Meta-recommendation Systems: User-controlled Integration of Diverse Recommendations

J. Ben Schafer
Department of Computer Science
University of Northern Iowa
Cedar Falls, IA 50614-0507 USA
+1 319 273 2187
schafer@cs.uni.edu

Joseph A. Konstan
Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455USA
+1 612 625 4002
konstan@cs.umn.edu

John Riedl
Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455USA
+1 612 625 4002
riedl@cs.umn.edu

ABSTRACT
In a world where the number of choices can be overwhelming, recommender systems help users find and evaluate items of interest. They do so by connecting users with information regarding the content of recommended items or the opinions of other individuals. Such systems have become powerful tools in domains such as electronic commerce, digital libraries, and knowledge management. In this paper, we address such systems and introduce a new class of recommender system called meta-recommenders. Meta-recommenders provide users with personalized control over the generation of a single recommendation list formed from a combination of rich data using multiple information sources and recommendation techniques. We discuss experiments conducted to aid in the design of interfaces for a meta-recommender in the domain of movies. We demonstrate that meta-recommendations fill a gap in the current design of recommender systems. Finally, we consider the challenges of building real-world, usable meta-recommenders across a variety of domains.

Categories and Subject Descriptors
H.1.2 [**Models and Principals**] User/Machine Systems – *human factors, human information systems*. H.5.2 [**Information Interfaces and Presentation**] User Interfaces – *interaction styles, screen design, user-centered design*

General Terms
Algorithms, Design, Experimentation, Human Factors.

Keywords
Recommender systems, collaborative filtering, information filtering.

CIKM'02, November 4-9, 2002, McLean, Virginia, USA.

1. INTRODUCTION
Consider the following scenario. Mary's 8-year-old nephew is visiting for the weekend, and she would like to take him to the movies. Mary has several criteria for the movie that she will select. She would like a comedy or family movie rated no "higher" than PG-13. She would prefer that the movie contain no sex, violence or offensive language, last less than two hours and, if possible, show at a theater in her neighborhood. Finally, she would like to select a movie that she herself might enjoy.

Traditionally, Mary might decide which movie to see by checking the theater listings in the newspaper and asking friends for recommendations. More recently, her quest might include the use of the Internet to access online theater listings and search databases of movie reviews. Additionally, computer technology has provided collaborative filtering based recommendations – those based on the opinions of a community of like-minded individuals. However, using these sources requires a significant amount of manual intervention; Mary must visit each source to gather the data that will help her make a decision.

Similar situations can be found across a variety of domains. A consumer can use dozens of sources to gather a variety of attribute data and opinions regarding a product. Internet users browsing for a website on a given topic can try any number of search engines, each using a slightly different mechanism for determining the "top recommendations." Knowledge workers would like to combine a variety of techniques including keyword analysis, citation analysis, and the recommendations of other users to select appropriate documents.

In this paper, we introduce a new class of recommendation interface called meta-recommendation systems. These systems present recommendations fused from "recommendation data" from multiple information sources. In allowing users to provide both ephemeral and persistent information requirements, these systems produce recommendations through a unique blend of query-fit and recommendation data. Furthermore, these systems provide a high level of user control over the *combination* of recommendation data, providing users with more unified and meaningful recommendations.

In presenting meta-recommenders, we discuss the results of several focused user studies concerning the design of meta-recommendation interfaces for the domain of movies. Additionally, we present results that indicate that users prefer the

recommendations from these meta systems to more traditional recommender systems.

2. RELATED WORK

Recommender systems have emerged in both research and commercial applications that demonstrate that such systems are powerful tools for helping connect users with items of interest. Although designed to help users identify the items that best fit their tastes or needs, these systems vary greatly in regards to how they assist the user (what they do) and the technologies used to accomplish these goals (how they do it).

2.1 What Recommender Systems Do

According to Resnick and Varian [14], "in a typical recommender system people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients." This definition includes three classes of systems. **Suggestion systems** provide a list of candidate items or recommendations. **Estimation systems** provide an estimate of user preference on specific items or predictions. **Comment systems** provide access to textual recommendations of members of a community.

Schafer et al. [16] have extended this definition by using the term "recommender system" to refer not only to systems that specifically recommend items but also to those that help users evaluate items. Such systems include **feature-search systems**, which provide users with the ability to express explicitly an interest in items with a particular set of features. While the line between feature-search systems and keyword retrieval systems is a fine one, the distinction lies in the overall feel of the system. These "recommendations" serve as an important first step in the user's decision-making process. In this paper, we will use the term "recommender system" to refer to any system that provides a recommendation, prediction, opinion, or user-configured list of items that assists the user in evaluating items.

2.2 How Systems "Recommend"

Although the algorithms used within these systems vary, most are based on one or more of three classes of technology: data mining, information filtering and retrieval, and collaborative filtering.

The term **data mining** refers to a broad spectrum of mathematical modeling techniques and software tools that are used to find patterns in data. Recommender systems that incorporate data mining techniques make their recommendations using knowledge learned from the actions and attributes of users. These systems are often based on the development of user profiles that can be persistent (based on demographic or item "consumption" history data), ephemeral (based on the actions during the current session), or both. While recommender systems using data mining techniques are common in the domain of e-commerce [16], these techniques are not used in this research, and no exemplars are discussed.

The earliest "recommender systems" were **information filtering and retrieval** systems designed to fight information overload in textual domains. Recommender systems that incorporate information retrieval methods are frequently used to satisfy ephemeral information needs from relatively static databases. Conversely, recommender systems that incorporate information filtering (IF) methods are frequently used to identify items that match relatively stable and specific information needs in domains with a rapid turnover or frequent additions. Although information retrieval and information filtering are considered fundamentally different tasks [2], they are based on similar techniques. In this paper we will consider both under the singular term "information filtering."

Recommender systems employing information filtering techniques often do so through the use of IF agents. Operating in the domain of Usenet news, NewT [11] employs a vector-space based genetic algorithm to learn which articles should be selected and which should not. Ripper [5] and RE:Agent [3] use learning techniques to classify e-mail based on a user's prior actions. Finally, Amalthaea [12] is a multi-agent system for recommending information sources on the Internet. Information filtering agents keep track of a user's interests while information discovery agents search and retrieve documents matching the user's interest profile. Commercial applications of IF-based recommender systems include library and clipping services, such as webclipping.com, which use keyword searches of online newspaper, magazines, Usenet groups, and web pages to deliver recommended information to customers.

Collaborative filtering (CF) is an attempt to facilitate the process of "word of mouth." Users provide the system with evaluations of items that may be used to make "recommendations" to other users. The simplest of CF systems provide generalized recommendations by aggregating the evaluations of the community at large. More advanced systems personalize the process by forming an individualized neighborhood for each user consisting of a subset of users whose opinions are highly correlated with those of the original user.

Recommender systems based on collaborative filtering have produced recommendations in a variety of domains. Operating on email and Usenet news postings, Tapestry [6] allows users to identify other users whose knowledge should be trusted ("show all books on 'agents' in which Nathan's evaluation contains 'outstanding'"). These rules actively establish a neighborhood for recommendations. The original GroupLens project [14] provides automated neighborhoods for recommendations in Usenet news. Users rate articles, and GroupLens automatically recommends other articles to them. Similarly, Ringo [17] uses CF techniques to provide users with recommendations about audio CDs. In addition, Ringo has support for message boards (independent of the recommender system) on which users can discuss their music tastes. Finally, while the previous examples rely on explicit ratings, PHOAKS [18] uses implicit ratings to create a recommender system by examining Usenet news postings to find "endorsements" of web sites and creating a listing of the top web sites endorsed in each newsgroup. Commercial applications of CF-based recommender systems include e-commerce sites, such as Amazon.com, which use implicit recommendations via purchase history and/or explicit recommendations via "rate it" features to generate recommendations of products to purchase.

As researchers have studied different recommender system technologies, many have suggested that no single technology works for all situations. Thus, **hybrid systems** have been built in an attempt to use the strengths of one technology to offset the weaknesses of another. ProfBuilder [19] uses separate IF and CF algorithms on different data sources to generate a pair of recommendation lists. SmartPad [10], Digital Video [13], Fab [1], and Filterbots [7] use similar methods but extend this concept

by integrating an algorithm to merge the recommendation lists. Tango [4] and Krakatoa [9] provide users partial access to their information filters; users are given the ability to provide keywords of positive interest that can affect the type of documents returned from the information filter. Krakatoa even provides users access to the combination filter. Through the use of an on-screen slider, users may dynamically adjust the ratio in which the information and collaborative filters are combined. Commercial applications of hybrid-based recommender systems include search tools such as Google (www.google.com) that combine results of both content searches and collaborative recommendations.

3. META-RECOMMENDERS

The hybrid systems discussed in the previous section have made significant contributions to the field of recommender systems. However, they do not allow users to provide information that might improve the recommendations produced by the combination algorithm. For example, a user can't tell Google to weigh the currency of the web pages more highly in a search for "CIKM author instructions" even if currency may be part of the underlying algorithm. Thus, the user may be forced to process manually the recommendations in order to weed out the instructions from previous years.

By giving users access to the combination algorithm, such systems may provide more meaningful recommendations in situations where a user has ephemeral needs. Consider our original scenario. A movie recommendation system based on these hybrid systems should provide Mary with lists of movies she will like based on her long-standing collaborative filtering-based profile. However, by being given access to the combination algorithm, Mary can indicate that the system should make predictions biased less towards the British art films she frequently likes and more towards the family movies appropriate for her nephew, or that the movie should be relatively free of offensive language and last less than two hours.

In an effort to create a hybrid system with this level of user control, we have defined a new class of recommender system called meta-recommenders. These provide users with personalized control over the generation of a single recommendation list formed from a combination of rich data using multiple information sources and recommendation techniques.

Based on the lessons we learned from existing hybrid systems, we built MetaLens; a meta-recommender for the domain of movies. Much like Mary, who makes her final choice by examining several movie data sources, MetaLens uses IF and CF technologies to generate recommendation scores from several Internet film sites. In the remainder of this section we will briefly explain the architecture and the process used in MetaLens (Figure 1).

The user interface for MetaLens centers on two screens. On the preferences screen, users indicate their ephemeral requirements for their movie search. They do this by providing both the specific factors they consider important to a feature and an item-feature weight that indicates how important it is that the recommended movie matches these factors. As an example, Figure 2 might represent Mary's requirements for the movie that she views with her nephew. When Mary submits her preferences, the interface layer validates the information provided, formats it, and transfers control to the computation layer.

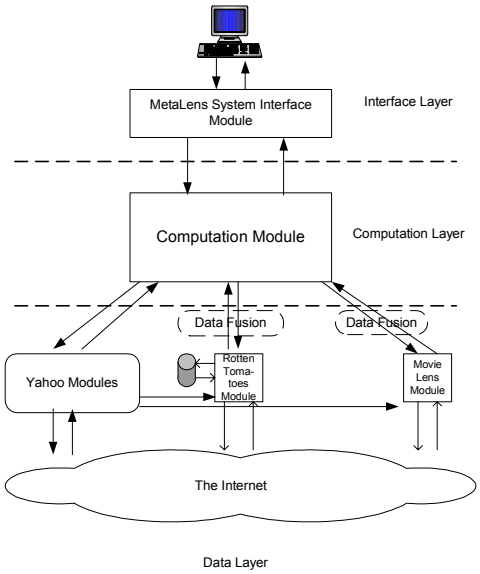

Figure 1: The MetaLens Architecture

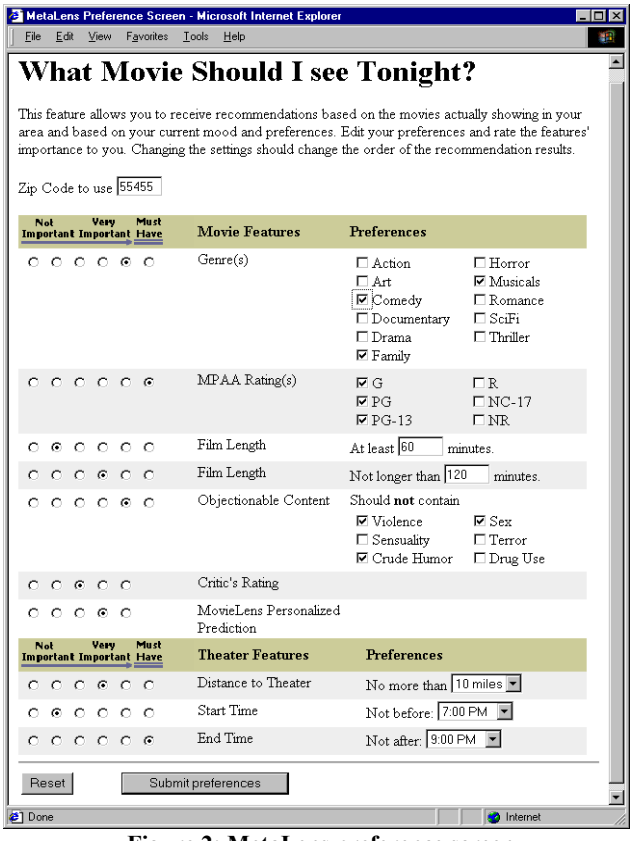

Figure 2: MetaLens preference screen

Prior to making any computation, the computation layer requests that the data layer produce the appropriate information concerning the theater/movie/show time triples for the user's ZIP Code. The data layer gathers the information from either from a local cache or through runtime data acquisition using three sources. Yahoo Movies (movies.yahoo.com) provides information concerning movies and theaters including genre, MPAA rating, content, show

times, and theater location. Rotten Tomatoes (www.rottentomatoes.com) provides critical review information including the number of critics rating the movie and the percentage of favorable reviews. Finally, MovieLens (movielens.umn.edu) provides personalized prediction information on a user/movie basis.

The Rotten Tomatoes and MovieLens modules must also negotiate a data fusion process to coordinate their data with that extracted from Yahoo Movies. While each of these three sites lists the title of each movie, we must resolve variations in title format (*The Thomas Crown Affair* vs. *Thomas Crown Affair, The*) and different releases of movies with the same name (Is that the 1999 or the 1968 version of *The Thomas Crown Affair*?).

Once all of the data is gathered, it is returned to the computation layer. The algorithm employed by the computation layer is based on the extended Boolean information retrieval algorithm proposed by Salton et al [15]. In essence, Mary's preferences create a query that says, "I want a movie that is a comedy or family movie rated no 'higher' than PG-13, containing no sex, violence or bad language, lasting less than two hours and, showing at a theater in my neighborhood." However, a traditional Boolean query based on these requirements will return only movies matching all of these features. This is problematic since most users will settle for a movie matching a significant subset of these features. Salton's algorithm provides a means to rank partial matches to Boolean queries.

MetaLens judges overall query fit based on recommendation scores from these multiple data sources. No attempt is made to resolve potential information conflicts. Instead, each piece of data is converted as-is, and the item match scores combined to calculate a query-fit score for each triple. These are returned to the interface layer where the recommendations are sorted to contain only the highest rated triple for each movie – each movie is recommended once in conjunction with the theater and show time that best fits the user's requirements – and the final recommendations displayed.

Thus, according to Figure 3, MetaLens recommends that Mary should take her nephew to see the 4:45 showing of *Toy Story 2* at the Yorktown Cinema Grill. Users may obtain additional information about any of the recommended movies or theaters by selecting the hyperlink of the item in question. This spawns a separate browser window containing information about the item. Furthermore, results may be "tuned" by the user who may modify the requirements or weights for each feature, thus modifying which subset of features is considered optimal.

4. META-RECOMMENDATION INTERFACES

While our eventual goal was to demonstrate that users prefer meta-recommendations when making decisions, we began our user studies by considering the design of the interface via which such recommendations would be made. Consider the recommendations presented in Figure 3. While the interface shows Mary that MetaLens finds *Toy Story 2* a slightly better choice than *The Tigger Movie*, it provides no information to help Mary decide to take this recommendation. A skeptical user might want to validate that *Toy Story 2* is indeed the better choice. An inquisitive user might wonder why MetaLens finds these much stronger choices than *Thomas* or *Chicken Run* – two movies which, on the surface, would also seem like reasonable alternatives. By following the "information links" available for each movie, Mary can discover that MovieLens predicts she will enjoy *Toy Story 2* slightly more than *The Tigger Movie*. Furthermore, she might notice that these are showing in theaters considered "close" to her location while *Thomas* and *Chicken Run* are showing at theaters considered "a long haul." Unfortunately, while this information is integrated into her final recommendations, Mary must search individual information screens to discover this.

MetaLens Score	Movie	Theater	Show Time
86.7	Toy Story 2	Yorktown Cinema Grill	4:45
80.7	The Tigger Movie	Cinema Cafe New Hope	12:15
56.8	Thomas and the Magic Railroad	GTI Shakopee Town Square Theatre	4:20
55.6	Chicken Run	UA Pavilion at Crossroads	6:45
53.4	Disney's The Kid	GC Har Mar 11	7:15
53.2	Return to Me	GTI Roseville 4 Theatre	5:00
52.9	Shower	Landmark Lagoon Cinema	7:10
51.3	Small Time Crooks	Mann Hopkins Cinema 6	7:10
50.7	Coyote Ugly	UA Pavilion at Crossroads	5:20
50.6	The Adventures of Rocky and Bullwinkle	Classic Cinemas Riverview Theatre	1:10
MetaLens Score	**Movie**	**Theater**	**Show Time**
50.2	Autumn in New York	GC Har Mar 11	7:00
46.3	Pokemon: The Movie 2000	GC Har Mar 11	5:10
44.8	Dinosaur	Cinema Cafe New Hope	4:10
43.9	The Perfect Storm	GC Har Mar 11	4:00

Figure 3: MetaLens recommendation screen

We hypothesized that users would find the recommendations more meaningful if additional "recommendation data" were displayed alongside the base recommendations. For example, columns containing data about "predicted rating" and "theater distance" may be of assistance to Mary. Experiments one and two were designed to consider what additional recommendation data, if any, users want displayed with their recommendations, and what format the recommendation interface should take[1].

4.1 Experiment One

4.1.1 Experimental Design

Experiment one was designed to consider this issue when the amount of recommendation data is limited. To begin, we identified four formats for displaying MetaLens' recommendations[2]:

[1] While it is equally important to consider what format the preferences interface should take, we chose to delay this research. It is our belief that no matter how good the interface for indicating preferences, users won't use a system if they don't find the recommendations helpful. Thus, an initial design of the preferences interface was selected based on "common sense" and commercial comparison-shopping sites such as Active Buyer's Guide (www.activebuyersguide.com) and Frictionless (www.frictionless.com).

[2] Names used are for clarity of explanation in this and future discussions and were never used with research subjects.

Default – Provides a ranked list of movie/theater/show time triples, and each triple's corresponding MetaLens score (Figure 3).

All – Displays all of the data considered in the recommendation of each triple.

Custom – Allows the user to select a subset of the data to include with recommendations via a "what information" screen.

Automatic – Selects which subset of the data to display based on the assumption that highly weighted features are important and should appear with the recommendations.

Subjects for this experiment were selected from the pool of active and established users of MovieLens. Members in this category had been members of MovieLens for a minimum of three months, had visited MovieLens a minimum of three times during that period, and had provided the system with at least ten ratings. A random sampling of users who met these criteria were sent email invitations to participate, and 50 responders were used in the study.

Subjects were asked to complete four randomized tasks, each consisting of a scenario for which they might be selecting a movie showing in local theaters ("Your 8 year old nephew is visiting. Pick a movie that is age appropriate, but that you might still enjoy."). Subjects used the MetaLens preference screen to indicate their requirements for the given scenario. Upon submission of their preferences, subjects were presented with the top recommendations presented in one of the four recommendation formats. These were also randomly ordered such that each subject saw each of the four recommendation formats once. Subjects were allowed to reconfigure and resubmit their preferences including, when appropriate, the additional information displayed. To finish the task, subjects were asked to select a triple they felt "fit the scenario."

Subjects completed mini-surveys between tasks asking them to provide an indication of their confidence in their movie selection, the helpfulness of the recommendations, and the extent to which they relied on previous knowledge in making their final selection[3]. Furthermore, upon completion of the experiment, subjects were asked to provide a unique ranking of the four interfaces they viewed from least to most helpful.

4.1.2 Results

Analysis of variance of the scores provided on the task-level surveys indicates that, regardless of the recommendation interface used, there is not a statistically significant difference in a subject's confidence or the amount of previous knowledge used in the decision-making process. However, the analysis does report a significant variance in subject-reported helpfulness of the interfaces (Table 1). When we control the overall alpha level at 0.05, Tukey's HSD post hoc test indicates subjects reported the default format less helpful than the other three formats (Table 2).

We observe similar results when evaluating the rankings provided during the exit survey. Recall that subjects were asked to rank uniquely the four formats from least helpful to most helpful (Table 3). Rank comparison analysis between pairs of formats via a binomial distribution ($\alpha = 0.05$) shows statistically significant preferences between each pair of formats (Table 4).

In considering these results, observe that 44 of the 50 subjects ranked the Default format the least helpful format. This was not surprising as the information provided with the Default format is minimal. Although subjects have access to additional information via the information links as previously described, the ability to view at least some of this information within the context of the recommendations appears beneficial. Also, observe that the "All" format was found most helpful by 35 of the 50 subjects. At first this surprised us as we felt that the All format would provide access to too much information. However, the data used in experiment one was rather limited. In fact, columns for all eight pieces of information were visible in a single browser window with no need for scrolling. Thus, it was easy for subjects to see the data that interested them and ignore the remaining data.

Table 1: Experiment one, ANOVA on task-level survey. (3 degrees of freedom in each analysis)

	F	p-value
Confidence	1.87	0.14
Helpfulness	8.37	0.00
Previous Knowledge	0.96	0.41

Table 2: Experiment one, format scores. [* indicates significance at $\alpha = 0.05$]

Score	Format	Mean	Std. Dev.
Confidence	Default	3.48	1.05
	Automatic	3.75	1.01
	All	3.77	0.87
	Custom	3.94	0.87
Helpfulness	Default	3.29*	1.18
	Automatic	3.89	0.72
	All	4.04	0.72
	Custom	4.04	0.66
Previous Knowledge	Default	3.27	1.07
	Automatic	3.11	1.08
	All	2.91	1.04
	Custom	3.02	1.05

Table 3: Experiment one, distribution of helpfulness rankings per format.

Format	Least			Most
Default	44	2	4	0
Automatic	3	37	6	4
All	1	4	10	35
Custom	2	7	30	11

Table 4: Experiment one, frequency with which Format A was ranked higher than Format B. [n=50, * indicates significance]

Format A	Format B	Frequency
All	**Default**	48*
Custom	**Default**	47*
Automatic	**Default**	45*
All	**Automatic**	44*
Custom	**Automatic**	40*
All	**Custom**	37*

[3] Subjects selected a text-based response for each of these questions. These responses were converted to an ordinal score (1-5) for analysis. A score of 1 corresponds with the least confidence, an interface that was very unhelpful, and a decision based largely on previous knowledge rather than use of the interface.

4.2 Experiment Two

4.2.1 Experimental Design

Experiment two was conducted to test if the results found in experiment one would change given a meta-recommender that used twice as much recommendation data (Table 5). The experimental design was similar to that used in experiment one. Once again, four recommendation formats were used. Subjects were selected based on criteria identical to that used in experiment one, although participants in experiment one were excluded. Thirty-two subjects completed experiment two.

Table 5: Data used in experiments one and two.

Experiment One	Added during Experiment Two
genre	avg. user rating
MPAA rating	film distributor
film length	release date
objectionable content	accommodations for handicapped
distance to the theater	discounted ticket prices
start/end time	min # of critics' ratings
% of "thumbs up"	% of major market "thumbs up"
MovieLens prediction	min # of major market ratings

In addition to a different data set, the Default and All formats from experiment one were replaced by two new formats: "Old All" and "True All." Observe that the "All" format in experiment one was considered the most helpful. We must consider whether this is because users recognized the display as all of the data used in the decision making process, or because it most closely represented the set of features the users actually wanted to see[4]. Thus, "Old All" displayed the eight features that were considered All in experiment one, while "True All" displayed the sixteen features which actually represent "All" in experiment two.

4.2.2 Results

Rank comparison analysis between the unique helpfulness rankings of pairs of formats indicates statistically significant preferences for the Custom format (Tables 8 and 9). These results indicate that as the amount of recommendation data increases, users find the True All format less helpful and begin to prefer the Custom format. In fact, the majority of subjects ranked the True All format the least helpful. Conversely, the majority of subjects ranked the Custom format as the most helpful.

Analysis of variance of the scores provided on the task-level surveys from experiment two does not show a significant difference between recommendation formats when comparing user confidence, helpfulness of the format, or the amount of previous knowledge used (Tables 6 and 7).

Table 6: Experiment two, ANOVA on task-level survey.

(3 degrees of freedom in each analysis)

	F	p-value
Confidence	0.38	0.77
Helpfulness	2.01	0.12
Previous Knowledge	0.08	0.97

[4] Further analysis to determine the "most important" recommendation data suggests this is not a trivial consideration and goes beyond the scope of this paper.

Table 7: Experiment two, format scores. [α = 0.05]

	Format	Mean	Std. Dev.
Confidence	Old All	3.79	1.10
	Automatic	3.61	1.05
	True All	3.47	1.36
	Custom	3.58	1.09
Helpfulness	Old All	3.82	0.90
	Automatic	3.39	1.05
	True All	3.17	1.18
	Custom	3.48	0.96
Previous Knowledge	Old All	3.14	1.04
	Automatic	3.23	1.02
	True All	3.10	1.32
	Custom	3.19	0.83

Table 8: Experiment two, distribution of helpfulness rankings per format.

	Least			Most
Old All	8	10	7	7
Automatic	10	8	8	6
True All	13	7	7	5
Custom	1	7	10	14

Table 9: Experiment two, frequency with which Format A was ranked higher than Format B.

[n=32, * indicates significance]

Format A	Format B	Frequency
Custom	**True All**	24*
Custom	**Automatic**	23*
Custom	**Old All**	22*
Automatic	**True All**	19
Old All	**Automatic**	18
Old All	**True All**	17

4.3 Impact

Our results indicate that users appreciate the rich set of recommendation data used in the decision-making process and that they prefer to have some portion of this data included with

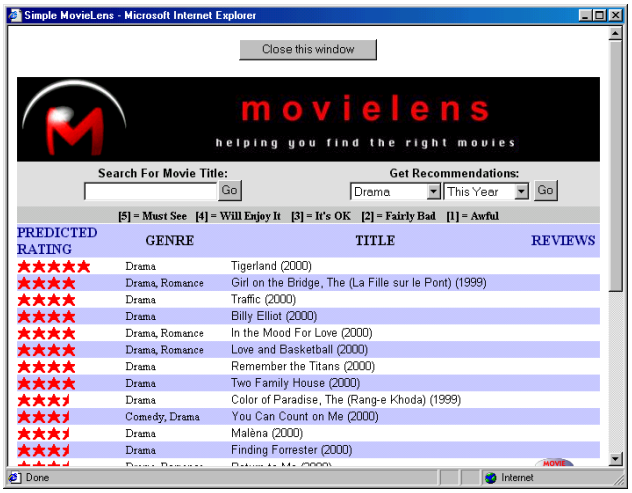

Figure 4: MovieLens++ Search Results Screen

their recommendations. When the amount of data is small enough for the system to display in a manner such that a user can extract meaning, users prefer having access to all of the data. When it becomes too large to display in a meaningful fashion, then users prefer to control the display of the data themselves. In both situations, automatic selection of the data – even that based on the user's own priorities – is not viewed as preferable. We believe these findings provide developers with a meaningful starting point for the construction of future meta-recommender applications, increasing the likelihood that users will find their systems helpful.

5. RECOMMENDER SYSTEM COMPARISONS

5.1 Experiment Three

5.1.1 Experimental Design

Experiment three was designed to consider whether users find a meta-recommender more helpful than "traditional" systems offering access to the same data. Participants for this experiment were selected using criteria identical to those used in the previous experiments, although participants from previous experiments were excluded from participation. Sixty subjects were asked to complete three sets of three tasks. Similar to previous experiments, each represented a situation for which they would be attempting to select a movie showing in local theaters. For each set of tasks, subjects used one of three recommender interfaces presented with a random ordering. The three interfaces used were MovieLens++, ContentLens, and MetaLens. While the same data was available via all three interfaces, they differed in the degree to which each integrated the data for a final recommendation.

MovieLens++ provides users with access to all of the data, but does not provide a means to integrate the data into a single recommendation. Users have access to two separate systems and may make their selections based on either or both of these systems. CF-based recommendations are obtained through MovieLens (Figure 4), but these come with little information about the content of the movies being recommended. Users may choose to coordinate these recommendations with a separate movie-listings system (Figure 5) offering access to a variety of content data and reviews. Users must manually combine the information they gather into a single "recommendation." In essence, MovieLens++ best approximates the way users might currently address such scenarios.

ContentLens provides separate integration of IF-based and CF-based recommendations but does not combine the results. Through an interface nearly identical to that used in experiments one and two, users submit content queries, receive integrated IF-based recommendations (Figure 6) using the Custom recommendation format, and may choose to view additional information for any of the recommended items. However, unlike previous interfaces, ContentLens does not include CF-based recommendations. Instead, subjects are provided a "show me" link for each movie on the recommendation list. Selecting this creates a separate window displaying the results of a MovieLens title search for the movie in question (Figure 7). If users choose to use both IF and CF-based recommendations, they must manually integrate the results.

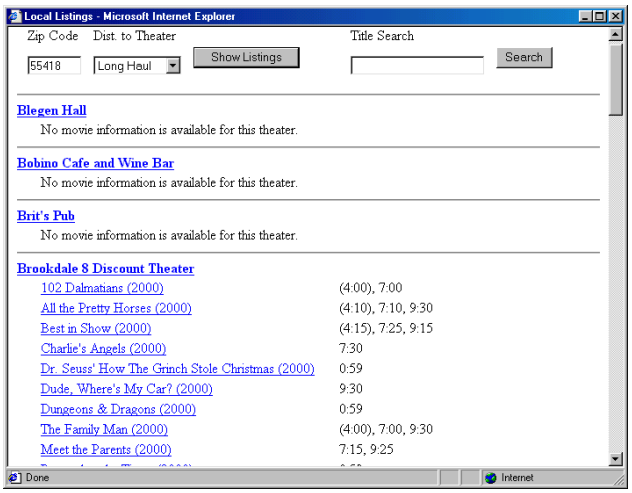

Figure 5: MovieLens++ Movie Listing Screen

Finally, MetaLens provides users with a fully integrated meta-recommendation system. As with ContentLens, subjects use the preference screen to indicate their requirements for the given scenario. However, unlike ContentLens, users receive final recommendations that include the integration of CF-based recommendations. While the difference between ContentLens and MetaLens may seem minor, we were interested in whether or not even this small degree of additional integration makes a difference to users.

While using a given interface, subjects were allowed to interact freely with the interface and manipulate it as much or as little as they felt was necessary in order to make their decision. To complete each task subjects selected a triple they felt "solved" the scenario at hand. Similar to the previous experiments, subjects completed interface level surveys asking them to indicate how confident they were with the interface they had just used and how much their decisions were based on prior knowledge. Furthermore, subjects completed an exit survey in which they provided a unique ranking of the three interfaces.

Figure 6: ContentLens Recommendation Screen

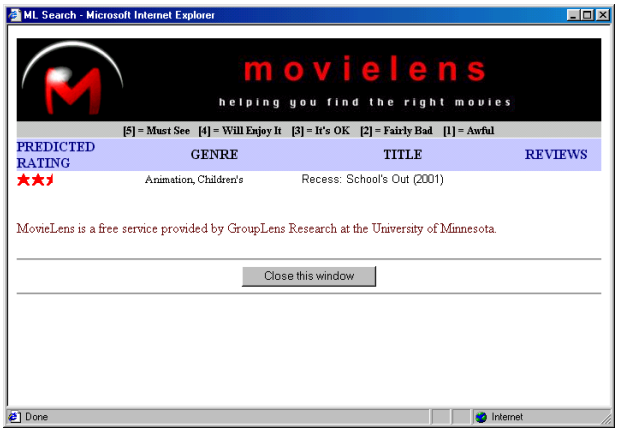

Figure 7: ContentLens access to MovieLens via "show me"

5.1.2 Results

Although MovieLens++ is likely the closest interface to how subjects currently solve these scenarios, results indicate that subjects prefer the alternatives. Analysis of variance indicates significant differences when considering scores for both subject confidence and the amount of previous knowledge required (Table 10). Post hoc analysis indicates that subjects are less confident and require more previous knowledge when using the MovieLens++ interface vs. either of the other interfaces (Table 11).

Table 10: Experiment three, ANOVA on task-level survey.

	F	Sig.
Confidence	4.54	0.01
Previous Knowledge	11.68	0.00

Table 11: Experiment three, interface scores.

[* indicates significance at α = 0.05]

	Interface	Mean	Std. Dev.
Confidence	MovieLens++	3.87*	1.10
	ContentLens	4.20	0.68
	MetaLens	4.32	0.70
Previous Knowledge	MovieLens++	3.03*	1.25
	ContentLens	3.73	0.86
	MetaLens	3.78	0.64

Additionally, results indicate that subjects find MetaLens to be more helpful than the alternatives. As with prior experiments, subjects were asked to rank uniquely the helpfulness of each interface. Binomial distribution analysis indicates that subjects found MovieLens++ the least helpful and MetaLens the most helpful (Tables 12 and 13). Results suggest that users do find even a small degree of additional integration to be beneficial.

Table 12: Experiment three, distribution of helpfulness rankings per interface.

Interface	Least		Most
MovieLens++	38	10	12
ContentLens	14	32	14

Table 13: Experiment three, frequency with which Interface A was ranked higher than Interface B.

[n=60, * indicates significance]

Interface A	Interface B	Frequency
MetaLens	**MovieLens++**	44*
ContentLens	**MovieLens++**	44*
MetaLens	**ContentLens**	42*

5.1.3 Impact

It is our belief that meta-recommenders provide the "best of both worlds." They allow recommendations to be based on persistent knowledge about the user, and they allow the user to input ephemeral requirements. Better yet, they do so by providing the users with specific control over how this recommendation data is combined. These results seem to confirm this belief. Although these results are currently limited to the domain of movies, they suggest a gap between current recommender system design and the actual needs of users, and we believe that meta-recommender systems will help fill this gap.

6. FUTURE WORK

We are interested in several areas of future work concerning meta-recommenders. These include the transfer of meta-recommenders to other domains, the role of personalization, and real-world acceptance of meta-recommendation systems.

While the architecture used in MetaLens was designed to be domain neutral, and though we expect our results to generalize to other domains, including e-commerce, web search engines, and knowledge management, we must consider how the design of meta-recommenders may change based on the domain. What adaptations have to be made in domains where item-features are less objective? For example, while we would expect a meta-recommender in the domain of books to work, we suspect that users may find the recommendations from such systems less helpful than those from domains such as automobiles or technical reports. How do interfaces change when item-feature weights are no longer sufficient for indicating how recommendations should be combined? How do such systems handle access to privileged data?

While users may not mind providing configuration information to a meta-recommender when the length of the task is relatively short or when encountering a new situation, it is very likely that users will not want to take the time to configure the system for longer or more frequent tasks. For example, if visits from Mary's nephew are a frequent occurrence, we would expect that Mary would want to have a mechanism for storing the configuration representing her preferences. How can meta-recommenders be extended through the use of personalization profiles? How will such changes affect the underlying system or the interface? How will the implementation of personalization impact the usage of such systems?

Finally, what are the real-world acceptance rates of meta-recommenders? We must acknowledge that our users for all three of these studies were drawn from experienced MovieLens users who may not represent users at large. Will real-world users avoid using meta-recommenders in the short-term because they look too complicated or because they seem too "magical?" Will they stop using them in the long-term after the "novelty" of such systems

wears off? Will decisions that users made in short, controlled studies turn out to be different from those made by long-term users?

7. CONCLUSIONS

In this paper we have introduced meta-recommenders as a new way to help users find recommendations that are understandable, usable, and helpful. A series of controlled use experiments in the domain of movies indicates that users prefer that these systems provide recommendation data alongside the recommendations and prefer to have control to the selection of this data. Additionally, results suggest that users prefer the recommendations provided by these systems when compared with recommendations provided by "traditional" recommender systems. All told, we feel these results provide a meaningful foundation for the design of future meta-recommenders.

8. REFERENCES

[1] Balabanovic, M. and Shoham, Y. (1997). Fab: Content-based Collaborative Recommendations. CACM 40(3) pp.66-72.

[2] Belkin, N.J. and Croft, W.B. (1992). Information Filtering and Information Retrieval: Two Sides of the Same Coin? CACM 35(12) pp.29-38.

[3] Boone, G. (1998). Concept Features in RE:Agent, an Intelligent Email Agent. Proceedings of Autonomous Agents 98. pp.141-148.

[4] Claypool, M., et al. (1999). Combining Content-Based and Collaborative Filters in an Online Newspaper. ACM SIGIR Workshop on Recommender Systems

[5] Cohen, W.W. (1996). Learning Rules that classify E-mail. Proceedings of the AAAI Spring Symposium on Machine Learning on Information Access.

[6] Goldberg, D., Nichols, D., Oki, B.M., and Terry, D. (1992). Using Collaborative Filtering to Weave an Information Tapestry. CACM 35(12) pp.31-70.

[7] Good, N., et al. (1999). Combining Collaborative Filtering with Personal Agents for Better Recommendations. Proceedings of AAAI-99 pp.439-446.

[8] Hill, W., Stead, L., Rosenstein, M., and Furnas, G. (1995). Recommending and Evaluating Choices in a Virtual Community of Use. CHI-95 pp.194-201.

[9] Khamba, T., Bharat, K., and Albers, M. (1993). The Krakatoa Chronicle - An Interactive, Personalized, Newspaper on the Web. Fourth International World Wide Web Conference pp.159-170.

[10] Lawrence, R.D., et al. (2001). Personalization of Supermarket Product Recommendations. Data Mining and Knowledge Discovery 5(1/2) pp.11-32.

[11] Maes, P. (1994). Agents that Reduce Work and Information Overload. CACM 37(7) pp.31-40.

[12] Moukas, A. and Zacharia, G. (1997). Evolving a Multi-agent Information Filtering solution in Amalthaea. Proceedings of Autonomous Agents 97 pp.394-403.

[13] Nakamura, A. and Abe, N. (2000). Automatic Recording Agent for Digital Video Server. Proceedings of MM-00 pp.57-66.

[14] Resnick, P., et al. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. CSCW-94 pp.175-186.

[15] Salton, G., Fox, E., and Wu, H. (1983). Extended Boolean Information Retrieval. CACM 26(11) pp.1022-1036.

[16] Schafer, J.B., Konstan, J.A., and Riedl, J. (2001). E-Commerce Recommendation Applications. Data Mining and Knowledge Discovery 5(1/2) pp.115-153.

[17] Shardanand, U. and Maes, P. (1995). Social Information Filtering: Algorithms for Automating Word of Mouth. Proceedings of CHI-95 pp.210-217.

[18] Terveen, L., Hill, W., Amento, B., McDonald, D., and Creter, J. (1997). PHOAKS: A System for Sharing Recommendations. CACM 40(3) pp.59-62.

[19] Wasfi, A. (1999). Collecting User Access Patterns for Building users Profiles and Collaborative Filtering. (IUI99) pp.57-64.