

2008

Online Recommendation System

Ankit Khera
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_projects

Part of the [Computer Sciences Commons](#)

Recommended Citation

Khera, Ankit, "Online Recommendation System" (2008). *Master's Projects*. 97.

DOI: <https://doi.org/10.31979/etd.v33b-ap2s>

https://scholarworks.sjsu.edu/etd_projects/97

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

Online Recommendation System

A Master's Project

Presented to

The Faculty of Computer Science

San Jose State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

by

Ankit Kamalkishore Khera

Fall 2008

© 2008

Ankit Kamalkishore Khera

ALL RIGHTS RESERVED

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

Dr. Chris Tseng

Dr. SoonTee Teoh

Dr. Anup Hosangadi

APPROVED FOR THE UNIVERSITY

ACKNOWLEDGEMENT

I would like to thank Dr. Tseng for helping, and guiding me throughout the semester in my master's project. I would also like to thank Dr. Teoh, and Dr. Anup for being part of my committee and supporting the concept.

ABSTRACT

Online Recommendation System

By Ankit Kamalkishore Khera

The vast amount of data available on the Internet has led to the development of recommendation systems. This project proposes the use of soft computing techniques to develop recommendation systems. It addresses the limitations of current algorithms used to implement recommendation systems, evaluation of experimental results, and conclusion. This report provides a detailed summary of the project "Online Recommendation System" as part of fulfillment of the Master's Writing Project, Computer Science Department, San Jose State University's. The report includes a description of the topic, system architecture, and provides a detailed description of the work done till point. Included in the report are the detailed descriptions of the work done: snapshots of the implementations, various approaches, and tools used so far. The report also includes the project schedule and deliverables.

TABLE OF CONTENTS

1. Introduction	8
2. Project Overview	10
3. Recommendation Systems:	13
3.1. Classification of Recommendation Systems	13
3.2. Methodologies	13
4. Implementation:	22
4.1 System Screenshots	22
4.2. Technical Specifications	31
4.3. Software Methodology	31
4.4. Web Services	32
4.5. Testing	35
5. Advantages of the System	36
6. Project Schedule/ Deliverables	37
7. Conclusion	39
References	40

LIST OF FIGURES

Figure 1: System Architecture	10
Figure 2: Taste architecture (Sean, 2008)	15
Figure 3: Vogoo implementation	15
Figure 4: Movie rating parameters	16
Figure 5: User/Movie ratings matrix(Pereira, 2006)	16
Figure 6: User similarity matrix(Klir, 1988)	17
Figure 7: Similar users(Klir, 1988)	18
Figure 8: Pearson's Correlation formula.	20
Figure 9: Cyclical methodology (Burback, 1998)	31

1. Introduction

Web Discovery applications like Stumble Upon, Reddit, Digg, Dice (Google Toolbar) Etc to name a few are becoming increasingly popular on the World Wide Web. Information on the Internet grows rapidly and users should be directed to high quality Websites those are relevant to their personal interests. However, there is no way to Judge these web pages. Displaying quality content to users based on ratings or past Search results are not adequate. There's a lacking of powerful automated process combining human opinions with machine learning of personal preference.

The goal of this project is to study recommendation engines and identify the shortcomings of traditional recommendation engines and to develop a web based recommendation engine by making use of user based collaborative filtering (CF) engine and combining context based results along with it. The system makes use of numerical ratings of similar items between the active user and other users of the system to assess the similarity between users' profiles to predict recommendations of unseen items to active user. The system makes use of Pearson's correlation to evaluate the similarity between users. The results show that the system rests in its assumption that active

users will always react constructively to items rated highly by similar users, shortage of ratings of some items, adapt quickly to change of user's interest, and identification of potential features of an item which could be of interest to the user.

This project will focus on making use of context based approach in addition to CF approach to recommend quality content to its users. It would be exploiting available contextual information, analyzing and summarizing user queries, and linking the metadata like tags and feedback to a richer information model to recommend content. The project also aims at using soft computing technologies to create an automated process and develop an intelligent web application. The System would benefit those users who have to scroll through pages of results to find relevant content.

2. Project Overview

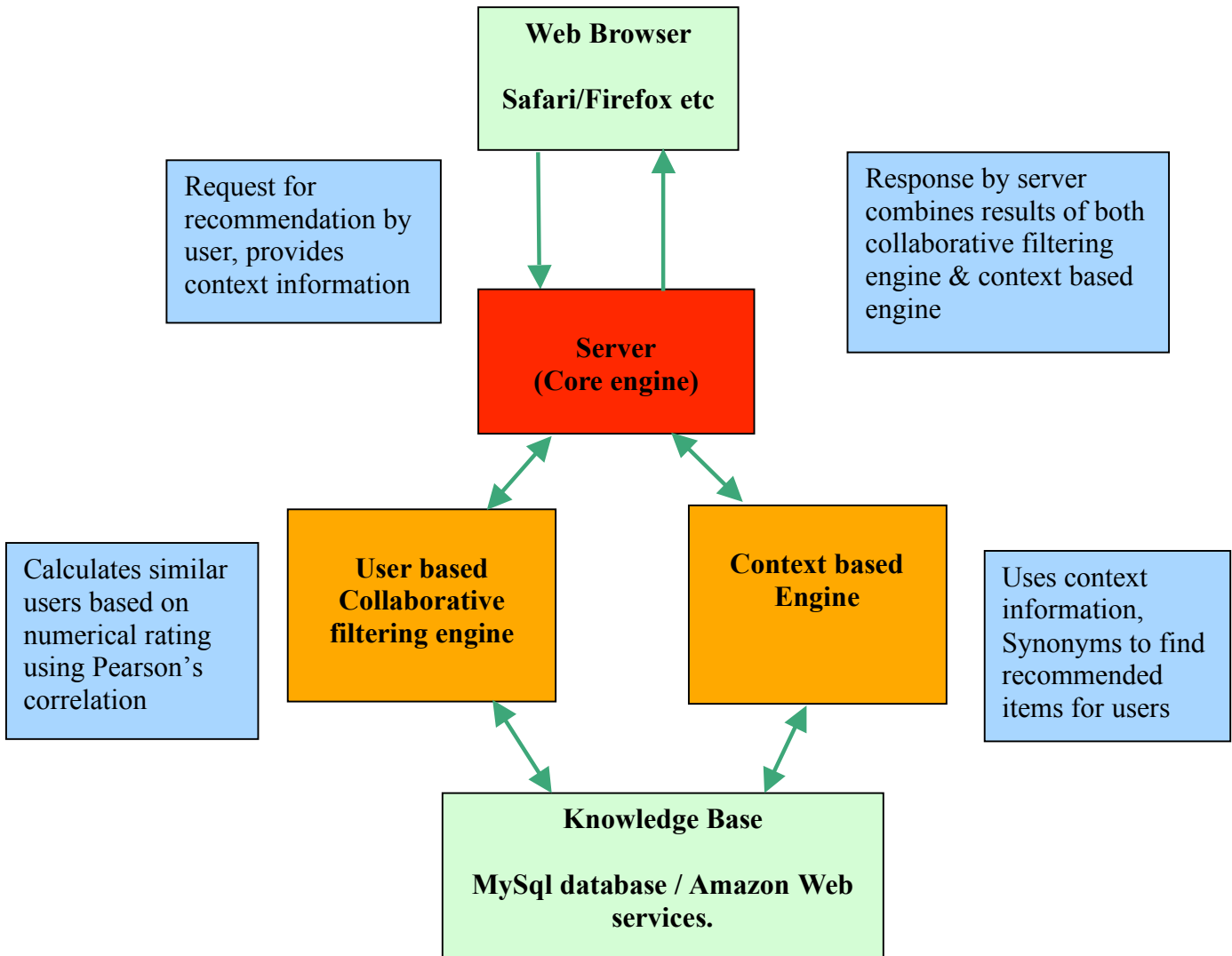


Figure 1: System Architecture

Description:

1. User types in the URL for the system on a Web Browser.
2. User logs into the system using his `userid`.

3. The user chooses from amongst the type 2 different types of recommendation systems available.
4. If the user chose 'Collaborative Filtering' option, the system calculates similar users making use of engineering algorithms, and then recommends items to the users based on the most similar user.
5. If the user chose 'Context based Filtering' option, the system then makes use of the context information, and Synonym Finder to make predictions.
6. The System provides the user with following functionalities:
 1. Different Search Features to search items
 - 1.1. Auto search complete: The System provides its users with auto search box, which automatically pulls the books matching the keywords typed, by the user. The auto search feature is automatically activated after the user has typed 3 characters. The feature also displays the averages rating of the book besides it. Auto search complete would display 10 results matching the users keywords. If the user is unable to find the match amongst the 10 results he/she can click on the 'more' link provided at the bottom of the results to view more results matching their search.
 - 1.2. The System also provides users with Advance search

benefits; users can search for books matching authors, publisher, ISBN etc. Users can also view all the available versions of a particular books released by the author so far.

2. Rate Books: Users can rate the movies which they like/dislike by providing numerical rating on a scale of one to ten. The system also allows the users to tag their books, and provide feedback.
3. View/Edit past books: The system allows the users to view and edit their past ratings, tags, and feedback.

3. Recommendation Systems:

Recommendation system is an information filtering technique, which provides users with information, which he/she may be interested in.

3.1. Classification of Recommendation Systems

Most of the recommendation systems can be classified into either User based collaborative filtering systems or Item based collaborative filtering systems (Billsus, 1998). In user based collaborative filtering a social network of users sharing same rating patterns is created. Then the most similar user is selected and a recommendation is provided to the user based on an item rated by most similar user. In item based collaborative filtering relationship between different items is established then making use of the active user's data and the relationship between items a prediction is made for the active user (Machine, 2008).

3.2. Methodologies

The proposed system makes use of Pearson's correlation to implement User based collaborative filtering, and context, Synonym

Finder to implement Context based filtering techniques to generate recommendations for the active user.

Following are the methodologies used/researched so far:

- Alternative approaches using engineering algorithms:
 - Taste: Taste is a flexible, fast collaborative filtering engine for Java. It takes the users' preferences for items and The engine takes users' preferences for items ("tastes") and recommends other similar items (Sean, 2008).

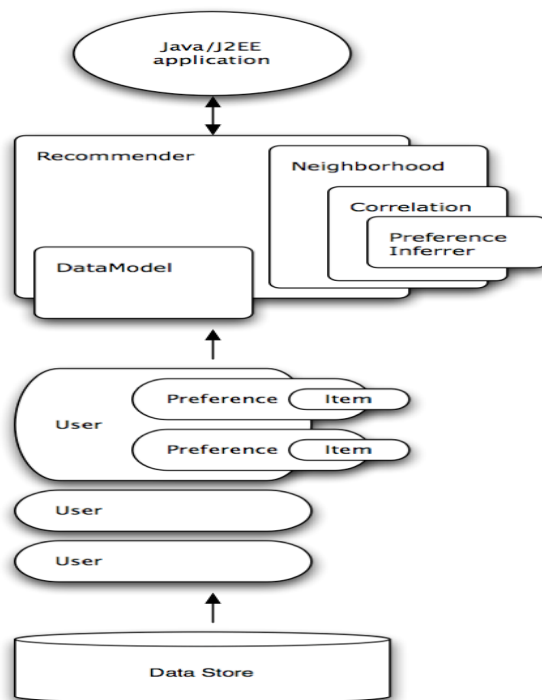
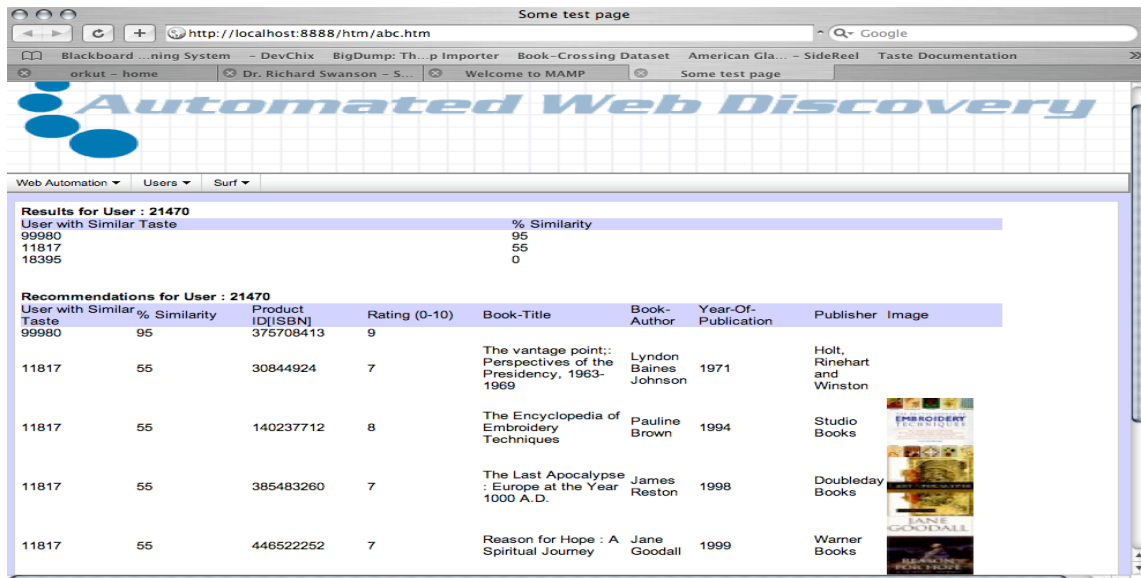


Figure 2: Taste architecture (Sean, 2008)

1. Vogoo: Vogoo is a php based collaborative filtering and recommendation library. It recommends items to users, which matches their tastes. It calculates similarities between users and creates communities based on them. The figure below shows the results of using vogoo to generate similar taste sharing users and recommendations made by the most similar users (Droux, 2008).



Results for User : 21470

User with Similar Taste	% Similarity
99980	95
11817	55
18395	0

Recommendations for User : 21470




User with Similar Taste	% Similarity	Product ID[ISBN]	Rating (0-10)	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image
99980	95	375708413	9					
11817	55	30844924	7	The vantage point; Perspectives of the Presidency, 1963-1969	Lyndon Baines Johnson	1971	Holt, Rinehart and Winston	
11817	55	140237712	8	The Encyclopedia of Embroidery Techniques	Pauline Brown	1994	Studio Books	
11817	55	385483260	7	The Last Apocalypse : Europe at the Year 1000 A.D.	James Reston	1998	Doubleday Books	
11817	55	446522252	7	Reason for Hope : A Spiritual Journey	Jane Goodall	1999	Warner Books	

Figure 3: Vogoo implementation

- Fuzzy Logic: Here I tried to make use of fuzzy logic to calculate similar users. We use a hybrid approach (Christakou, 2005) and accept inputs from the users in three forms:
 - Numeric rating between 0.0 – 1.0
 - Three rating for context between 0.0 – 1.0

○ Tags (free tagging)

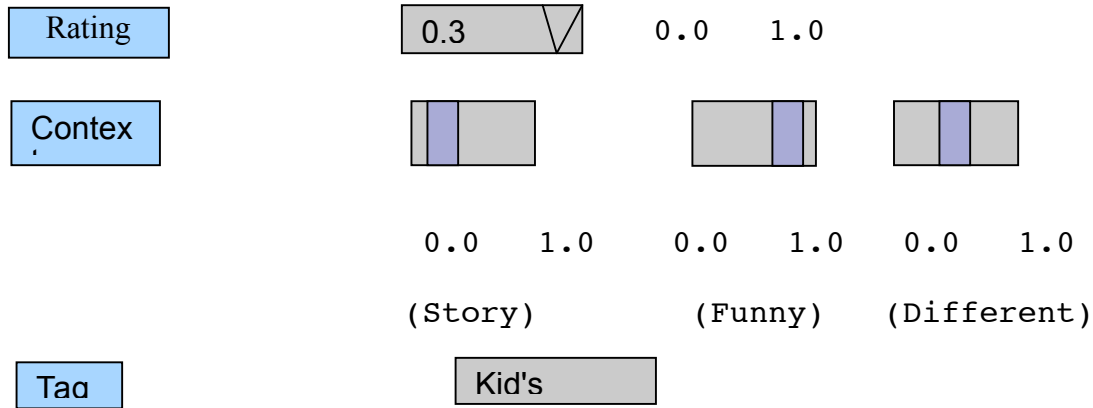


Figure 4: Movie rating parameters

In order to calculate similar users for the active user we first reduce the three ratings for any movie to a single movie rating between zero and one, after that we generate a user/movie matrix(Pereira, 2006) as shown in the following fig

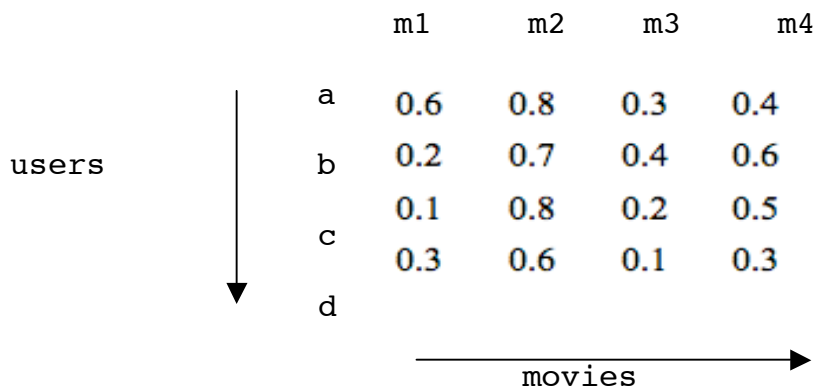


Figure 5: User/Movie ratings matrix(Pereira, 2006)

Once the (user/movie rating) matrix is generated we apply fuzzy

logic to it and generate a user similarity matrix as shown in the figure:



Figure 6: User similarity matrix(Klir, 1988)

The above figure shows the user similarity matrix in which the ratings between different users are listed. Now in order to calculate similar users we define α to be a partition set where, $\alpha > 0$ for example let $\alpha = \{0.4, 0.5, 0.8, 0.9, 1.0\}$. Now for every value of α we will get a similar user group satisfying the condition example: $(ab=0.8) > (\alpha=0.4)$ so user 'a' and user 'b' are related(Klir, 1988). This is shown in the figure below:

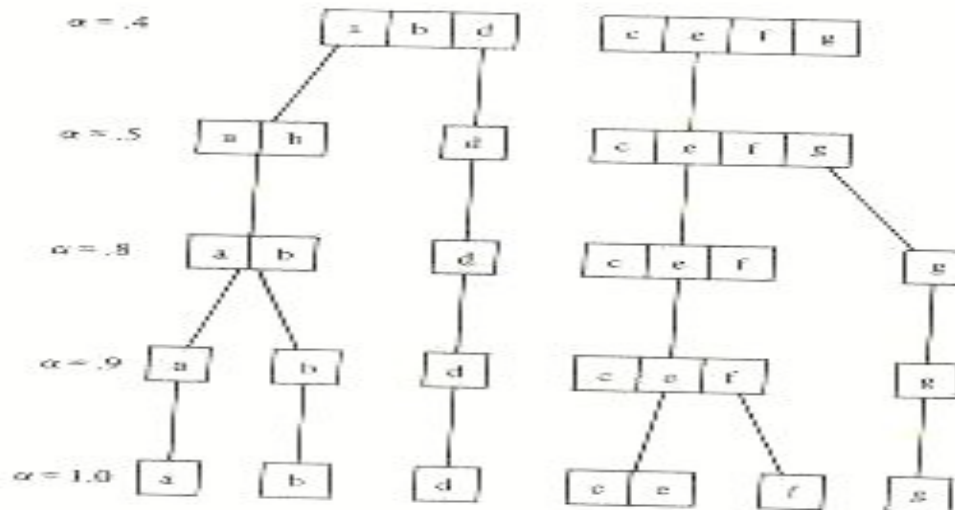


Figure 7: Similar users(Klir, 1988)

Following is the currently used approach:

User Request: - User makes a request for recommendation by clicking on the recommendation menu. User is asked to provide contextual information.

Server: - The information provided by the user is send to the server. The server is composed on 2 sub engines: user based collaborative filtering engine, and context based engine. The server sends users request to both the sub engines.

User based collaborative filtering engine: - calculates similar

users based on the numerical ratings of common items rated by the active users and other users of the system. The system achieves this by making use of the Pearson's correlation.

- **Pearson's Correlation:** is a way to find out similar users. The correlation is a way to represent data sets on graph. Pearson's correlation is x-y axis graph where we have a straight line known as the best fit as it comes as close to all the items on the chart as possible. If two users rated the books identically then this would result as a straight line (diagonal) and would pass through every book rated by the users. The resultant score in this case is 1. The more the users disagree from each other the lower their similarity score would be from 1. Pearson's Correlation helps correct grade inflation. Suppose a user 'A' tends to give high scores than user 'B' but both tend to like the book they rated. The correlation could still give perfect score if the differences between their scores are consistent.

Algorithm:

The algorithm first finds all the common books rated by user 'A' and user 'B'. It then finds out the sums and sum of the squares of the ratings for both the users. It then finds the sum of the products of their ratings. These scores are then used to find out Pearson's correlation.

$$sim(c, p) = \frac{\sum_{j \in I_c} (r_{c,j} - \bar{r}_c)(r_{p,j} - \bar{r}_p)}{\sqrt{\sum_{j \in I_c} (r_{c,j} - \bar{r}_c)^2} \sqrt{\sum_{j \in I_c} (r_{p,j} - \bar{r}_p)^2}}$$

Figure 8: Pearson's Correlation formula.

Context Engine: - was initiated with an item based collaborative filtering approach example: Amazon related books etc. The item based collaborative filtering approach was build using Pearson's correlation, but instead of calculating similarity between users here we calculated similarity between items. The results were good but it did not meet the goals set for the context-based engine initially. The system did not give good results due to lack of ratings, the system did not fill up the deficiencies of the CF based engine, the system did not do justice to the word 'related' items, because of all these reasons the below approach was followed. This engine makes use of contextual information provided by the user, synonyms, meta data about the products to find recommended items.

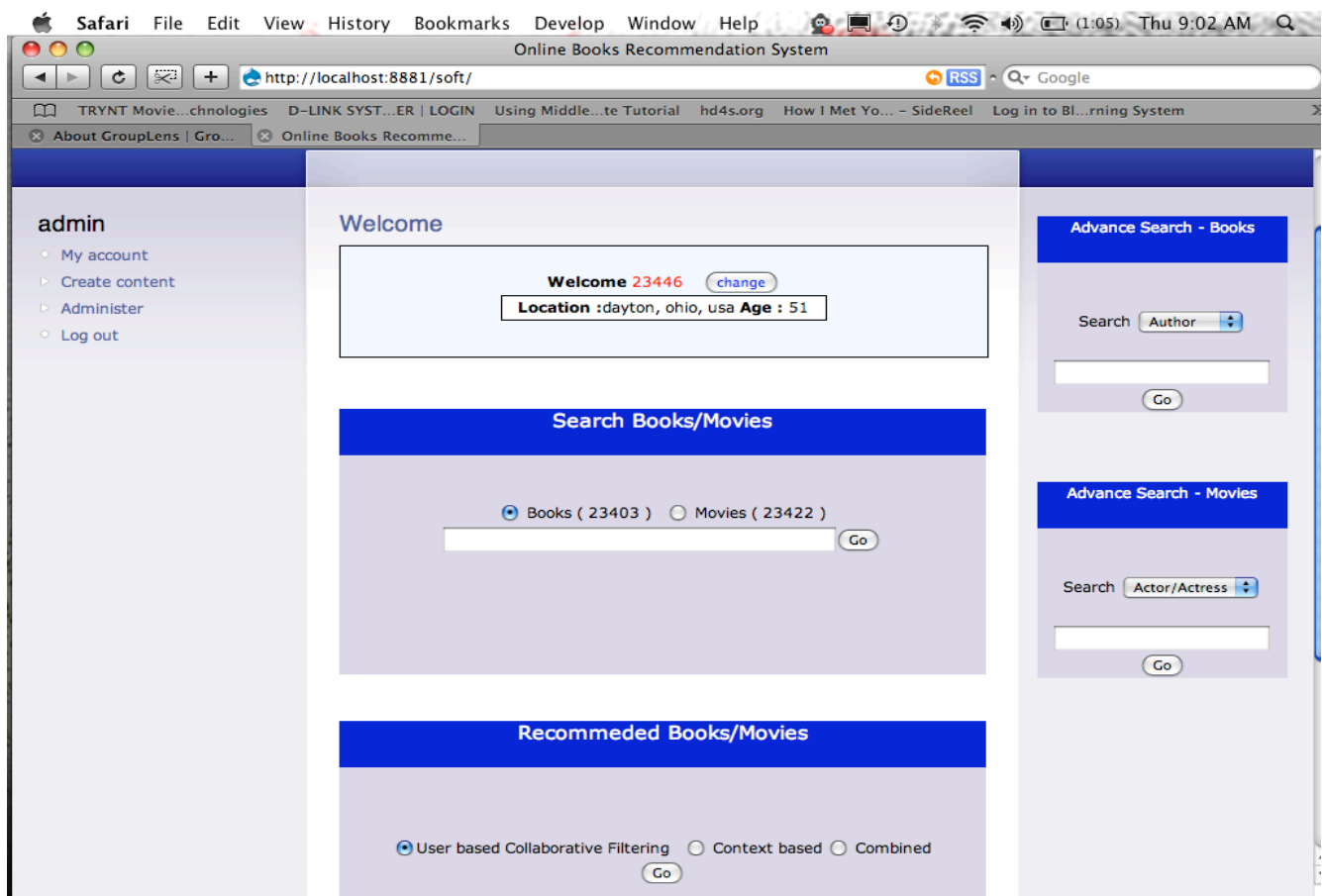
- The system first asks the user to provide context information example: author, publisher, and ISBN, and tags. The system does not expect the user to provide the complete

author, ISBN; publisher name example 'oxf' could be typed in as part of publisher name. The system then asks the user to type any free keywords. Once the user clicks the submit button. The information is first fed into the query engine, which makes use of the context information to narrow down the search results. The free keywords are fed to the Synonym Finder engine, which makes use of screen scraping techniques to find different senses of the entered keywords. This is done to find out the correct sense of the keyword used. All the results of the query parser (books) and Synonym Finder (senses) are then shown to the user. The user is then expected to see the results and if he/she is not yet satisfied, they can click on the 'refine' button, as soon as the refine button is clicked the results the Synonym Finder i.e. different senses are fed to the query parser. Simultaneously a web service call is made to the Amazon Web Services to capture the editorial reviews of the books shown to the user earlier. Once this is done. The parser searches for these senses in the editorial reviews, if a match is found then the results (books) are shown in that category. The advantage of using this approach is that it helps to cover the disadvantages of the User based collaborative filtering engine like lack of user ratings, false ratings etc and deliver accurate predictions to the users.

4. Implementation:

4.1 System Screenshots

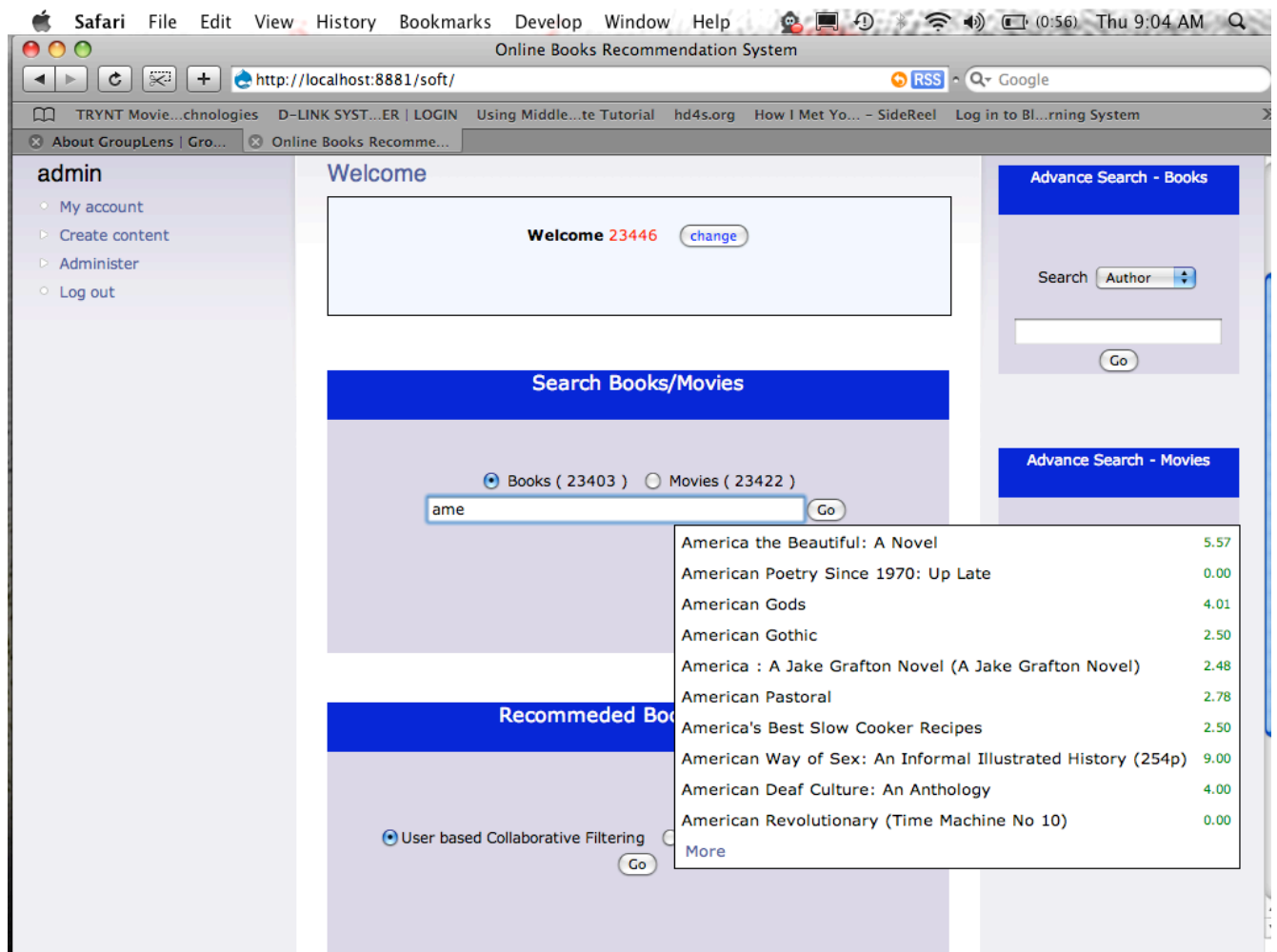
1) Home Screen



This is how the home screen for the online recommendation system looks like. To begin recommendation process the user first has to

enter the 'userID'. We can see this in the above figure were User '23446' has just logged. The session for this user has to remain active through out the recommendation process in order for the system to make recommendations.

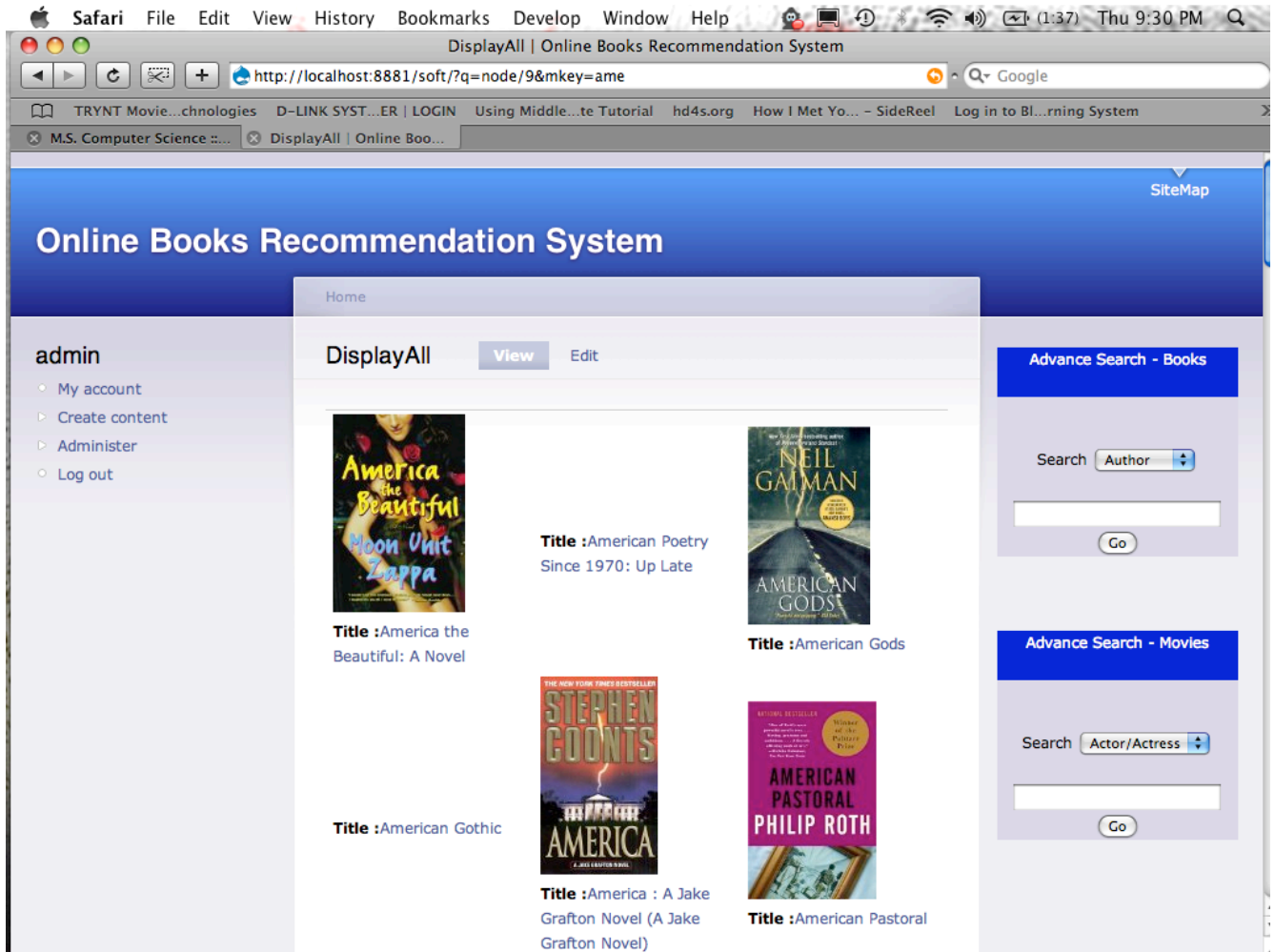
2) Books Search



The above figure shows the implementation of the auto search feature

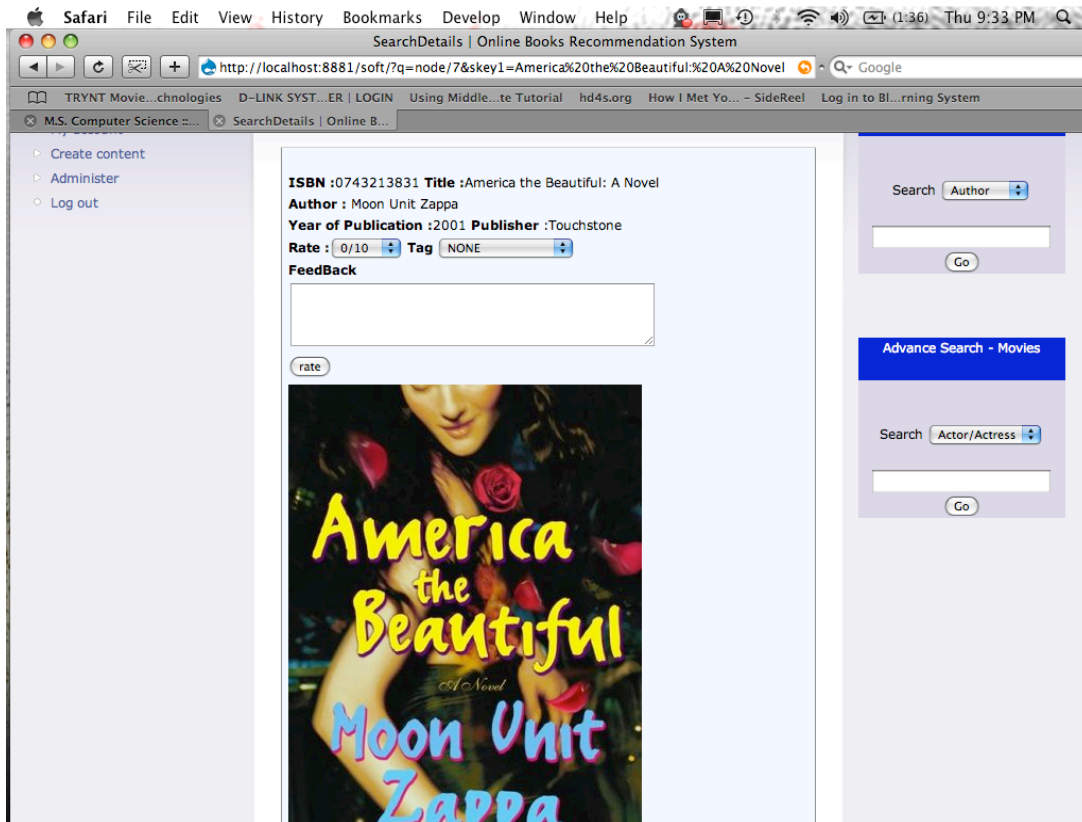
as described above, the figure displays 10 books with their average ratings along side matching the keyword 'ame' entered by the active user. If the match is not seen the more link can be clicked to see other matching results.

3) `More` Keyword



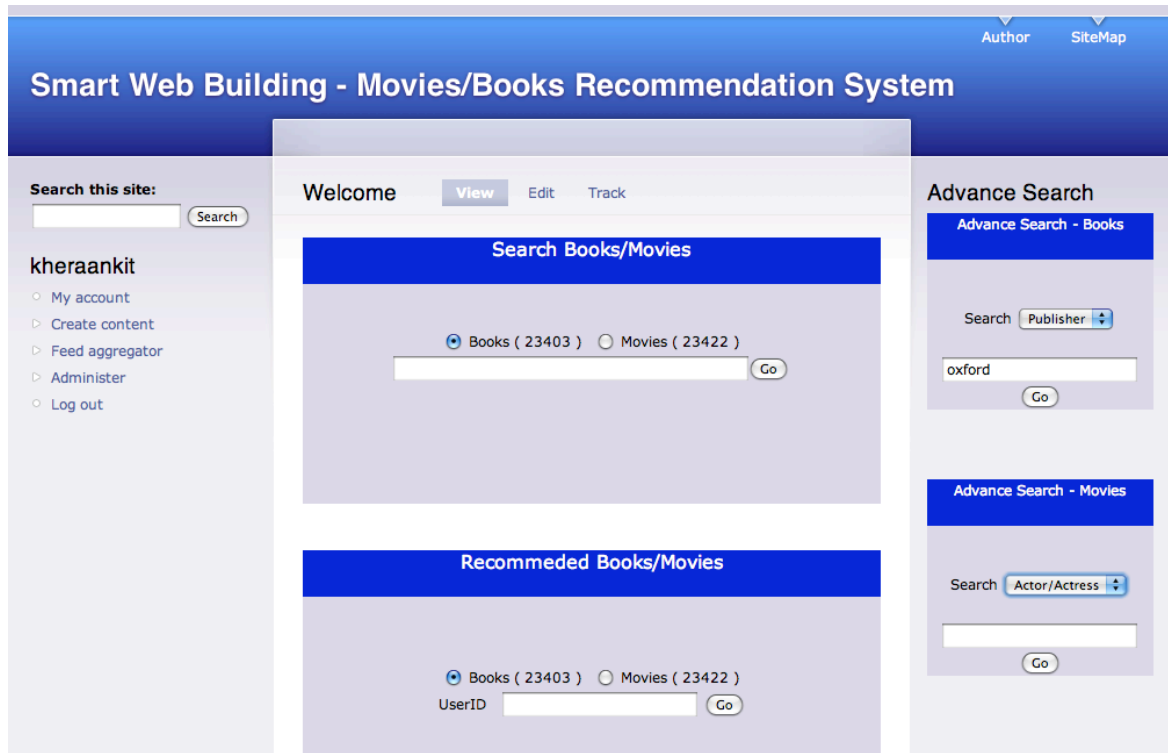
The above figure shows the results of top books matching the keyword 'ame' when the more link is clicked.

4) Results Books Search

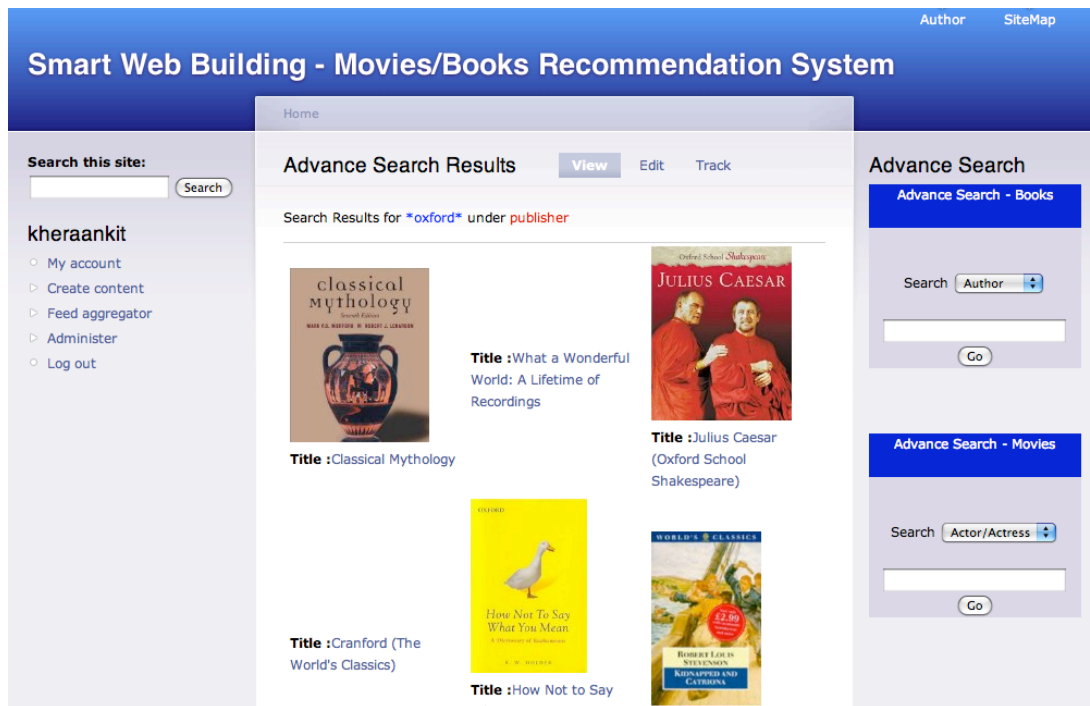


The above figure shows the details of the book like isbn, title, author, year of publication, publisher, rating, tag, feedback, and description [not visible in snapshot due to lack of space] etc. The user can rate the new book or update his current ratings here.

5) Advance Search Books (publisher): -

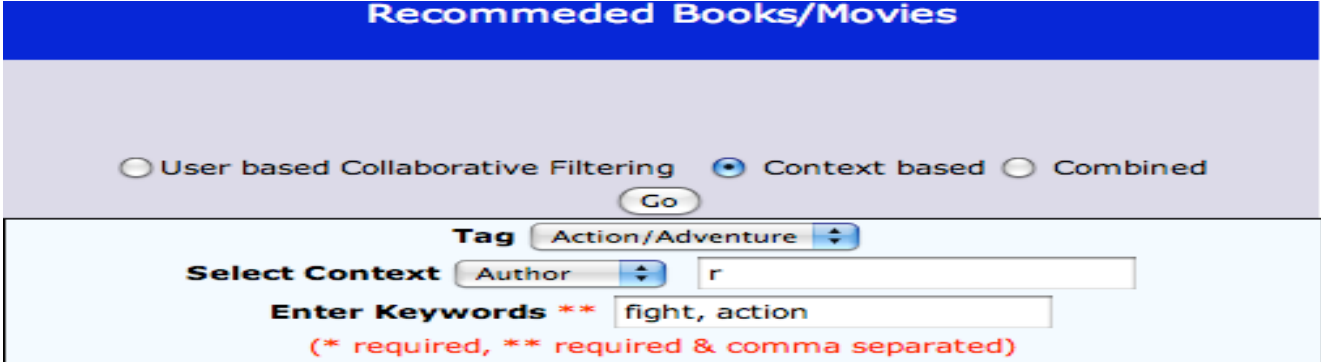


6) Advance Search Books (publisher) results: -



This feature provides the user with advance search capabilities. The user can search under categories author, ISBN, publisher. The above figure displays the books found on category 'publisher' matching keyword 'oxford'

7) Recommendation



The screenshot shows a web interface titled "Recommended Books/Movies". At the top, there are three radio buttons for filtering: "User based Collaborative Filtering", "Context based" (which is selected), and "Combined". Below these is a "Go" button. The main section has a "Tag" dropdown menu set to "Action/Adventure". Underneath, there is a "Select Context" dropdown menu set to "Author" and an adjacent text input field containing the letter "r". Below that is an "Enter Keywords **" text input field containing "fight, action". At the bottom, there is a red note: "(* required, ** required & comma separated)".

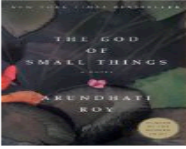
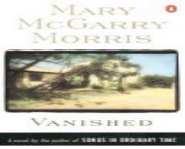



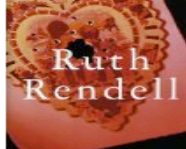
The above figure shows the initial screen shown to the user where the context information is gathered from the user. The active user chooses the tag, selects the parent context category, enters keyword to be searched under the parent context category and finally enters the free keywords, which he/she might be of interested in.

8) Collaborative filtering, recommendation results

Home

Movies/Books Recommendations View Edit

Similar Users using Pearson's Correlation'

User ID	Similarity	Books in Common
243	-0.430665176349	9
<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Title :The God of Small Things</p> </div> <div style="text-align: center;">  <p>Title :Vanished</p> </div> <div style="text-align: center;">  <p>Title :River, Cross My Heart</p> </div> </div>		
User ID	Similarity	Books in Common
244	0.0372648398777	3
<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Title :The God of Small Things</p> </div> <div style="text-align: center;">  <p>Title :Wicked: The Life and Times of the Wicked Witch of the West</p> </div> <div style="text-align: center;">  <p>Title :A Judgement in Stone</p> </div> </div>		

The above figure shows the collaborative filtering engines results. It displays the user id of the similar users, similarity score, books in common, and predictions by them for active user.

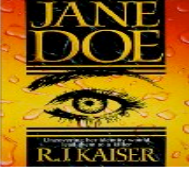
9) Context filtering, recommendation results

Home

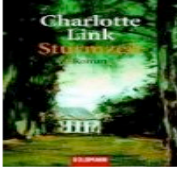
Movies/Books Recommendations [View](#) [Edit](#)

Category name ::Book-Author
 Context ::fight
 fight struggle battle dispute resist oppose
 Context ::action
 action activity operation deed case lawsuit

[refine](#)
[set2](#)

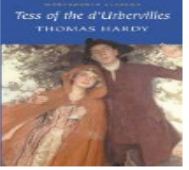


Title :Jane Doe




Title :Sturmzeit. Roman.

Title :Love, Medicine and Miracles



Title :Tess of the D'Urbervilles
(Wordsworth Classics)

Title :El Señor De Los Anillos: LA Comunidad Del Anillo (Lord of the Rings (Spanish))




Title :Corelli's Mandolin : A Novel

The above figure shows the first set of results shown to the users matching the context information provided by the user. Here different senses of the free keywords entered by the user are shown to the user to further refine the recommendation results.

10) Context filtering, recommendation results

In the figure below final results of the context based engine are displayed to the user.

Results for context :: fight



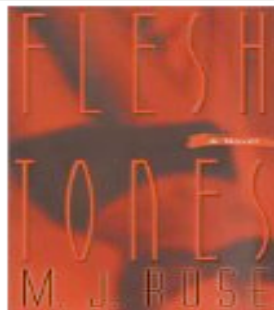
Title :Harry Potter and the Order of the Phoenix (Book 5)

Results for context :: resist



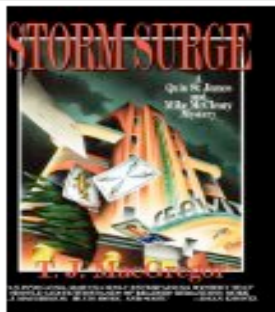
Title :Flesh Tones: A Novel

Results for context :: deed



Title :Flesh Tones: A Novel

Results for context :: case



Title :Storm Surge: A Quin St. James and Mike McCleary Mystery

4.2. Technical Specifications

Content Management System: Drupal 6.6

Languages: PHP 5, Ajax, JavaScript

Database: MySQL 5.x

Server: Apache 2.x.x

Datasets: [MovieLens Data set](#), [Book-Crossing Data set](#)

Screen scrapping websites: <http://www.get-synonym.com/>

4.3. Software Methodology

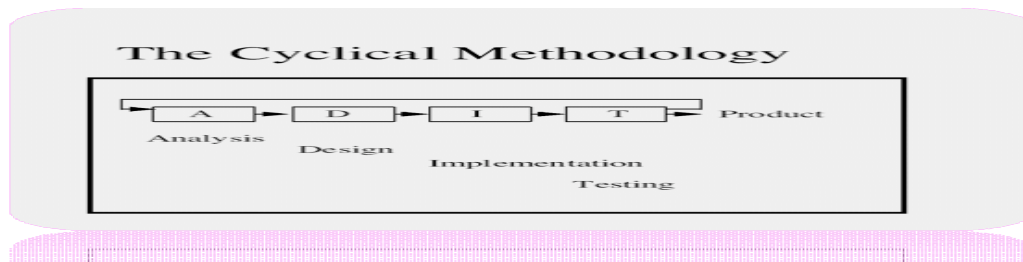


Figure 9: Cyclical methodology (Burback, 1998)

Cyclical methodology is being used for implementation. The system is been generated in an incremental approach by following the various phases shown in the diagram above.

4.4. Web Services

Amazon Web Services

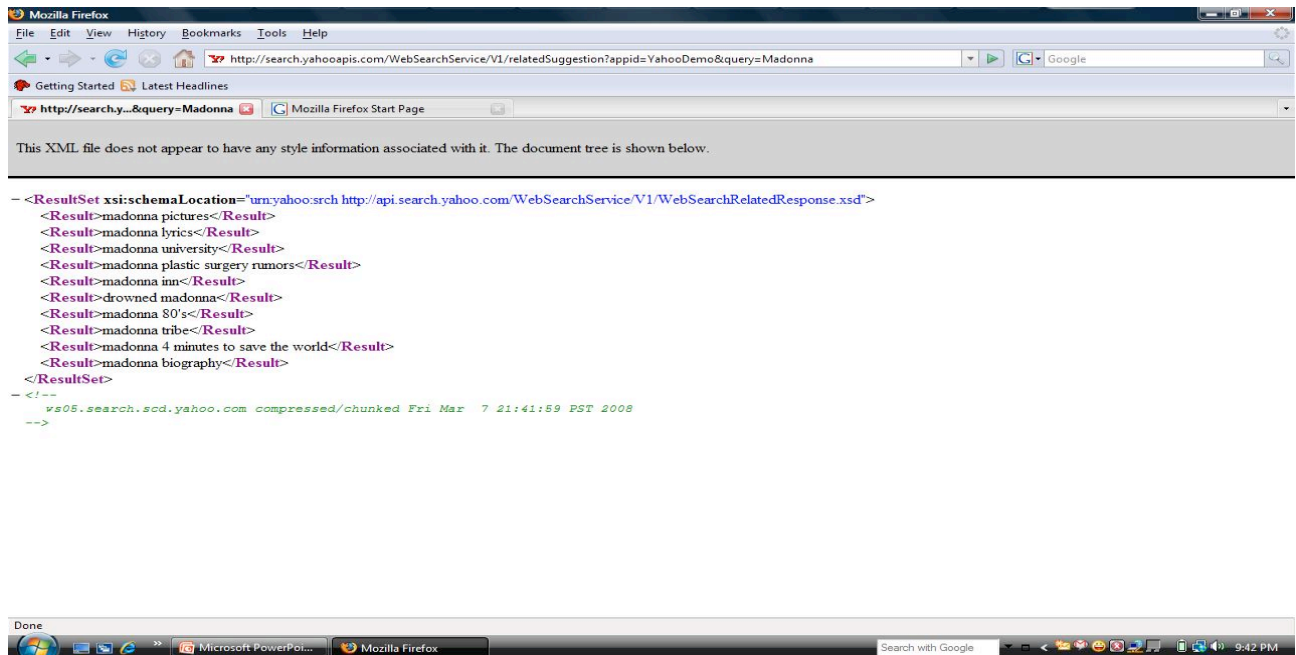
The system makes use of the Amazon Rest (representational state transfer) web service ecs4.0 to fetch metadata about the book.

Yahoo Web Search Services

Allows the user to tap into the Yahoo! Search technologies from other applications. **Related Suggestion/ Term extraction** returns suggested queries to extend the power of a submitted query, providing variations on a theme to help you dig deeper. I tried to make use of yahoo web service in order to get related/main keywords, so that these keywords could be used to search the free tags entered by users. This would help to improve the results of context based engine and in turn would help to provide better recommendations. (Later this approach was dropped and replaced with Screen Scrapping technique discussed after this).

Snapshots of Implementation of Yahoo Related Suggestion

- 1) The php script accepts the keyword 'Madonna' and queries that keyword to Yahoo Web Service, which returns the results of the query as related suggestion keyword.



Screen Scapping

I decided to screen scrap related words based on the tags entered by the user. This would help the system to find output-improved results.

Snapshots of Implementation of Screen Scapping Technique

The website from which data is scrapped:

<http://thesaurus.reference.com/>

<http://www.get-synonym.com/>

Main Entry: fight
Part of Speech: noun¹
Definition: encounter
Synonyms: action, affray, altercation, argument, battle, battle royal*, bout, brawl, broil, brush, clash, combat, conflict, confrontation, contention, contest, controversy, difficulty, disagreement, dispute, dissension, dogfight, duel, engagement, exchange, feud, fisticuffs*, fracas, fray, free-for-all*, fuss, hostility, joust, match, melee, quarrel, riot, rivalry, round, row, ruckus, rumble, scrap*, scrimmage, scuffle, set-to*, skirmish, sparring match, strife, struggle, tiff*, tussle, war, wrangling
Antonyms: retreat, withdrawal
Source: Roget's New Millennium™ Thesaurus, First Edition (v 1.3.1) Copyright © 2008 by Lexico Publishing Group, LLC. All rights reserved. * = informal or slang

Main Entry: fight
Part of Speech: noun²
Definition: courage
Synonyms: aggression, aggressiveness, attack, belligerence, boldness, combativeness, gameness, hardihood, mettle, militancy, pluck, pugnacity, resistance, spirit, will
Antonyms: fear
Source: Roget's New Millennium™ Thesaurus, First Edition (v 1.3.1) Copyright © 2008 by Lexico Publishing Group, LLC. All rights reserved.

Main Entry: fight
Part of Speech: verb¹
Definition: encounter
Synonyms: altercation, assault, attack, bandy with*, battle, bear arms, bicker, box, brawl, brush with*, buck*, challenge, clash, contend, cross swords, dispute, do battle, duel, exchange blows, feud, flare up, grapple, joust, meet, oppugn, ply

Related Ads:

- [Fight Water](#)
- [Fight Like This](#)
- [Fight Video Clip](#)
- [NBA Sports](#)
- [Fight Sports](#)
- [Boxing Fight News](#)

LendingTree

- [Refinance](#)
- [Cash Out Refinance](#)
- [Compare Low Rates](#)
- [Pay Off Debt](#)
- [30-Yr Fixed Rates](#)
- [15-Yr Fixed Rates](#)
- [5-Year ARM](#)

Current Offers from LendingTree®

- Refinance \$400,000 for Only \$2,147/Mo*
- Refinance \$300,000 for Only \$1,810/Mo*

2) The purpose of this php script is to screen scrap synonyms from a website and use it for recommendations. The script captures the first keyword (synonym) in each sense amongst the number of keywords in each sense.

```

Created on Mar 14, 2008
Author : Ankit Khera.
To change the template for this generated file go to
Window - Preferences - PHPeclipse - PHP - Code Templates
*/

/* Function: get_synonym
 * Purpose : retrieves all synonyms from
 * Input : 2 parameter( a word, and web-link http://thesaurus.reference.com/)
 * Output : List of synonyms
 */
function get_synonym($description, $web_url)
{
    $words = array();
    $synonym_url = $web_url . $description;
    $data = file_get_contents($synonym_url);
    $newlines = array("\t", "\n", "\r", "\x20", "\0", "\x0B");
    $content = str_replace($newlines, "", html_entity_decode($data));
    $send=0;
    $count=1;
    echo "<b>Synonyms for ".$description."</b>\n";
    for($i=0;$i<8;$i++)
    {
        $start = strpos($content, "<tr><td align='right' valign='top'><b>Synonyms:'</b>";
        $send = strpos($content, "</td></tr>", $start);
        $table = substr($content, $start, $send-$start);
        $pattern1="/Synonyms:(.*)/"; // extracting string after symbol ':'
        if (preg_match($pattern1, $table, $arr1))
        {
            $send=1;
            $tok=strtok($arr1[1], ",");
            $start=1;
            $start=1;
            $start=1;
            $final_tok=substr($tok, $start, $send-$start);
            $final_tok2=strtok($final_tok, "--");
            $words[$count++]=$final_tok2;
        }
    }
}

```

3) Result



4.5. Testing

The system has been tested by keeping a small set of data from the BX-Crossing dataset aside and then monitor whether the system is able to make correct predictions matching the results available in the set aside database. The system was also tested to see whether the results of context-based engine would match some of the items resulting from the Amazon related books web service which served as a benchmark, besides this engine was tested to see if it gave satisfactory results in scenarios where collaborative filtering engine failed due to less ratings.

5. Advantages of the System

- 1) The System would benefit those users who have to use search engines to locate relevant content. They have to scroll through pages of results to find relevant content.
- 2) Rather than searching for quality web pages, the users of this system would be directly taken to quality web pages matching their personal interests and preferences.
- 3) The system would deliver quality web pages as it is not just dependent on the rating given by other users which could be deceiving at times.

6. Project Schedule/ Deliverables

Schedule

Date	Work Description
Week 1-4	System - Architecture and workflow design.
Week 5-13	<p>Implementation of the Online Recommendation System.</p> <p>a) Developing an algorithm to add items and their descriptions to the system ontology/Taxonomy, extracting features, and maintaining distance scores between items.</p> <p>b) Developing an algorithm to for extracting important keyword,</p>

	<p>from user queries and feedback.</p> <p>c) Developing an algorithm to make use of the Keywords while recommending.</p> <p>d) Integrating/Implementing the system with user based Collaborative-filtering engine findings in cs297</p>
Week 13-14	Analysis and optimization of planned system.
Week 15	Preparing for project defense
Week 16	Project defense

Deliverables

1. A web-based application in which the users may obtain recommended content related to their preferences and interests.
2. CS 298 Final Report

7. Conclusion

In this semester a recommendation system has been implemented based on hybrid approach of collaborative filtering engine and context based engine. The system can be highly improved by making use of caching mechanisms, user clustering which will definitely boost the speed of the system, using yahoo term extraction web service to parse and get important keywords from the feedback provided by the user for an item and utilizing these keywords in context based engine. Further enhancements include storing users past history of results, contexts for future predictions.

References

- Billsus, D., & Pazzani, M. (1998). Learning collaborative information filters. Paper presented at the *Proceedings of the Fifteenth International Conference on Machine Learning*, Madison, Wisconsin, USA. 46-54. Retrieved from <http://portal.acm.org/citation.cfm?coll=GUIDE&dl=GUIDE&id=657311>
- Burback, R. (1998). *A cyclical methodology, introduction*. Retrieved March 10, 2008, from Ronald LeRoi Burback
- Christakou, C., & Stafylopatis, A. (2005). A hybrid movie recommender system based on neural networks. Paper presented at the *5th International Conference on Intelligent Systems Design and Applications*, 500-505. doi:10.1109/ISDA.2005.9
- Droux, S. (2008). *Vogoo - recommendation engine and collaborative filtering*. Retrieved March 30, 2008, from <http://www.vogoo-api.com/>
- E. Peis, J. M. Morales-del-Castillo (2008) Semantic Recommender Systems. Analysis of the state of the topic. Web: www.hipertext.net, ISSN 1695-5498.
- Gediminas Adomavicius, Alexander Tuzhilin (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE, Transactions on Knowledge and Data Engineering*, 734-749.
- GroupLens research*. (2008). Retrieved March 29, 2008, from <http://www.grouplens.org/>
- Klir, G., & Folger, T. (1988). Equivalence and similarity relations. *Fuzzy sets, uncertainty, and information* (1st ed., pp. 82-85). Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Kumar P, Gopalan S, Sridhar V (2005). Context enabled multi-CBR based recommendation engine for e-commerce. IEEE International Conference on e-Business Engineering, 237-255.

Machine learning. (2008). Retrieved March 27, 2008, from http://en.wikipedia.org/wiki/Machine_learning

Olfa Nasraoui, Mrudula Pavuluri (2004). Accurate web recommendations based on profile-specific url-predictor neural networks.. ACM, Proceedings of the 13th international World Wide Web conference on Alternate track papers and posters, 300-301.

Pereira, R., Ricarte, I., & Gomide, F. (2006). Fuzzy relational ontological model in information search systems. In E. Sanchez, & F. Harmelen (Eds.), *Fuzzy logic and the semantic web* (1st ed.,). New York, NY, USA: Elsevier Science Inc. Retrieved from http://books.google.com/books?hl=en&lr=&id=Cidej8b4ESIC&oi=fnd&pg=PA395&dq=%22Pereira%22+%22Fuzzy+Relational+Ontological+Model+in+Information+Search+...%22+&ots=mt7470caQ3&sig=Yx8CZ49DEqnzS7ICaI5D_pDDPko#PPA395,M1

Sean, O. (2008). *Taste collaborative filtering for java*. Retrieved March 26, 2008, from <http://taste.sourceforge.net/>

Stuart E. Middleton, Nigel R. Shadbolt (2004). Ontological user profiling in recommender systems. ACM Transactions on Information Systems, 54-88.

Taste, (2008). Collaborative filtering for Java. Retrieved on August 15, 2008, from <http://taste.sourceforge.net/>

Thesaurus.com. (2008). Retrieved September 10, 2008, from <http://thesaurus.reference.com/>

Toby Segaran (2007). Programming Collective Intelligence: Building Smart Web 2.0 Applications, Page11

Yahoo developer network. (2008). Retrieved April 1, 2008, from <http://developer.yahoo.com/>