis of the items in the domain. Rather, it relies on user opinions about the items to generate recommendations.

Despite being a successful technique in many domains, CF has its share of shortcomings [2]:

(i) *First-rater problem*: items need to be rated by at least one neighbor to be recommended, so the item cannot be recommended until someone rates it first.

(ii) *Sparsity problem*: in many domains, a user is likely to rate only a very small percentage of the available items. This can make it difficult to find agreement among individuals, since they may have little overlap in the set of items they've rated.

2.2 Content-Based Filtering

Content-Based Filtering (CBF) is also commonly used in recommender systems. Applied mostly in textual domains, such as news [11], CBF recommends items to a user if these items are similar in content to items the user has liked in the past. Many algorithms, most notably TF-IDF [22], have been used in CBF systems. CBF has many strengths, including the ability to generate recommendations over all items in the domain. CBF also has its shortcomings [2]:

(i) *Content limitation in domains*: in non-textual domains like movies and audio, current algorithms can't successfully and reliably analyze item contents.

(ii) *Analysis of quality and taste*: subjective aspects of the item, such as style and quality of writing or authoritativeness of the author are hard to analyze.

(iii) *Narrow content analysis*: CBF recommends items similar in content to the items rated in the past, and cannot produce recommendations for items that may have different but related content.

3. COMBINING CF and CBF

Taking a closer look at the characteristics of both CF and CBF, we can see that they are complementary. For example, CBF does not suffer from the first-rater problem as long as the content of a new item can be compared against all existing user profiles. In addition, CBF also does not suffer from sparsity, since every item in the systems can be related to every user profile. On the other hand, CF does not suffer from content-dependency, since it can be applied to every domain in which humans can evaluate items. Also, CF uses quality and taste when recommending items. Finally, the serendipitous nature of CF guarantees that there is no over-specialization problem.

In our previous work, we explored CF recommenders in the domain of research papers [16]. We were able to generate recommendations by mapping the web of citations between papers into the CF user-item ratings matrix. In our mapping, a 'user' in this matrix is a paper and an 'item' is a citation. Thus, every paper 'votes' for the citations it references. This mapping does not suffer from the new-user problem, common to most techniques, because each 'user' always provides ratings in the form of citations.

Further, in previous work we found that different CF algorithms generated *qualitatively* different recommendations: for example, some recommendations were more novel than others. CBF was only used as a baseline comparison against the CF, and no hybrid recommender approaches were considered. This paper builds upon our previous work by exploring how to combine CF and CBF to generate recommendations for research papers. We propose a set of new hybrid algorithms that combine a TF-IDF CBF algorithm [22], with a k-nearest neighbor (User-User) CF algorithm from our previous work.

We hypothesize that CF and CBF can be successfully combined to produce recommendations of research papers. In line with results from our previous work, we also hypothesize that different hybrid algorithms might be more suitable for recommending different kinds of papers. Finally, we hypothesize that users with different levels of experience perceive recommendations differently due to their own background, needs, and expectations.

Many hybrid recommender systems have been successfully built in the past. P-Tango recommended news items by combining recommendations from CBF and CF recommenders together using a weighted-average function [6]. Fab recommended Web pages by choosing neighbors for CF-based recommendations using CBF-based user profiles [2]. A 'boosted' combination of CF and CBF was proposed in [17], where CBF calculations were used to augment the CF ratings matrix. PTV recommended TV Guides using CF and CBF in parallel [7]. Finally, Woodruff developed six hybrid recommender algorithms that combined textual and citation information in order to recommend the next paper a user should read from within a single digital book [24]. In our approach, we developed a set of hybrid recommender algorithms and compared their performance to non-hybrid baseline recommenders over a corpus of computer science research papers in both online and offline experiments.

Our algorithms follow the taxonomy of hybrid recommender algorithms first proposed by Burke [5]. In particular, we implemented algorithms that follow the two most straightforward and appealing of Burke's categories: "feature augmentation" and "mixed" recommenders. In both categories, the hybrid recommender is composed of two standalone non-hybrid recommender algorithms. In 'feature augmentation' hybrid recommenders, the results generated from the first algorithm are fed as inputs to the second algorithm. In 'mixed' hybrid recommenders, the two algorithms are run independently with the same input and their results are merged together.

3.1 Recommender Algorithms

Before describing our recommender algorithms, we have to be precise with our language. We will draw a subtle but important difference between a paper and a citation. A citation is a paper for which the text may not be available. A citation therefore is a pointer to a paper. On the other hand, a paper is a citation for which we also have its text. This is important because many citations may be references to papers that we do not have in digital format.

We developed ten recommender algorithms; each algorithm receives a set of citations as input and generates an ordered list of citations as recommendations. Unless stated otherwise, the input set of citations is generated from the reference list of one input paper ('active' paper).

All algorithms make use of standalone CF and CBF recommendation engines. For our CF engine, we used the 'user-based' CF algorithm from the Suggest library, an implementation of standard CF algorithms [12]. For our CBF engine, we used TF-IDF from the Bow Toolkit, a library that performs document retrieval [15].

3.2 Non-hybrid Algorithms

These algorithms run either CF or CBF. They are used as baseline comparison for the hybrid algorithms.

3.2.1 *Only Collaborative Filtering*

We developed two CF-only algorithms: *Pure-CF* and *Denser-CF*. Pure-CF is the standard k-nearest neighbor CF algorithm [9]. It takes the citations of the active paper as input and gives a list of recommended citations as output. Denser-CF augments the input list of citations by adding the citations in the papers cited by the active paper to the input list.

3.2.2 *Only Content-Based Filtering*

Three content-based algorithms were built: *Pure-CBF*, *CBF-Separated* and *CBF-Combined*. They all are based on TF-IDF and uses Porter's stemming and stopword elimination [15]. All content analysis was performed on paper titles and abstracts.

Pure-CBF searches for similar papers based on TF-IDF similarity and recommends the most similar papers. CBF-Separated is an extension of Pure-CBF and it explores not only the text of the paper but also the text of the papers it cites. For instance, for the paper P the algorithm generates a list of similar papers L_P, and for every citation (C_1, C_2, …, C_n) of the paper P, it generates a list of similar papers (L_{C1}, L_{C2}, …, L_{Cn}). All lists are merged into one single list, sorted based on the returned similarity coefficient. Papers are discarded if they have already been added to the resulting list. Papers with the highest similarity scores are recommended. Recommending papers similar to the citations of a given paper should broaden the search space, leading to more diverse recommendations.

Finally, CBF-Combined is an extension of CBF-Separated. Instead of generating one list of similar papers for every citation, this algorithm merges the text of the paper and the text of all of the papers it cites together into one large chunk of text. This larger text is submitted as the query to search for similar papers. The most similar papers are then recommended. In this algorithm, the presence of more words in a single query to the CBF engine should more effectively return similar papers based on content.

3.3 Hybrid Algorithms

Each hybrid algorithm is composed of two independent modules: A CF algorithm and a CBF algorithm. Each module is responsible for getting an input and generating recommendations. The CBF module uses the text of the active paper as input and the CF module uses the citations from the active paper as input. Hybrid algorithms following Burke's feature augmentation model run these modules in sequence, with either CBF (CBF-Separated or CBF-Combined) or CF (Pure-CF) first. The output of the first module (up to 20 papers) is used as input to the second. In contrast, hybrid algorithms following Burke's mixed model runs the two modules in parallel and merges the recommendations together into a final recommended list.

3.3.1 *CF – CBF Separated*

In this algorithm, recommendations from Pure-CF are used as input to CBF-Separated. For every recommendation from CF, the CBF module recommends a set of similar papers (up to 80). Because the recommendations generated by the CF module are ordered, the recommendations generated by the CBF module have to be scaled by this ordering. Thus, these CBF recommendations are weighted, with the first set (generated from the top CF recommendation) receiving weight 1 and the following sets' weights decreased accordingly. The similarity scores of the CBF recommendations are multiplied by these weights and sorted.

3.3.2 *CF – CBF Combined*

This algorithm is similar to CF-CBF Separated. However, instead of recommending a set of similar papers for every recommendation received from the CF module, the CBF module aggregates the text of all of the recommendations given by CF and uses this large chunk of text as its input to CBF. The results are sorted by similarity.

3.3.3 *CBF Separated – CF*

Here, CBF-Separated generates recommendations for the active paper. These recommendations are used to augment the active paper's set of citations. The active paper with its modified set of citations is used as input to Pure-CF to generate recommendations.

3.3.4 *CBF Combined – CF*

This algorithm is identical to CBF Separated – CF, except that CBF-Combined is used in place of CBF-Separated.

3.3.5 *Fusion*

Fusion, our 'mixed' hybrid algorithm, runs the two recommender modules in parallel and generates a final recommendation list by merging the results together. The generation of the final recommendation list is as follows: every recommendation that is present in both modules' result lists is added to the final list with a rank score. This score is the summation of the ranks of the recommendation in their original lists. The final recommendation list is sorted based on these scores. Therefore, a paper that was ranked 3rd from the CF module and 2nd from the CBF module would receive a score of 5. The lower the score, the closer to the top an item goes. Recommendations that don't appear on both lists are appended, alternately, to the end of the final list. Fusion recommendation process is shown in Figure 1. A similar algorithm has been developed by Cotter [7].

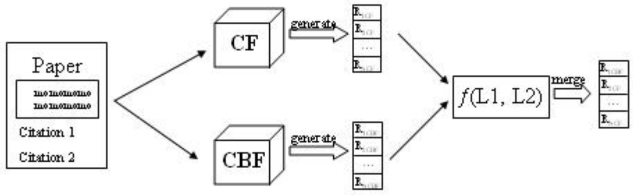


Figure 1. Fusion Algorithm

4. EXPERIMENTS

To test the utility of our algorithms, we ran two experiments: an offline experiment to evaluate the ability of the algorithms to recommend papers known to be related to a selected paper and an online experiment to evaluate users' perceptions about the quality of the recommendations.

4.1 Dataset

To test our algorithms we created a dataset with papers extracted from CiteSeer [3]. This dataset initially had over 500,000 papers and 2 million citations. We limited this dataset in two ways. First, we removed papers that cited fewer than 3 other papers, as we believe these loosely connected papers introduced noise to the dataset. Second, we removed citations for which we did not have the full text of the paper in our dataset. We performed this trimming so that both CF and CBF would be able to analyze every item in out dataset. The pruned dataset has 102,295 papers with an average of 14 connections per paper, where the number of connections is the number of citations a paper makes plus the number of papers that cite it.

4.2 User Profiling

In order to recommend papers to users we need to have a model of the user's interests; we need a user profile. This profile represents the user's tastes and opinions about the papers that she has read. Such a profile could contain both long-term and short-term user interests and the profile could gather data either explicitly or implicitly. See Table 1 for a listing of the advantages and disadvantages for creating profiles in these different ways.

An example of gathering information implicitly for long-term interests is to build the profile from all of the papers that the user has read in the past. This can be accomplished by building a system to monitor what papers the user downloads and reads, and silently incorporating all papers into the user's profile. This approach has the benefit of knowing everything a user reads and when the user read it, but it cannot know how closely the user read it. Thus, over time the user's profile could become bloated with papers that the user may have quickly skimmed and discarded. These false positives can erode the user's profile and reduce recommendation quality.

A way to combat this problem is to have the user explicitly state which papers are to be added to the user profile. Moreover, the user could also provide extra information such as a rating, commentary, or classification of the paper (e.g. by research area, field, etc.). The user could also review her profile to make sure that it accurately represented her. This explicitly gathered information would help the system generate personalized recommendations. It is unlikely, however, that a user would be willing to invest the time and energy needed to maintain such a user profile.

An example of gathering information implicitly for short-term interests is to build a paper-monitoring system similar to the ones previously described, with one large difference: this system would only remember the last paper a user downloaded and read. This one paper would be used as the current user profile and would be the basis for generating recommendations.

Finally, we could gather information explicitly for short-term interests. In such a system, the user would choose one paper to be his short-term profile; the recommender would use this paper to generate recommendations. We will use this approach in this research as it is not only the most straightforward and simple to understand, but it is also the only approach that does not require creating and installing a system on users' computers to monitor the papers they are downloading and reading.

Table 1: User Profile Alternatives

Approach	Gathering Information / Users' Interests	Information Used	Advantages	Disadvantages
All Papers	Implicit and Long-term interests	All papers read in the past	Keep track of user's reading habits over time	Privacy issues, bloated profiles, requires monitoring system
All Papers by Field	Explicit and Long-term interests	All papers read in the past	Filter recommendations based on user's field interests	Requires monitoring system, user must manually adjust profile
One Paper	Explicit and Short-term interests	Only one paper of interest	Does not require a system	Does not keep track of reading habits over time
One Paper	Implicit and Short-term interests	Only one paper of interest	Requires limited system	Privacy issues, does not track habits over time

5. OFFLINE EXPERIMENT

In this experiment, we randomly removed one citation from the active paper and then checked whether our algorithms could recommend that removed citation. This "leave one out" methodology has been frequently used in other recommender systems offline experiments [4, 16].

We divided the dataset into training and test datasets at a 90% to 10% ratio. Ten different training and testing datasets were created for 10-fold cross validation. For each trial, every paper in the test dataset had one randomly removed citation.

Although being successfully used in other research, this method of experimentation has some limitations. The recommender algorithms could recommend a paper that didn't exist at the time the active paper was published. To handle that, we filtered out recommendations with a publication year later than that of the active paper. The algorithms could also recommend papers that are very similar to or even better than the removed citation, possibly "diminishing" the algorithms' performance. Although this is a possibility, we still expect the removed citation to be recommended.

5.1 Metrics

There are two metrics used in the offline experiment: hit percentage and rank. We define "hit-percentage" as the percentage of the time the recommender algorithm correctly recommended the removed citation anywhere in the recommendation list. Similarly, we define "rank" as the location where the removed citation was found in the recommendation list.

We chose these metrics because of how the reflect real world use of recommenders: Users implicitly trust recommenders to generate meaningful, correct recommendations and that the order of recommended items is important. So, we validated our algorithms by first examining if they can generate the 'correct' recommendation (hit-percentage metric) and then how well they generate that recommendation (rank metric). Since both measures are important, we combined them.

We segmented our hit-percentage analysis into bins based on rank, where lower is better. Thus, a recommendation in the top-10 bin is better than a recommendation in the top-30 bin. For example, if an algorithm had a top-10 hit percentage of 25%, then 25% of the time that algorithm recommended the removed citation at a rank of 10 or better (i.e. lower). Recommendations beyond the 40th position are considered "all" because users are not likely to see items recommended beyond this position. The "all" bin is equivalent to the overall hit percentage of the algorithm.

To select the best algorithms, we focused on two particular criteria. As we think that users prefer to see the best recommendations first, our first criterion is the algorithm's performance in the top-10 hit-percentage. The second criterion is the algorithm's ability to recommend the removed citation independently of its rank. It is measured by the "all" hit-percentage. As a possible tie-breaker, the algorithm's top-1 performance is also examined.

5.2 Offline Results

Based on the above criteria, the best hybrid algorithms were Fusion, CBF Combined–CF, and CF-CBF Separated. The best non-hybrid algorithms were CBF-Separated and Pure-CF.

CBF-Separated was expected to perform better than Pure-CBF because it widens the search space by using the text of the active paper's citations. However, Pure-CF was not expected to perform better than Denser-CF.

Fusion performed significantly better compared to every other algorithm at both top-1 (28%) and all hit-percentage (78%). Pure-CF also did well, usually having the best or second best performance among all bins. The results of the five best algorithms are shown in Figure 2.

The other hybrid algorithms did not perform well. CBF Separated–CF was slightly inferior to the CBF Combined–CF. On the other hand, CF–CBF Combined had a very poor performance compared to CF–CBF Separated—CF–CBF Combined had an "all" hit-percentage of 0.1%. More research is needed to explore what happened here.

5.3 Offline Discussion

The poor performance of Denser-CF was a surprising result because we thought that the denser input used in this algorithm should improve the quality of recommendations, since sparsity is a known CF problem. But, if for example, we assumed each paper had 7 citations, then Denser-CF would add 49 more citations to the input set, these additions might be not closely enough related to the active paper, and thus generate only noise.

To help us understand the behavior of the algorithms, we also looked at recommendation coverage. Coverage is the percentage of items for which the system could generate a recommendation [10]. All of the algorithms had 100% coverage, except Pure-CF which had coverage of 93%. This high coverage in the hybrid algorithms is due to the presence of the CBF as an algorithm step and it is a significant advantage of building a hybrid recommender.

The results from the offline experiment were used to select algorithms for the online experiment. We chose to use the following algorithms online: CBF-Separated and Pure-CF as non-hybrids, and CF–CBF Separated, CBF Combined–CF, and Fusion as the hybrid algorithms.

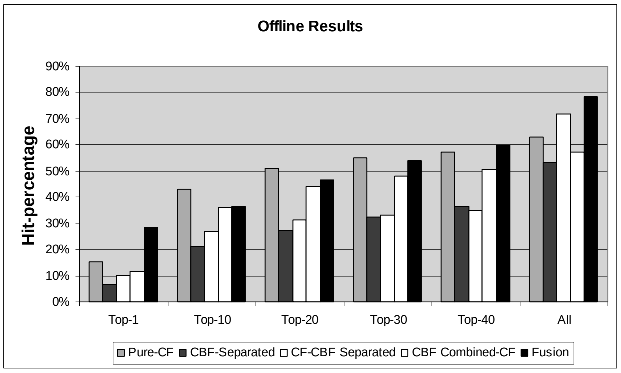


Figure 2: Results for the 5 best algorithms

6. ONLINE EXPERIMENT

The online experiment was aimed to assess users' perceptions about the recommendations they received.

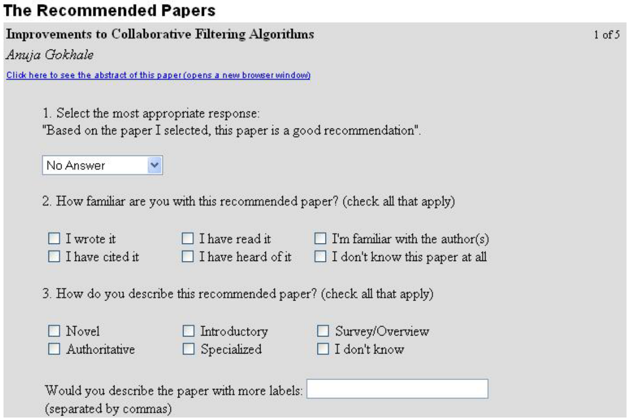
6.1 Experimental Design

We developed an online experimental system, called TechLens$^+$, consisting of a six-page Web-based experiment where users evaluated recommendations of research papers.

Users were invited to anonymously participate through links at the Penn State version of CiteSeer [20] and EBizSearch [8]. Users were also invited through messages posted in e-mail lists, such as internal lists of the Computer Science departments at:

the University of Minnesota, Universidade Federal do Rio Grande do Sul, Georgia Institute of Technology, and UC Berkeley. Additional e-mail invitations were sent to the UC Berkeley Collaborative Filtering Interest List[1], the User Modeling Interest List[2], and interest lists of the Brazilian Computer Society[3]. Users ranged from graduate students to professors and professional researchers. Because of the nature of the e-mail lists and websites we chose for recruiting users, we expected to have knowledgeable subjects all of whom would be able to complete the experiment.

There were six pages in the TechLens$^+$ system, an example from one is shown in Figure 3. When the user came to the website, she had to consent to participate in the experiment. After that, the user was randomly assigned to one algorithm. The user was asked for the name of an author whose work the user was familiar with. All of the papers with that author's name were retrieved and the user chose a paper to get recommendations for. This paper became the user's active paper. The system then generated five recommendations based on the active paper and asked the user questions about the recommendation she received. The user was able to read the title, author list, and abstract for each recommendation.



The Recommended Papers

Improvements to Collaborative Filtering Algorithms 1 of 5

Anuja Gokhale

Click here to see the abstract of this paper (opens a new browser window)

1. Select the most appropriate response:
"Based on the paper I selected, this paper is a good recommendation".

No Answer

2. How familiar are you with this recommended paper? (check all that apply)

□ I wrote it □ I have read it □ I'm familiar with the author(s)
□ I have cited it □ I have heard of it □ I don't know this paper at all

3. How do you describe this recommended paper? (check all that apply)

□ Novel □ Introductory □ Survey/Overview
□ Authoritative □ Specialized □ I don't know

Would you describe the paper with more labels:
(separated by commas)

Figure 3. TechLens$^+$ Experiment

To gather user opinion about the recommendation she received, the user was asked to answer the following three questions for each recommendation:

1. "Based on the paper I chose, this is a good recommendation", with the options of "no answer", "strongly agree", "agree", "maybe or unsure", "disagree", and "strongly disagree".
2. "How familiar are you with this recommendation?", with checkbox options of "I wrote it", "I have cited it", "I have read it", "I have heard of it", "I'm familiar with author(s)", and "I don't know this paper at all".
3. "How do you describe this recommended paper?", with checkbox options of "novel", "authoritative", "introductory", "specialized", "survey/overview", and "I don't know".

Question 1 aims to find out how each algorithm generates high quality recommendations. Question 2 aims to give support for the results found, checking how familiar users were with the recommendations they were evaluating. Question 3 is used classify papers into different 'classes' (e.g. authoritative, specialized, etc.) to find out which algorithms are better for recommending specific classes of papers.

After reviewing all of the recommendations, we asked a final set of questions to assess the overall quality of the recommendations generated:

4. "For what applications would you be interested in using a research paper recommender system like this one?", with checkbox options of "weekly/monthly newsletter", "finding related paper to a chosen paper", "finding citations for a current working paper", "find papers to an unfamiliar area", "finding reviewers for a paper", and "finding new papers that build upon previous research".
5. "Do you think that the overall set of recommendations was good?", with the options of "strongly agree", "agree", "maybe or unsure", "disagree", and "strongly disagree".
6. "Which of the following attributes a recommender system like this one should take into account when generating recommendations?", with checkbox options of "narrowing the search based on the year of the paper", "narrowing the search based on authors", "recommending papers from certain journals or conferences", and "recommending papers that were cited at least a certain number of times".
7. "How do you describe yourself?", with the options of "undergraduate student", "masters student", "PhD student", "researcher", "professor", and "professional". Professor and researcher had a field to enter their years of experience.

Question 4 aims to gather from users what kind of applications they would be interested in using. Question 5 aims to find out if one algorithm can generate a good *set* of recommendations. This differs from Question 1 because one user can consider a set of recommendations good even if it has only one good recommendation. This provides us with another measure to verify the ability of an algorithm to generate quality recommendations. Question 6 aims to gather which attributes and features might be valuable to take into account when integrating recommenders into digital libraries. Finally, Question 7 aims to give support to our results by gathering information about our users.

The questions 3, 4, and 6 had an empty textbox for entering other answers. Also, at the end of the experiment we provided room for any additional comments.

6.2 Online Results

During the 32-day experimental run, 110 subjects participated in the experiment: 33 from the United States, 43 from Brazil and 34 from other countries. On average, subjects spent 20 minutes answering our questions. We had 20 Masters students, 33 Ph.D. students, 27 researchers and 23 professors. Undergraduate and professionals represented 6 subjects and were not separately analyzed. The number of users per algorithm is shown in Table 2.

To evaluate the user's satisfaction with their recommendations, we categorized their answers. The options *strongly agree* and *agree* options are considered as "satisfied" and the *strongly disagree* and *disagree* are considered as "dissatisfied" about the recommendations. Non-committal answers (e.g. unsure) were ignored. Figure 4 shows user satisfaction for each algorithm. CBF-Separated, Fusion, and CBF Combined-CF scored higher

[1] http://www.sims.berkeley.edu/resources/collab/

[2] http://www.um.org/

[3] http://www.sbc.org.br/

than Pure-CF and CF-CBF Separated. Also, as Table 3 shows, subjects were satisfied both for each individual recommendation and overall.

Table 2: Users per Algorithm[4]

Algorithm	Number of users
CBF Separated	14
Pure-CF	28
CF-CBF Separated	25
CBF Combined – CF	18
Fusion	25

To evaluate the user's familiarity with the papers, we broke down our analysis into three groups: a user is considered *very familiar* if he cited or read the paper, *familiar* if he has heard about the paper or is familiar with the authors, and *unfamiliar* if he doesn't know the recommendations at all. Of all the recommendations, 27% were very familiar to the users, 34% were familiar, and 36% of the papers were unfamiliar. Only 2% of the recommendations received were written by the users who were evaluating them.

Recommendation satisfaction also varied by user type with 75% of the masters students, 61% of the PhD students, 67% of the researchers, and only 52% of the professors saying they were satisfied with their recommendations. Researchers and professors are considered *professionals* and masters and PhD students are considered *students*.

6.2.1 Paper Class Analysis

In Table 4, we review the best and worst algorithms for each class of paper. Pure-CF and Fusion are better than CF-CBF Separated for recommending novel and authoritative papers ($p < 0.05$).

For introductory papers, CBF-Separated and CF-CBF Separated are better than Pure-CF. Finally, CBF-Separated is better than Pure-CF and Pure-CF was worse than all of the other algorithms to recommend survey papers ($p < 0.1$).

Table 3: Users' Satisfaction with Recommendations

	Individual Recommendations	Overall Set of Recommendations
Satisfied	46%	62%
Dissatisfied	21%	19%

6.2.2 Cross-country Analysis

Approximately 2/3 of the users came either from the United States or Brazil. A user is considered from one of these countries based on where the user was physically located by IP address when he/she accessed the experiment. The breakdown of the subject population is shown in Table 5b.

Between countries, user satisfaction with individual recommendations is similar, with 50% satisfaction reported by the Americans and 49% reported by Brazilians. Dissatisfaction is similar too: Americans at 15% and Brazilians at 17%. On the other hand, satisfaction with the overall set of recommendations varied greatly. Americans were satisfied with 42% and not satisfied with 33% of the recommendations, while the Brazilians were satisfied with 70% and not satisfied with 12% of the recommendations.

There were also strong differences in familiarity. Americans were more familiar with the recommendations, with 31% very familiar, 41% familiar and 24% unfamiliar. Brazilians, on the other hand, were 24% very familiar, 31% familiar and 44% unfamiliar with the recommendations.

Thus, Americans and Brazilians have roughly equal satisfaction with individual recommendations. Brazilians are much more satisfied with whole set of recommendations than the Americans. Finally, Americans are more familiar with the recommendations they received when compared to the Brazilians.

Table 4: Recommended Algorithms by Paper Class

Class of Papers	Best Algorithms	Worst Algorithms	P Value
Novel	Pure-CF, Fusion	CBF-Sep.	< 0.05
Authoritative	Pure-CF, Fusion	CF-CBF Sep.	< 0.05
Introductory	CBF Sep., CF-CBF Sep.	Pure-CF	< 0.1
Survey/Overview	CBF-Sep.	Pure-CF	< 0.1

6.2.3 Cross-Language Analysis

The Portuguese version of TechLens$^+$ started 6 days after the English version. During this time, 12 Brazilian users participated in the English Version of the experiment. After the launch of the Portuguese version, Brazilian users preferred to participate in this version. We then divided the Brazilians into two groups: those that participated in the English and those that participated in the Portuguese version. This population distribution is shown in Table 5a.

Overall recommendation quality between the two language groups shows strong differences: Brazilians were satisfied with 42% and dissatisfied with 33% of the recommendations in the English version, while in Portuguese, they were satisfied with 81% and dissatisfied with only 3%.

These differences also carried over into familiarity. In English, Brazilians were 11% very familiar, 32% familiar and 57% unfamiliar with the recommendations they received. While in Portuguese, they were 29% very familiar, 31% familiar and 40% unfamiliar with the recommendations.

Table 5: Distribution of Users

Brazil Eng.	Brazil Port.	Type of User	Total Brazil	Total USA
3	12	Masters Students	15	4
3	7	PhD Students	10	13
0	6	Researchers	6	8
5	5	Professors	10	5

(a) (b)

[4] Users were randomly assigned to each algorithm. Only three users left the experiment after receiving recommendations.

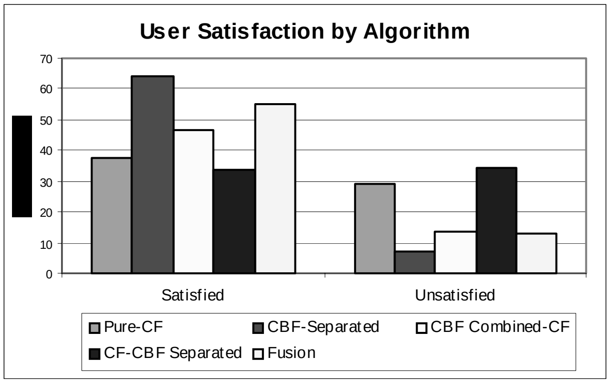


Figure 4: User Satisfaction by Algorithm

6.3 Online Experiment Discussion

Users that participated in our experiment were mostly masters and PhD students (*students)*, researchers and professors (*professionals)*. Professors were more experienced on average than researchers. This population provided highly valuable answers. Our analysis showed that the less experienced the user was, the more the user liked the recommendations. In addition, professors were more familiar with the papers (as expected), which might make them "less happy".

Our cross-country analysis showed that there are no strong cultural differences in receiving research paper recommendations. In addition, our analysis reinforced the results that level of experience influences user satisfaction. Brazilian users had a higher percentage of masters' students than American users had. Consequently, Brazilian users were less familiar with the papers than American users. Therefore, the recommendations given to Brazilian users made them more satisfied. These results suggest that research paper recommender systems should be tailored to the experience level of each user.

Our analysis also showed strong language differences. Brazilians in the Portuguese experiment were more satisfied with the recommendations they received. We hypothesize that because most of what is in the Internet is written in English, Brazilian users might be more satisfied being invited to participate in a Portuguese experiment. This suggests that research paper recommender systems interface should be localized to the user's native language, reducing the users' burden of finding good research papers. This is independent of the language of the papers, because most of them were written in English, and Brazilians were happy either way.

Finally, in order for a recommender system to add value to a digital library, it has to generate high quality recommendations consistently. Not every single recommendation has to be good, however. Users want a recommendation *set* that is of high quality. As we found both in this work and in our previous work, users are happy even if they receive only one or two good recommendations out of five.

For example, one user commented: "I was looking for papers that would help me writing a compiler without writing code generators for many different processors". This user considered only one recommendation as relevant. Although the user was looking for a very specific topic, the system was voted 'very useful' and the user considered the whole set of recommendations as 'good'. Overall, we found that, 85% of the users said they received at least one good recommendation. This encourages us that our recommender algorithms can be used in digital libraries.

7. CONCLUSIONS

In this paper we described, implemented and tested different techniques for combining content-based and collaborative filtering-based recommender algorithms for recommending research papers.

Returning to our hypotheses, we found that many of our CF-CBF hybrid recommender algorithms can generate research paper recommendations that users were very happy to receive. In addition, because 85% of our users received at least one good recommendation, we believe that our algorithms can aid digital libraries.

Some of the feature augmentation algorithms we tested, however, did not perform well. We believe this is due to the sequential nature of these hybrid algorithms: the second module is only able to make recommendations seeded by the results of the first module. In general, we believe sequential hybrid recommendation algorithms will not perform well because pure recommender algorithms are not designed to receive input from another recommender algorithms.

Our algorithms were tested using a dataset of computer science research papers. However, the algorithms are designed to be used in any domain, as long as the text and citations of the papers are available in digital format. Thus, we believe that most existing and emergent digital libraries, such as [1], [13] or [19], can successfully incorporate our hybrid algorithms. Of particular note is our Fusion algorithm, where any enhancement to each component technique (CF or CBF) can be promptly incorporated into the overall algorithm.

Our online results showed that different algorithms should be used for recommending different kinds of papers, reinforcing results found in our previous work. In addition, our results showed that users with different levels of experience perceive recommendations differently. For example, professionals were not as "happy" as students.

We have a vision for the future of a completely personalized or 'tailored' digital library. Such a digital library might tailor recommender algorithms for particular user tasks using Table 4 as a guide. For example, suppose that the task of "finding related work" could be solved by recommending novel and authoritative papers. Then a system that wanted to support this task should use Pure-CF and Fusion to generate paper recommendations. Second, the digital library might tailor itself to the user's native language, independent of the language of the papers. Finally, the digital library might tailor the recommendations it displays based on the level of experience a user has. More research is needed to understand in detail which approaches are best for which users.

7.1 Future Work

Our results were based on a user profile with explicit input of preferences from the users and for short-term interests. We believe that other user profiles should be tested in order to track evolving reading habits over time. Further studies with users of multiple nationalities would also be desirable and to determine why our feature augmentation algorithms did not perform well online.

Our algorithms were based on Burke's taxonomy of hybrid recommender algorithms [5]. In this work, we only implemented algorithms in two of his seven categories. It would be interesting to implement algorithms in all of his categories to compare them against each other. Moreover, Burke only classified hybrid algorithms that could be built from standalone recommender algorithms. Further studies comparing Burke's taxonomy with more tightly integrated hybrid algorithms would be worthwhile to perform.

It would also be interesting to investigate algorithm differences in recommending recent compared to older research papers. We believe this leads to the possibility of recommending "research paths" to users. Given a query of a research area and knowledge of what a user has already read, a recommender could generate a display of how this area has evolved over time and produce an ordered list of "must-read" papers in that field. We believe this is an important area to look into, not only for educational purposes, but because over 69% of our subjects said they would like recommender systems to help them find papers that built off of known research.

8. ACKNOWLEDGMENTS

We'd like to thank GroupLens Research, mainly Dan Cosley, Shyong (Tony) Lam, and Dan Frankowski. Also thanks to Lee Giles, Isaac Council, and Pradeep Teregowda for linking to our online experiment. Finally, thanks to NECI for the access to the CiteSeer database. This work was supported by grants from the NSF (DGE 95-54517, IIS 96-13960, IIS 97-34442 IIS 99-78717, and IIS 01-02229), and FINEP-CTPETRO (65.00.0286.00).

9. REFERENCES

1. ACM Digital Library, http://www.acm.org/dl, 2004.
2. Balabanovic, M. and Y. Shoham. Fab: Content-Based, Collaborative Recommendation. *Communications of the ACM*, 40(3), 1997, pp. 66-72.
3. Bollacker, K., S. Lawrence, and C.L. Giles. A System for Automatic Personalized Tracking of Scientific Literature on the Web. In *Proceedings of the Fourth ACM Conference on Digital Libraries (DL 99)*, Berkeley, CA, 1999, pp. 105-113.
4. Breese, J., D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI 98)*, Madison, WI, 1998, pp. 43-52.
5. Burke, R. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-adapted Interaction*, 12(4), 2002, pp. 331-370.
6. Claypool, M., A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining Content-Based and Collaborative Filters in an Online Newspaper. *ACM SIGIR '99 Workshop on Recommender Systems*, Berkeley, CA, 1999.
7. Cotter, P. and B. Smyth. PTV: Intelligent Personalised TV Guides. In *Proceedings of the Twelfth Innovative Applications of Artificial Intelligence Conference on Artificial Intelligence (IAAI-2000)*, Austin, TX, 2000, pp. 957-964.
8. eBizSearch, http://gunther.smeal.psu.edu/, 2004.
9. Google, http://www.google.com/, 2004.
10. Herlocker, J., J. Konstan, A. Borchers, and J. Riedl. An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the Twenty-Second Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*, Berkeley, CA, 1999, pp. 230-237.
11. Bharat, K., T. Kamba, and M. Albers. Personalized, Interactive News on the Web. *Multimedia Systems*, 6(5), 1998, pp. 249-358.
12. Karypis, G., SUGGEST Top-N Recommendation Engine, http://www.cs.umn.edu/~karypis/suggest/, 2000.
13. LANL (arXiv) e-Print Archive, http://arxiv.org/, 2004.
14. Lawrence, S., Access to Scientific literature. *The Nature Yearbook of Science and Technology*, 420(19), 2001, p. 86-88.
15. McCallum, A.K., Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering, http://www-2.cs.cmu.edu/~mccallum/ bow/, 1996.
16. McNee, S., I. Albert, D. Cosley, P. Gopalkrishnan, S.K. Lam, A.M. Rashid, J.A. Konstan, and J. Riedl. On the Recommending of Citations for Research Papers. In Proceedings of the ACM 2002 Conference on Computer Supported Cooperative Work (CSCW 2002), New Orleans, LA, 2002, pp. 116-125.
17. Melville, P., R.J. Mooney, and R. Nagarajan. Content-Boosted Collaborative Filtering. *ACM SIGIR 2001 Workshop on Recommender Systems*, New Orleans, LA, 2001.
18. NSF, *N.S.F. Academic Research and Development*. 1999.
19. New-Zealand Digital Library, http://www.sadl.uleth.ca/nz/cgi-bin/library, 2004.
20. Penn State University, CiteSeer.IST, http://citeseer.ist.psu.edu/, 2004.
21. Resnick, P., N. Iacovou, M. Sushak, P. Bergstrom, and J. Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In Proceedings of the ACM 1994 Conference on Computer Supported Collaborative Work (CSCW '94), Chapel Hill, NC, 1994, pp. 175-186.
22. Salton, G. and C. Buckley, Term weighting approaches in automatic text retrieval. I*nformation Processing and Management*, 24(5), 1988, p. 513-523.
23. Shardanand, U. and P. Maes. Social information Filtering: Algorithms for automating "word of mouth". In *Proceedings of the 1995 ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 95)*, Denver, CO, 1995, pp. 210-217.
24. Woodruff, A., R. Gossweiler, J. Pitkow, E.H. Chi, and S.K. Card. Enhancing a Digital Book with a Reading Recommender. In *Proceedings of the 2000 ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2000)*, The Hague, The Netherlands, 2000, pp. 153-160.