Personalized Book Retrieval System Using Amazon-LibraryThing Collection

A thesis
Submitted to the faculty of the Graduate school
of the University of Minnesota
by

VenkataRaviKiran Ravva

In partial fulfillment of the requirements
for the degree of
Master of Science

Dr. Carolyn J. Crouch

August 2014

Acknowledgements

I would like to take this opportunity to thank Dr. Carolyn Crouch and Dr. Donald Crouch for providing chance to work in the field of Information Retrieval (IR), constant support and guidance throughout my thesis work. I also like to thank Dr. Ted Pedersen for offering Advanced Natural Language Processing course from which I got some of the ideas to complete my thesis work. I also like to thank Dr. Marshall Hampton for being a member of my thesis committee.

I would like to thank Mr. Sai Subramanyam Chittila and Mr. Kiran kumar Bushireddy for their guidance in the initial days of my thesis work.

I would also like to thank my fellow students Vamshi Krishna Thotempudi and Lavanya Singampalli for their invaluable critics of my work and help with the experiments.

To Jim Luttenin, Lori Lucia, Clare ford, Trisha O'Keefe, and Anna Gilmore, thank you so much for assisting me with all the tasks required to complete my masters program. I would also like to thank all my seniors and juniors, and friends here in Duluth who made my stay in Duluth unforgettable.

Last, but not the least, I am grateful to the UMD Graduate School and the Director of Graduate Studies, Dr. Carolyn Crouch, for the financial support provided to me in support of my research through the Quality Metrics Allocation/Chancellor's Graduate Fellowships.

Abstract

Information retrieval is the science of retrieving documents or information from a corpus based on the need of user. Selecting a book from a collection of available books based on its topical relevance to the query may not give us the "best" (or all the "best") such book(s). However, including social data, such as popularity, reviews and ratings, may improve the results. So we include social data with book metadata for this purpose.

The main goal of this research is to provide a book retrieval system for the Social Book Search (SBS) Track of the INEX forum. For the SBS track, participants are provided with an XML collection of data from Amazon and LibraryThing (LT) forum, a set of topics from the LT forum enriched with user catalogue data (i.e., books that the topic creator has in his LibraryThing personal catalogue), and anonymous user profiles. Participants must devise a system which provides the ISBN/work IDs of the books which are relevant to the topic creator. For this purpose, we designed a recommender system which provides personalized search results.

Contents

List of Tables

List of Figures

List of Equations

1 Introduction

Information retrieval is the science of retrieving documents or information from a corpus based on the need of user. When people like to know about something, they generally get information from books because books are still prominent sources of information. Selecting the best book from available books based on the relevance of the book to the user’s query may not be sufficient, because reading all books which are related to the topic may be infeasible. So we can say that taking only topical relevance into consideration in providing the relevant books may not be the most effective process. Considering other information, such as popularity, reviews and ratings, may provide better results than topical relevance. As Web 2.0 applications are providing a wealth of information related to the above-mentioned features, we can use them in providing a better choice of books for user needs.

Extensible Markup Language (XML) [29] is a markup language used to represent information in structured format with characteristics such as simplicity, generality and usability. The entire web is designed based on hypertext, which is very similar to XML. It is now possible to represent web documents in terms of their components, which we call XML *elements*. XML is a common method for representing web documents and retrieving information in terms of XML elements is now feasible.

INEX, the *Initiative for Evaluation of XML retrieval* [1], was established in 2002 to measure the performance of XML retrieval systems. INEX provides a large document collection, queries, and uniform evaluation metrics. All the participants use this collection and query set in testing the design and implementation of their algorithms. The evaluation measures provided by INEX serve as a common measure to compare the performance of participants [1]. Each year INEX provides a set of tracks, such as the Social Book Search, Tweet Contextualization, Snippet Retrieval and Linked Data tracks, from which each participant may choose one or more tracks.

The Main goal of this research is to provide a book search system for the Social Book Search (SBS) Track [5] of the INEX forum. For the SBS track, participants are provided with an XML collection of data from Amazon and the LibraryThing (LT) forum, a set of topics collected from LT forum enriched with user catalogue data (i.e., books that the topic creator has in his LibraryThing personal catalogue), and anonymised user profiles. Participants must devise a system which provides the ISBN/work IDs of books which are not only relevant to that particular topic but also relevant to that topic creator [5].

We use Indri [10] to generate a ranked list of books from both traditional metadata and user-generated data. Traditional meta data and user-generated data (social data) are selected from the XML files provided by INEX. Traditional data has been collected by INEX from the Amazon website, Library of Congress (LOC), British Library, and Dewey Decimal Codes (DDC) [19], and Social data was collected by INEX from Amazon reviews and User Tags in LT. The Main focus of the 2014 SBS track is to provide personalized results based on user profile data. For this purpose, we design a recommender system which provides personalized search results.

The details of the SBS Track, evaluation metrics, collection format and submission format are described in Chapter 2, implementation of retrieval and recommender systems are described in Chapter 3, experiments and results are discussed and analyzed in Chapter 4 and conclusions with suggestions for future research are presented in Chapter 5.

2 Background

This chapter provides background for our research, including a description of the two basic retrieval models and the SBS track history, collection format, and evaluation metrics.

2.1 Retrieval Models

With the rapid growth of the data available on Internet, the need for IR techniques is also evolving rapidly. To retrieve books, we depend upon standard retrieval models.

2.1.1 The Vector Space Model

The Vector Space Model (VSM) [21] is the basic, traditional model of information retrieval. It represents documents and queries as vectors in n-dimensional space. Each document D_i is represented as term vector $(t_{1,i},t_{2,i},t_{3,i},t_{4,i}\ldots\ldots,t_{n,i})$ where $t_{n,i}$ represents the frequency of term t_n in document D_i. Each query is also represented as a vector of terms, $(t_{1,i},t_{2,i},t_{3,i},t_{4,i}\ldots\ldots,t_{m,i})$, where $t_{m,i}$ represents the frequency of term t_m in query Q_i. Common measures of similarity, such as cosine, inner product, etc., are used to calculate the distance between vectors. The distance between the query vector and the document vector in n-dimensional space indicates how closely a given document correlates with a given document vector.

Smart [2] is an information retrieval system based on the VSM. It creates an index for both documents and queries using term frequency. Smart allows re-weighting of the term vectors by choosing appropriate weighting schemes such as *Lnu–ltu* [3] for the retrieval process. In this method, document vectors are converted to *Lnu* vectors, and query vectors are converted to *ltu* vectors. After term weighting, Smart produces ranked list of document vectors for each query, based on the similarity between vectors. *Lnu-ltu* utilizes pivoted document normalization to avoid biasing the results towards longer documents.

2.1.2 The Language Model

Language modeling is based on the probabilistic estimation of linguistic units such as words, sentences, queries, etc. Language modeling is a common NLP technique used in the noisy channel model-based applications such as statistical machine translation, speech recognition and document classification. The Language modeling approach to IR was proposed in [15].

The basic idea behind the language modeling approach to IR is: given query Q and document D, what is the probability (P(D/Q)) that document D will be retrieved given by query Q? By applying Bayes theorem, this probability is changed to: What is the probability that query Q is generated from Document D P(Q/D)? The language model computes this value, P(Q/D), to retrieve a set of documents for a particular query. This is the model used in Indri [10], which we use for document retrieval. Chapter 3 gives a more detailed explanation of Indri and related information.

2.2 The Social Book Search Track at INEX

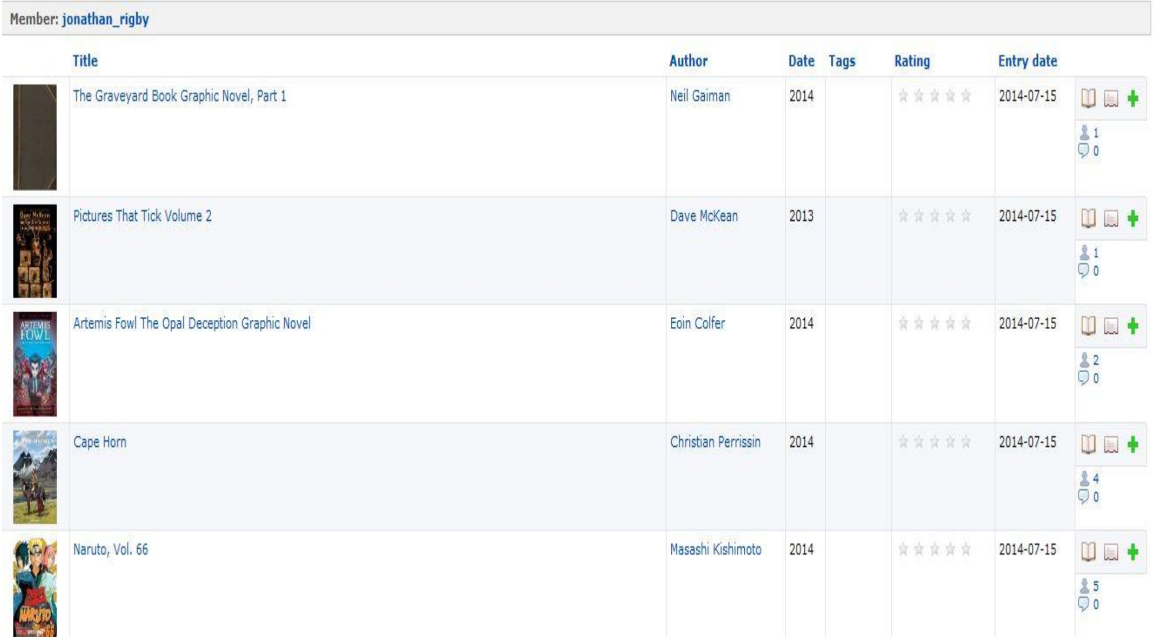
The INEX Social Book Search Track [5] started in 2011. The main aim of this track is to investigate techniques to support users in searching and navigating professional metadata and user-generated content from social media based on their profiles [5].

LibraryThing [16] is an online service which helps people in cataloging their books. The user can post queries for suggestions about books on this site. Any user can suggest books based on his/her opinion.

LibraryThing assigns a number (work ID) to each book based on its ISBN number. A mapping from ISBN to work ID is provided by INEX in plain text format. The catalogue of a user contains for each book listed tags (specific to the particular book) and the corresponding work IDs. Table 1 shows a sample mapping file, and Figure 1 shows a typical user catalogue.

ISBN	work ID
0030843278	6
0675076455	7
1582099855	14
0681047992	14
0843111577	16
0440428130	17
0330308297	17

Table 1: A Sample ISBN to work ID Mapping



Member: jonathan_rigby

	Title	Author	Date	Tags	Rating	Entry date	
	The Graveyard Book Graphic Novel, Part 1	Neil Gaiman	2014		☆☆☆☆☆	2014-07-15	1 0
	Pictures That Tick Volume 2	Dave McKean	2013		☆☆☆☆☆	2014-07-15	1 0
	Artemis Fowl The Opal Deception Graphic Novel	Eoin Colfer	2014		☆☆☆☆☆	2014-07-15	2 0
	Cape Horn	Christian Perrissin	2014		☆☆☆☆☆	2014-07-15	4 0
	Naruto, Vol. 66	Masashi Kishimoto	2014		☆☆☆☆☆	2014-07-15	5 0

Figure 1: A Typical User Catalogue

Amazon [17] is an e-commerce website where a user can order things online and get them through postal/courier service. At this site, users can enter reviews and provide ratings for the products they purchase from Amazon. For books, Amazon also provides data from editorial reviews and similar books based on its own recommender system. INEX collected data from both Amazon and LibraryThing and provided it as input for the SBS track participants in XML format. It also provided topics for this track from the LibraryThing forum. It evaluates results submitted by participants using the specified metrics [13].

2.2.1 History of the SBS Track

In 2011, the Book Search Task originated under the name of Social Search for Best Books (SB). INEX provided a collection of 2.8 million records from Amazon books and LibraryThing (LT) for this track. The aim of this task was to retrieve a ranked list of books based on their metadata (author, publisher, and title) and social metadata (Amazon user reviews, ratings and LibraryThing tags), i.e., to evaluate the use of user-generated metadata such as reviews and tags in addition to traditional publisher metadata. Results were to consist of recommended books for each topic in the order of rank. 211 topics were provided by INEX in 2011. Evaluations were performed using relevance judgments from both Amazon Mechanical Turks (AMT) [30] and LibraryThing (LT). The specific evaluation metrics were nDCG@10, P@10, MRR and MAP.

In 2012, the SB task was changed to the Social Book Search Task (SBS). This task uses the 2011 collection, but some new tags were added. The data contains user profiles which could be used to filter books based on that profile. This task investigates the value of user information and both traditional and user-generated book metadata in retrieval. The 2011 evaluation metrics are retained this year. The query set consists of 300 topics.

The 2013 SBS task is the same as in 2012 except there are 386 topics which contain the mediated query written by Amazon Mechanical Turks along with the normal topic information (narrative, group, member and title).

In 2014, INEX provides two tasks in the SBS track. One is a system-oriented task as the Recommender Task and the other is a user-oriented task named the Interactive Task. The Recommendation Task is similar to the 2013 SBS Task. In 2014, topics are provided with user profiles (books the topic creator had in his/her catalogue at the time of creation). In order to run recommendation experiments, INEX also provides a set of anonymous profiles from other LT forum members. In 2014, INEX uses recall (R@1000) as one of the evaluation metric rather than precision (P@10).

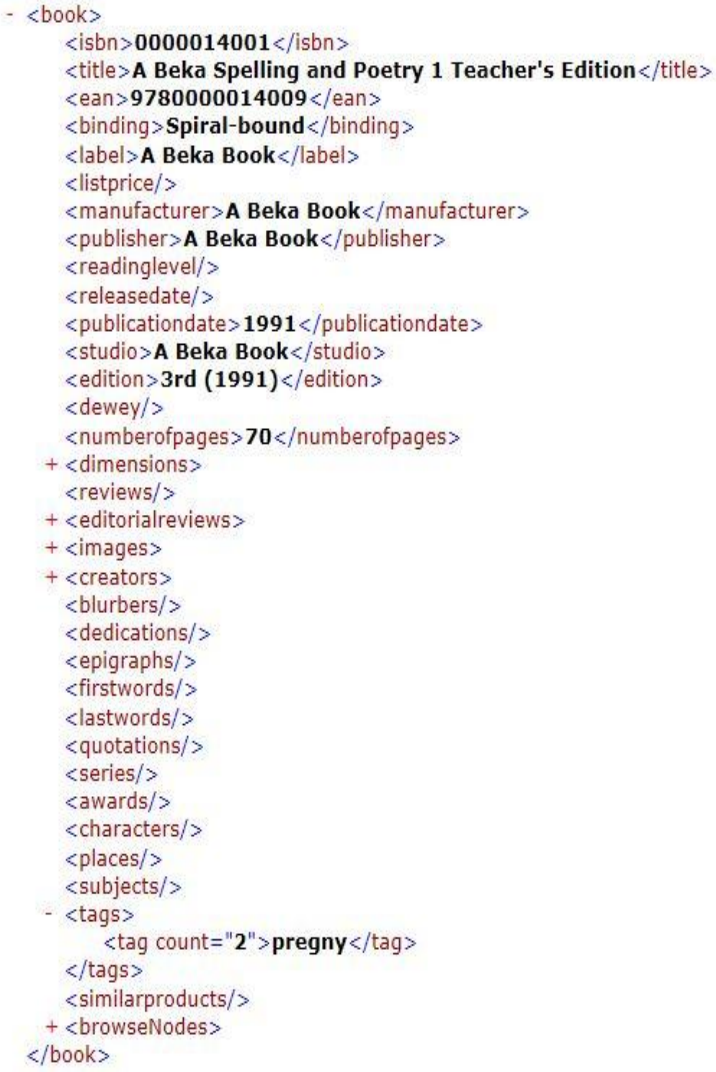
2.2.2 The Collection

Recall that the collection is almost unchanged from 2011 to 2014. The document collection for the INEX SBS track was provided by taking content from the Amazon and LibraryThing websites. It is in structured XML format. Each ISBN of a book available on Amazon is taken as a single XML file. Content about a book on Amazon, such as reviews, ratings, and price, are combined with tags, blurbers, and epigraphs in LibraryThing and provided as an XML file. These files also contain Dewey Decimal Codes (DDC), data from the Library of Congress (LOC) and the British Library (BL) identified by separate XML tags [9]. Each XML file follows a specific Document Type Definition (DTD) [5]. Figure 2 shows the DTD for a typical XML file. Figure 3 shows the structure of a typical XML file.

```
<!ELEMENT book (isbn, title, ean, binding, label, listprice, manufacturer,
publisher, readinglevel, releasedate, publicationdate, studio, edition, dewey,
numberofpages, dimensions, reviews, editorialreviews, images, creators,
blurbers, dedications, epigraphs, firstwords, lastwords, quotations, series,
awards, characters, places, subjects, tags, similarproducts, browseNodes)>
<!ELEMENT dimensions (height?, width?, length?, weight?)>
<!ELEMENT reviews (review*)>
<!ELEMENT review (authorid?, date, summary?, content?, rating, totalvotes,
helpfulvotes)>
<!ELEMENT editorialreviews (editorialreview*)>
<!ELEMENT editorialreview (source, content?)>
<!ELEMENT images (image*)>
<!ELEMENT image (url, height?, width?, imageCategories)>
<!ELEMENT imageCategories (imagecategory*)>
<!ELEMENT creators (creator*)>
<!ELEMENT blurbers (blurber*)>
<!ELEMENT dedications (dedication*)>
<!ELEMENT epigraphs (epigraph*)>
<!ELEMENT firstwords (firstwordsitem*)>
<!ELEMENT lastwords (lastwordsitem*)>
<!ELEMENT quotations (quotation*)>
<!ELEMENT series (seriesitem*)>
<!ELEMENT awards (award*)>
<!ELEMENT browseNodes (browseNode*)>
<!ELEMENT characters (character*)>
<!ELEMENT places (place*)>
<!ELEMENT subjects (subject*)>
<!ELEMENT similarproducts (similarproduct*)>
<!ELEMENT tags (tag*)>
<!ELEMENT isbn (#PCDATA)>
<!ELEMENT ean (#PCDATA)>
<!ELEMENT binding (#PCDATA)>
<!ELEMENT label (#PCDATA)>
<!ELEMENT listprice (#PCDATA)>
<!ELEMENT manufacturer (#PCDATA)>
<!ELEMENT numberofpages (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT height (#PCDATA)>
<!ELEMENT width (#PCDATA)>
<!ELEMENT length (#PCDATA)>
<!ELEMENT weight (#PCDATA)>
<!ELEMENT readinglevel (#PCDATA)>
<!ELEMENT releasedate (#PCDATA)>
<!ELEMENT publicationdate (#PCDATA)>
```

```
<!ELEMENT studio (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT edition (#PCDATA)>
<!ELEMENT dewey (#PCDATA)>
<!ELEMENT creator (name, role)>
<!ELEMENT rating (#PCDATA)>
<!ELEMENT authorid (#PCDATA)>
<!ELEMENT totalvotes (#PCDATA)>
<!ELEMENT helpfulvotes (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT summary (#PCDATA)>
<!ELEMENT content (#PCDATA)>
<!ELEMENT source (#PCDATA)>
<!ELEMENT url (#PCDATA)>
<!ELEMENT data (#PCDATA)>
<!ELEMENT imagecategory (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT role (#PCDATA)>
<!ELEMENT blurber (#PCDATA)>
<!ELEMENT dedication (#PCDATA)>
<!ELEMENT epigraph (#PCDATA)>
<!ELEMENT firstwordsitem (#PCDATA)>
<!ELEMENT lastwordsitem (#PCDATA)>
<!ELEMENT quotation (#PCDATA)>
<!ELEMENT seriesitem (#PCDATA)>
<!ELEMENT award (#PCDATA)>
<!ELEMENT browseNode (#PCDATA)>
<!ATTLIST browseNode id CDATA #REQUIRED>
<!ELEMENT character (#PCDATA)>
<!ELEMENT place (#PCDATA)>
<!ELEMENT subject (#PCDATA)>
<!ELEMENT similarproduct (#PCDATA)>
<!ELEMENT tag (#PCDATA)>
 <!ATTLIST tag count CDATA #REQUIRED>
```

Figure 2: DTD for Amazon-LT XML File



```
- <book>
    <isbn>0000014001</isbn>
    <title>A Beka Spelling and Poetry 1 Teacher's Edition</title>
    <ean>9780000014009</ean>
    <binding>Spiral-bound</binding>
    <label>A Beka Book</label>
    <listprice/>
    <manufacturer>A Beka Book</manufacturer>
    <publisher>A Beka Book</publisher>
    <readinglevel/>
    <releasedate/>
    <publicationdate>1991</publicationdate>
    <studio>A Beka Book</studio>
    <edition>3rd (1991)</edition>
    <dewey/>
    <numberofpages>70</numberofpages>
  + <dimensions>
    <reviews/>
  + <editorialreviews>
  + <images>
  + <creators>
    <blurbers/>
    <dedications/>
    <epigraphs/>
    <firstwords/>
    <lastwords/>
    <quotations/>
    <series/>
    <awards/>
    <characters/>
    <places/>
    <subjects/>
  - <tags>
        <tag count="2">pregny</tag>
    </tags>
    <similarproducts/>
  + <browseNodes>
  </book>
```

Figure 3: An XML File from the Amazon-LT Collection

2.2.3 Submission Format

All participants of the SBS track must follow a specific format when submitting their results to INEX. Each participant can submit at most 1000 results per each topic. The format of submission is shown in Table 2.

2.2.4 QRels

Each topic is evaluated against the set of QRels provided by INEX, based on the evaluation metrics of 2.2.5. These QRels are selected from answers in LibraryThing forums for that topic. Relevance scores are based on the algorithm given in [5]. Table 3 shows a sample QRels file.

Topic ID	N/A	Work ID	Rank	Relevance Score	Run name
100635	Q0	6050	1	-4.85389964380361	UMD_2014_SBS_Indri_dir_2500
100635	Q0	97880	2	-4.9433015783599	UMD_2014_SBS_Indri_dir_2500
100635	Q0	1489	3	-5.18384170853538	UMD_2014_SBS_Indri_dir_2500
100635	Q0	24308	4	-5.26403535359177	UMD_2014_SBS_Indri_dir_2500
100635	Q0	1061783	5	-5.2974131585874	UMD_2014_SBS_Indri_dir_2500

Table 2: Submission Format for the SBS track

Topic ID	N/A	Work ID	Relevance Score
1116	0	135255	4
1116	0	13008088	0
1116	0	135088	4
1116	0	1044275	4
1116	0	24048	0
1116	0	2300468	4
1116	0	195721	6

Table 3: Sample QRels

2.2.5 Evaluation Metrics

All results submitted by participants of the SBS track are evaluated against four metrics. Among all the metrics, nDCG@10 is used as the measure to rank participants.

Normalized discounted cumulative gain (nDCG) is a measure based on graded relevance of the results. It varies from 0.0 to 1.0, with 1.0

representing ideal results. This metric is commonly used in evaluating web search engines. See in [13] for details. Equation 1 provides the formula for calculating nDCG.

Cumulative Gain CG[i] = $\begin{cases} G[i] & if\ i = 1 \\ CG[i-1] + G[i], & otherwise. \end{cases}$

Discounted Cumulative Gain DCG[i] = $\begin{cases} CG[i] & if\ i < b \\ DCG[i-1] + \frac{G[i]}{log_b(i)} & if\ i \geq b \end{cases}$

Idealized DCG = DCG value after sorting the gain values

Normalized Discounted Cumulative Gain nDCG = $\frac{DCG}{IDCG}$

Equation 1: Normalized Discounted Cumulative Gain (nDCG)

Mean Reciprocal Rank (MRR) is a measure for evaluating systems which produces rank-ordered results to queries. Reciprocal rank of a query is the multiplicative inverse of the rank of the first correct result. MRR is average of reciprocal ranks of all queries. Equation 2 provides formula for calculating MRR value.

Mean Reciprocal Rank (MRR) = $\frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$

rank_{i} = rank of the correct first answer

Equation 2: Mean Reciprocal Rank (MRR)

Precision (P@10) and Recall (R@1000) are general measures of any IR system which are based on the number of relevant documents and the number of retrieved documents. Precision is the fraction of retrieved results which are relevant and recall is fraction of relevant results which are retrieved.

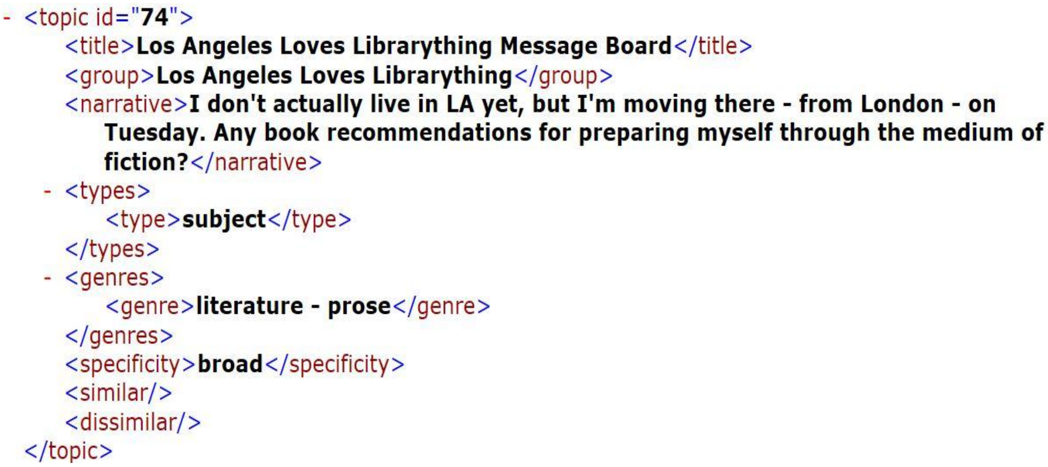
Mean averaged precision (MAP) is the mean of the average precision values for each query.

2.3 INEX 2011 SBS Track Topics

This research mainly focuses on the 2011 traditional retrieval system and 2014 recommender system. The 2011 topic set for the SBS track is taken from the LibraryThing forums where a user can post his/her request and a brief description of his/her requirement. These topics are selected by INEX and are provided in XML form. Each topic contains the title of the topic, group in which it was posted on the LibraryThing forum, a description as narrative in XML and genre data. For corresponding information in 2013 and 2014, see [11, 12]. Figure 4 is a sample topic taken from the 2011 topic set, and Figure 5 represents the LibraryThing forum post for this particular topic.

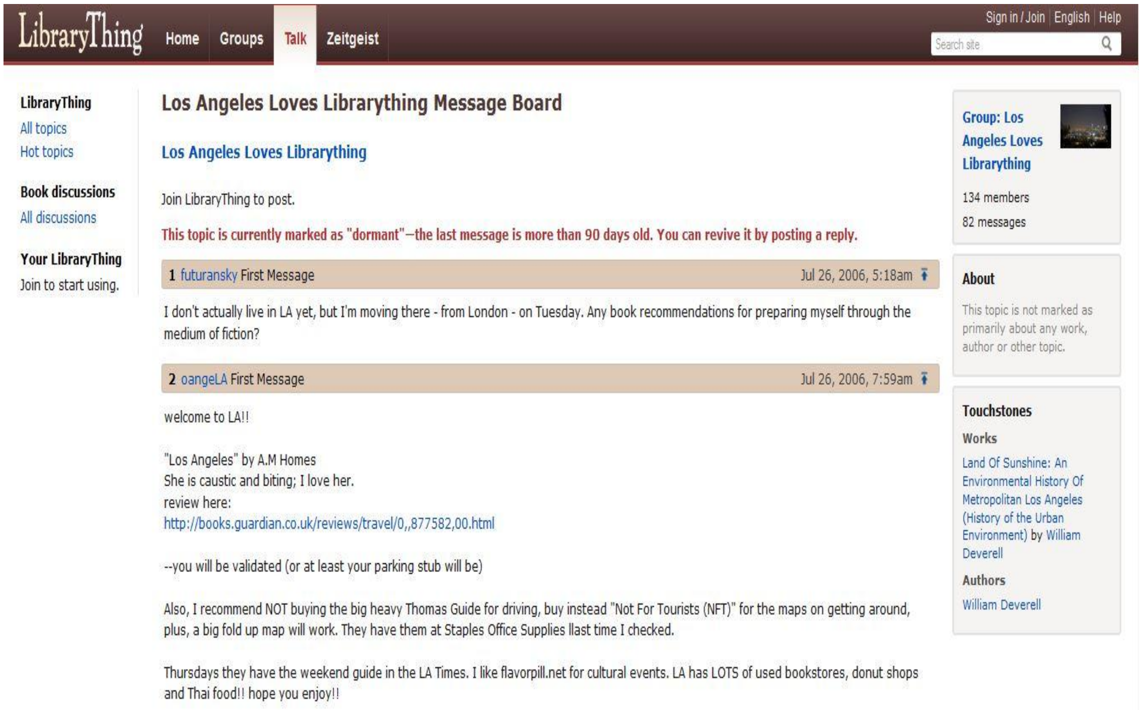
2.4 Related Work

We started working on this track in 2014. Significant work was done earlier by other participants. We referred in particular to the work done by the University of Amsterdam [8, 14, 9] and the Royal School of Library and Information Science (RSLIS) in Denmark [6, 7, 9] when starting work on the SBS track.



```
- <topic id="74">
    <title>Los Angeles Loves Librarything Message Board</title>
    <group>Los Angeles Loves Librarything</group>
    <narrative>I don't actually live in LA yet, but I'm moving there - from London - on
       Tuesday. Any book recommendations for preparing myself through the medium of
       fiction?</narrative>
  - <types>
       <type>subject</type>
    </types>
  - <genres>
       <genre>literature - prose</genre>
    </genres>
    <specificity>broad</specificity>
    <similar/>
    <dissimilar/>
  </topic>
```

Figure 4: A Sample from the 2011 SBS Topic Set



LibraryThing
All topics
Hot topics

Book discussions
All discussions

Your LibraryThing
Join to start using.

Los Angeles Loves Librarything Message Board

Los Angeles Loves Librarything

Join LibraryThing to post.

This topic is currently marked as "dormant"—the last message is more than 90 days old. You can revive it by posting a reply.

1 futuransky First Message — Jul 26, 2006, 5:18am

I don't actually live in LA yet, but I'm moving there - from London - on Tuesday. Any book recommendations for preparing myself through the medium of fiction?

2 oangeLA First Message — Jul 26, 2006, 7:59am

welcome to LA!!

"Los Angeles" by A.M Homes
She is caustic and biting; I love her.
review here:
http://books.guardian.co.uk/reviews/travel/0,,877582,00.html

--you will be validated (or at least your parking stub will be)

Also, I recommend NOT buying the big heavy Thomas Guide for driving, buy instead "Not For Tourists (NFT)" for the maps on getting around, plus, a big fold up map will work. They have them at Staples Office Supplies llast time I checked.

Thursdays they have the weekend guide in the LA Times. I like flavorpill.net for cultural events. LA has LOTS of used bookstores, donut shops and Thai food!! hope you enjoy!!

Group: Los Angeles Loves Librarything
134 members
82 messages

About
This topic is not marked as primarily about any work, author or other topic.

Touchstones
Works
Land Of Sunshine: An Environmental History Of Metropolitan Los Angeles (History of the Urban Environment) by William Deverell
Authors
William Deverell

Figure 5: A Snapshot from LibraryThing Website Forum

3 Implementation

Our SBS system uses Indri for the initial retrieval of ISBNs for a particular topic. This chapter gives an overview of Indri and SBS system architecture.

3.1 Indri

Indri [10] is a retrieval system which combines Bayesian inference networks and language modeling in information retrieval. It is a part of the Lemur [22] project at the University of Massachusetts Amherst and the Language Technologies Institute (LTI) at Carnegie Mellon University.

Indri is an open source tool which provides variety of capabilities e.g., (indexing and retrieval of documents, field and passage retrieval). It parses documents in PDF, XML, HTML, and TREC formats. It also supports INQUERY, which is popular structured query language. Indri supports UTF-8 encoded text. It provides an API which can be used from C++, JAVA and PHP. It can also be used on clusters of machines for faster indexing and retrieval.

Using the *dumpindex* command, we can a look at document vectors and inverted lists of Indri indices. The Smart stop-list [18] and Krovetz stemmer [23] are used in document processing.

Smoothing is a method used to overcome both the *zero probability* and *data sparseness* problem. Indri supports a variety of smoothing functions, such as Jelinek-Mercer [24], Dirichlet [24] and Two-Stage smoothing [25]. We use Dirchlet smoothing as suggested by [24] for concise title queries. Indri also supports pseudo feedback using language modeling [26].

3.2 Literature review

The University of Amsterdam (UAms) first participated in SBS track in 2011 [8]. They experimented with different indexes (Amazon, LT, Professional, Title, Social, and Full), combinations of XML data, and topic field combinations. They used Indri with Dirchlet smoothing for this purpose. The Royal School of Library and Information Science (RSLIS) used six different indices (metadata, content, controlled metadata, tags, reviews [book centric, review centric] and all fields) [6]. They also used Indri with Jelinik-Mercer smoothing.

In 2012, UAms used quality and quantity priors for retrieval purposes [14]. They submitted results by considering Bayesian average rating and neighborhood-based and model-based collaborative filtering methodologies. They crawled user profiles from LibraryThing for this purpose. RSLIS implemented social re-ranking of initial results using user ratings, Amazon similar products, tags and authors [7]. In 2013, both UAms and RSLIS used the mediated query, which was new that year [9].

3.3 INEX 2011 – Social Book Search Track

To develop a base for our 2014 system, we first designed and implemented a system for the 2011 SBS track. This section describes its architecture.

In 2011, INEX provided the Amazon-LibraryThing collection, which consists of 2.8 million book records from Amazon along with data from LibraryThing. INEX also provided 211 topics collected from LT forum discussions. INEX used 2 types of QRels; one is based on answers from the LibraryThing forum (touchstone recommendations) and the other on crowd sourcing judgments provided by Amazon Mechanical Turks (AMT). As each book may have different editions and different editions have different ISBN numbers, INEX provided a mapping between ISBNs and work IDs.

3.3.1 Architecture

Our 2011 SBS system used Indri to index the corpus and retrieve relevant documents. Figure 6 shows its architecture with a workflow of steps from the processing of the corpus to retrieval of results.

3.3.2 Scrubbing

Scrubbing is the process of removing unwanted characters and data from the corpus. Figure 7 identifies the Xpaths of XML nodes which were removed during scrubbing. We also removed special characters from the content of each tag. We used the *libxml* parser which is available in Perl for this purpose.

3.3.3 Parsing

We created six different parses from the remaining XML nodes after scrubbing, namely, Title, Official/Professional, Social, LT, Amazon, and Full parses. Table 4 identifies Xpaths of nodes and the flag (yes/no) which determines whether the given XML node is included in that particular parse or not.

Some tags (e.g., numeric values, similar products and browse nodes) are not included in the parsing process because we felt they would not affect the retrieved results. We count the number of times a tag occurs (say n) with respect to a book by repeating that tag in the parse n times. We included the description of the Dewey decimal classes (DDC) while parsing the scrubbed corpus.

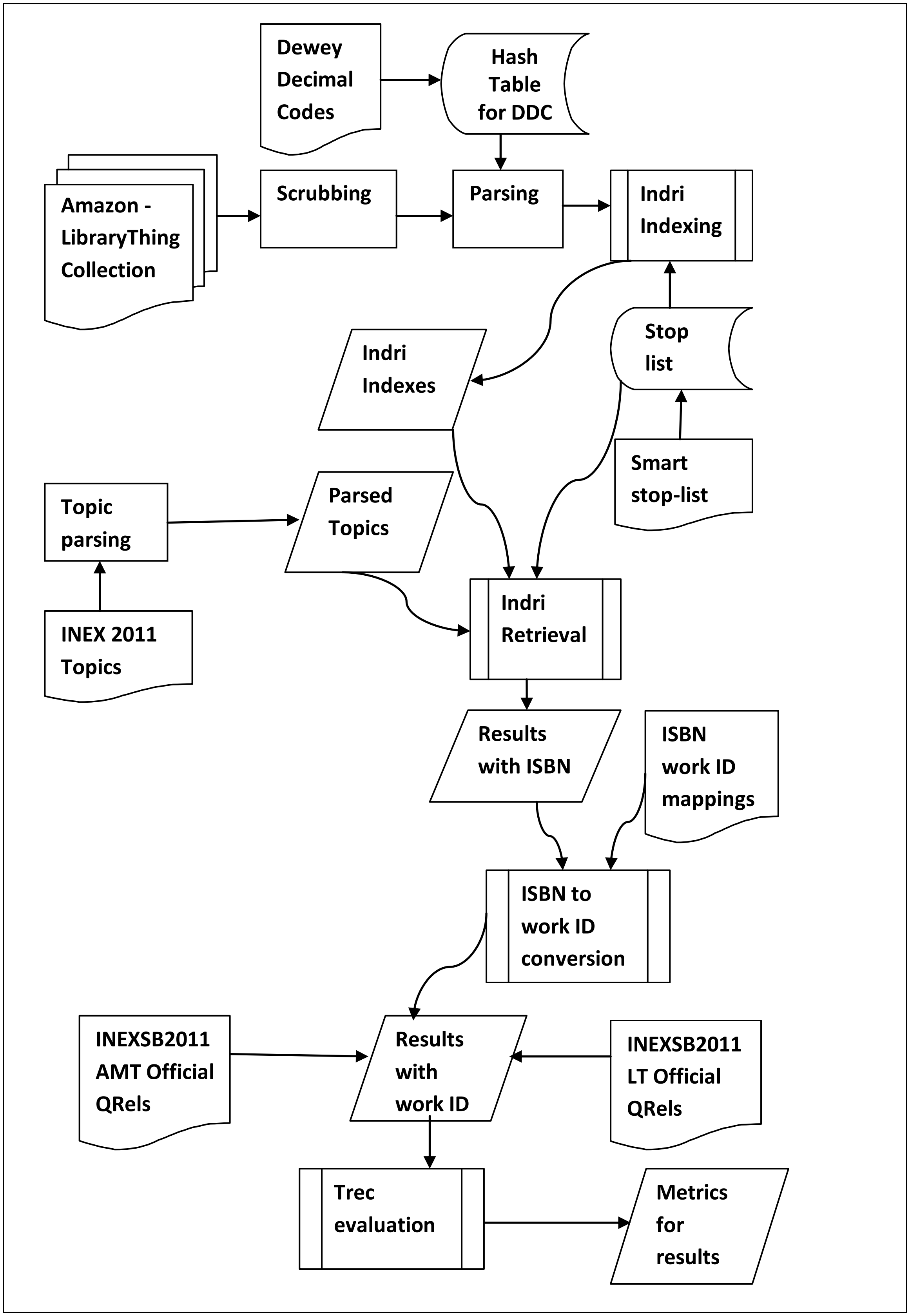


Figure 6: Architecture of 2011 SBS Retrieval System

```
/book/dimensions
/book/images
/book/dedications
/book/studio
/book/binding
/book/listprice
/book/label
/book/edition
/book/ean
/book/manufacturer
/book/numberofpages
/book/readinglevel
/book/publicationdate
/book/authorid
/book/creators/creator/role
/book/creators/creator/releasedate
/book/reviews/review/authorid
/book/reviews/review/date
```

Figure 7: Xpaths of XML Nodes Removed from Corpus

Topics are provided in XML format. Each topic is converted to Indri format as shown in Figure 8. We considered only the title field as the topic statement. We created indices of each parse using Indri, the Smart stop-list, and Krovetz stemmer.

3.3.4 Retrieving Results

We retrieved results using Indri with Dirchlet smoothing parameter μ at 2500 (default value). We retrieved the top 1000 results for each topic; ISBNs are converted to work IDs using the mappings provided by INEX. Figure 9 shows the format of the result.

XPath	Title	Offi/ Prof	Soc	LT	Amazon	Full
/book/title	Yes	Yes	No	Yes	Yes	Yes
/book/publisher	Yes	Yes	No	No	Yes	Yes
/book/dewey	No	Yes	No	No	No	Yes
/book/editorialreviews/editorialreview/source	No	No	Yes	No	Yes	Yes
/book/editorialreviews/editorialreview/content	No	No	Yes	No	Yes	Yes
/book/creators/creator/name	Yes	Yes	No	No	No	Yes
/book/reviews/review/summary	No	No	Yes	No	Yes	Yes
/book/reviews/review/content	No	No	Yes	No	Yes	Yes
/book/blurbers/blurber	No	No	Yes	Yes	No	Yes
/book/epigraphs/epigraph	No	No	Yes	Yes	No	Yes
/book/firstwords/firstwordsitem	No	No	No	No	No	Yes
/book/lastwords/lastwordsitem	No	No	No	No	No	Yes
/book/quotations/quotation	No	No	Yes	Yes	No	Yes
/book/series/seriesitem	No	No	No	No	No	Yes
/book/awards/awarditem	No	No	No	No	No	Yes
/book/characters/character	No	No	No	No	No	Yes
/book/places/place	No	No	No	No	No	Yes
/book/subjects/subject	No	Yes	No	No	No	Yes
/book/tags/tag	No	No	Yes	Yes	No	Yes

Table 4: Inclusion of Nodes in Specified Parses

```
<parameters>
  <query>
    <type>indri</type>
    <number>1116</number>
    <text>Which LISP</text>
  </query>
<parameters>
```

Figure 8: Indri Query Format

```
<result>
  <file>ISBN number</file>
  <path>/[0]</path>
  <rsv> -9.65336</rsv>
</result>
```

Figure 9: Sample from Indri Results

3.4 INEX 2014 – Building a Recommender System

For the 2014 SBS track, INEX provides anonymous user profiles with the aim of determining the impact of these profiles on a personalized recommender system. We considered a combination of traditional retrieval system with this system to generate final rank ordered results. The anonymous user profile and the profiles of the user are described in [11, 12]. A user profile [12] is a list, which includes work ID, rating, entry date and tags for each book in his/her catalogue. A recommender system is designed to make use of information from users "similar to" the user who posted the query. Here we assume that similar users tend to have similar preferences and tastes in books.

There are 4 main steps in the implementation of the recommender system: (1) generating context vectors, (2) finding "similar users", (3) determining the contribution of the recommender system, and (4) producing the final scores. Details are found in the following sections.

3.4.1 Generating the Context Vector

The first step in our recommender system generates a matrix for each topic originator. These matrices consist of work IDs and tags, because we want to identify similar users who may have similar books (work IDs) and similar genres (tags) in their user catalogues. We consider tags and work IDs as feature vectors for this purpose because we are trying to find users similar to this particular topic originator. The matrices contain numeric values (ratings for work IDs and counts for tags). We used a rating of 0.1 to differentiate between a non-catalogued book (0.1) and catalogued book with zero rating (0.0). To reduce the dimensionality of the matrix, we decided that each person must have a minimum of 5 work IDs in common with the topic originator before a context vector is created for him/her. We label this matrix the *num-num* matrix. Figure 10 provides the algorithm for generating context vectors.

```
For each topic creator ‘t’
  Select tags and work IDs of ‘t’ as features
For each profile ‘p’
  If p has at least 5 work IDs  in common with ‘t’
    Build context vector for ‘p’ as
    For each work ID
    If work ID is a feature
        value = rating
    For each tag
    If Tag is a feature
        value = tag count
```

Figure 10: Algorithm for Context Vectors Generation

Figure 11 presents a sample topic originator profile, Figure 12 provides sample anonymous user profile and, Figure 13 shows an example of a context vector.

```
<catalog>
  <book>
      <LT_id>11162</LT_id>
      <entry_date>2006-04 </entry_date>
      <rating> 0.0</rating>
      <tags> history, science</tags>
  </book>
  …
  …
  …
   <book>
      <LT_id>56748</LT_id>
      <entry_date>2009-05</entry_date>
      <rating>2.0 </rating>
      <tags>ancient history, technology</tags>
  </book>
</catalog>
```

Figure 11: Example of a Topic Originator Profile

User ID	Work ID	Entry Date	Rating	Tags
u8218518	356331	2012-09	10.0	
u8218518	2081	2010-12	3.0	Adventure, potter
u9054475	5403381	2010-09	1.0	Philosopher
…..	……	……	……	……
u3174144	2856	2009-09	0.0	Supermarket, fastfood

Figure 12: Sample Profile

	356331	**2081**	**5403381**	**2856**	**philosopher**	**potter**	**supermarket**	**fastfood**
t1116	0.1	..	..	..	1	3	1	1
u8218518	9.0	..	..	..	1	1	..	..
u9054475	..	3.0	..	..	..	1	..	1
u3174144	..	..	1.0	..	..	..	4	..

Figure 13: Example of a Context Vector Matrix

3.4.2 Finding "Similar Users"

Once the context vectors are generated, the next step is to generate a list of *similar users* based on the context vectors. Pair-wise cosine similarity is used as the similarity between the user (query originator) and all other persons (each represented by his/her context vector). We identify the top-ranked 50 and 100 *similar users* as sets of interest.

3.4.3 Determining the Contribution of the Recommender System

We now generate Δ, the contribution of the recommender system, using as input, for each primary user: (1) the rank-ordered list of *similar users*, (2) the similarity score of each such user, (3) the rating for each work ID identified by document retrieval, and (4) the count of *similar users* having that same work ID in their catalogs. We use six different metrics to calculate the contribution of the recommender system R_{ij}. The *binary score* uses the count of the number of *similar users* whereas the *numeric score* uses the ratings associated with work IDs. In a cold start situation, wherein the user's profile does not exist, a contribution of 0.1 is made by the recommender system. Table 6 provides equations used to generate the contribution of the recommender system.

3.4.4 Producing the Final Score

A linear combination of the score produced by traditional retrieval (T_{ij}) and, the contribution of the recommender system (R_{ij}), produce a re-ranked list of "recommended" documents. The formula used to calculate the final score (F_{ij}) is

$$F_{ij} = (1-\lambda)*T_{ij} +\lambda*R_{ij}.$$

3.5 Related Work on the SBS Track

Singampalli [11] and Thotempudi [12] implemented different context vector representations. Table 5 summarizes the four context vector representations and their corresponding feature values. For details see [20].

Matrix Representation	Work ID Value	Tag Value
bin_bin	binary 1 = work ID exists 0 = otherwise	binary 1 = tag exists 0 = otherwise
bin_num	binary 1 = work ID exists 0 = otherwise	numeric tag frequency
num_bin	numeric rating for work ID	binary 1 = tag exists 0 = otherwise
num_num	numeric rating for work ID	numeric rating for work ID

Table 5: Context Vector Representations

Metric	**Binary Score**	**Numeric Score**
Metric 1	$R_{ij} = \sum_{k=1}^{50/100} s_{ik} + n$	$R_{ij} = \left(\sum_{k=1}^{50/100} S_{ik} + r_{jk}\right)$
Metric 2	$R_{ij} = n$	$R_{ij} = n$
Metric 3 (DCG-style)	$R_{ij} = \sum_{k=1}^{50/100} \frac{S_{ik} + 1}{\log_2(\text{rank}) + 1}$	$R_{ij} = \sum_{k=1}^{50/100} \frac{S_{ik} + r_{jk}}{\log_2(\text{rank}) + 1}$
Metric 4 (DCG-style variation)	$R_{ij} = \sum_{k=1}^{50/100} \left(\frac{S_{ik}}{\log_2(\text{rank}) + 1}\right) + n$	$R_{ij} = \sum_{k=1}^{50/100} \left(\frac{S_{ik}}{\log_2(\text{rank}) + 1} + r_{jk}\right)$
Metric 5 (MRR- style)	$R_{ij} = \sum_{k=1}^{50/100} \frac{S_{ik} + 1}{\text{rank}}$	$R_{ij} = \sum_{k=1}^{50/100} \frac{S_{ik} + r_{jk}}{\text{rank}}$
Metric 6 (MRR-style variation)	$R_{ij} = \sum_{k=1}^{50/100} \left(\frac{S_{ik}}{\text{rank}}\right) + n$	$R_{ij} = \sum_{k=1}^{50/100} \frac{S_{ik}}{\text{rank}} + r_{jk}$
i = topic id *j = work ID* *n= total no. of similar users having work ID 'j'* *k = similar user for topic 'i' (50/100)* *R_{ij} = Recommended score for topic 'i' work ID 'j'* *S_{ik} = Similarity score for user 'k'* *r_{jk} =Rating given by user 'k' for work ID 'j'*		

Table 6: Metrics for Calculating Δ (the Contribution of the Recommender System)

Figure 14 shows a high level view of the architecture of our recommender system (2014 UMD SBS system).

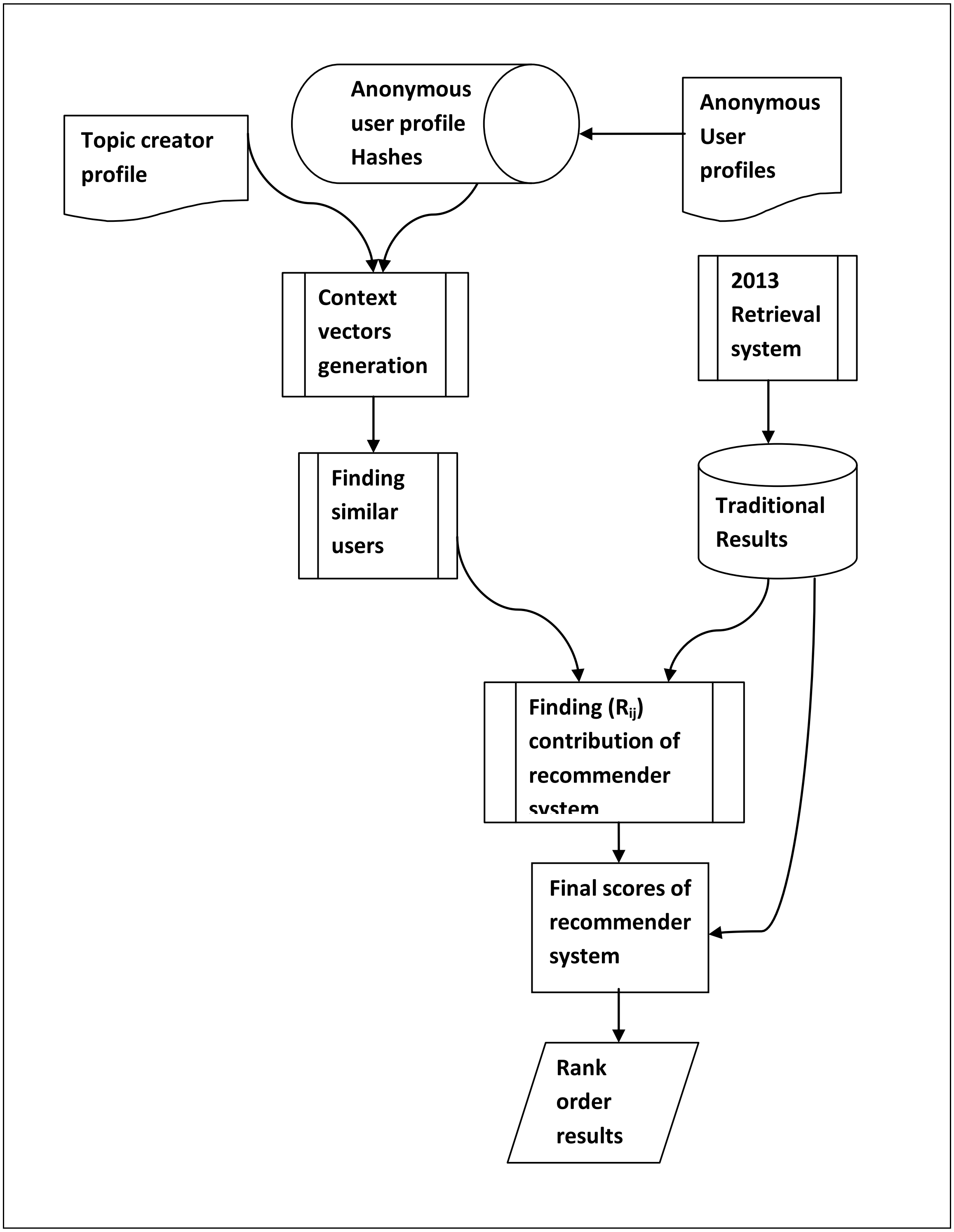

Anonymous user profile Hashes
Topic creator profile
Anonymous User profiles
Context vectors generation
2013 Retrieval system
Finding similar users
Traditional Results
Finding (R_{ij}) contribution of recommender system
Final scores of recommender system
Rank order results

Figure 14: Architecture of 2014 UMD SBS System

4 Experiments and Results

This section presents various experiments and their corresponding results.

4.1 INEX 2011 Results

For the 2011 SBS system, we used only title (T) as the query with six different indices. Pseudo-feedback experiments were also performed. We used 10 documents (**d**) and the top 50 terms (**t**) as feedback parameters. These values were selected based on [8]. QRels are required to evaluate the metrics here. INEX made 3 different QRel sets available (AMT, LT Official, LT Expanded). Results of traditional experiments are produced for each set. See Table 7, Table 8, and Table 9, respectively. Note that no feedback results are generated for professional and title indices as all early experiments produced dismal results. Note also that only 24 QRels were provided for AMT QRels. So the results are not necessarily meaningful.

For the LT Official QRels, ISBNs in the submitted runs are mapped to LT work IDs. The highest-ranked ISBN is mapped to the work ID; lower-ranked ISBNs (representing, perhaps, different editions of the same book) are removed from the results list. For LT Expanded QRels, multiple editions of the same book are retained.

Experiment 1:

(a) Feedback using Amazon Mechanical Turks (AMT) QRels

Run name	**nDCG@10**	**MRR**	**MAP**	**P@10**
Amazon_T	**0.5126**	0.7512	0.2855	0.4696
Amzon_T_fb.10.50	0.4522	0.7129	0.2827	0.4000
Full_T	0.4872	**0.6793**	**0.3019**	**0.4522**
Full_T_fb.10.50	0.4510	0.6486	0.2818	0.4174
LT_T	0.4106	0.6793	0.2062	0.3609
LT_T_fb.10.50	0.4072	0.6687	0.2088	0.3739
Professional_T	0.1815	0.4230	0.0934	0.1565
Social_T	0.4946	0.6778	0.2880	0.4609
Social_T_fb.10.50	0.4530	0.6838	0.2701	0.4087
Title_T	0.1655	0.3642	0.0946	0.1565

Table 7: 2011 Base Case & Feedback Runs, Traditional Retrieval, and 24 QRels from AMT

(b) Feedback using LT Official QRels with work ID values

Run name	**nDCG@10**	**MRR**	**MAP**	**P@10**
Amazon_T	0.2125	0.3415	0.1561	0.1392
Amzon_T_fb.10.50	0.2413	0.3824	0.1777	0.1598
Full_T	0.2780	0.4479	0.2051	0.1828
Full_T_fb.10.50	**0.2966**	**0.4664**	**0.2233**	0.1971
LT_T	0.2454	0.3747	0.1791	0.1765
LT_T_fb.10.50	0.2715	0.3931	0.1955	**0.2034**
Professional_T	0.0788	0.1438	0.0608	0.0534
Social_T	0.2826	0.4552	0.2043	0.1863
Social_T_fb.10.50	0.2948	0.4599	0.2190	0.1971
Title_T	0.0773	0.1404	0.0620	0.0525

Table 8: 2011 Base Case & Feedback Runs, Traditional Retrieval, and QRels from LT Official Set

(c) Feedback using LT Expanded QRels with ISBN values

Run name	nDCG@10	MRR	MAP	P@10
Amazon_T	0.2173	0.2963	0.1495	0.1917
Amzon_T_fb.10.50	0.2412	0.3296	0.1698	0.2103
Full_T	0.2781	0.3889	0.1988	0.2422
Full_T_fb.10.50	**0.3045**	**0.3974**	**0.2253**	**0.2721**
LT_T	0.2497	0.3033	0.1930	0.2353
LT_T_fb.10.50	0.2646	0.3121	0.2048	0.2554
Professional_T	0.0714	0.1334	0.0587	0.0588
Social_T	0.2814	0.3968	0.1984	0.2441
Social_T_fb.10.50	0.2985	0.3897	0.2225	0.2672
Title_T	0.0729	0.1355	0.0660	0.0583

Table 9: 2011 Base Case & Feedback Runs, Traditional Retrieval, and QRels from LT Expanded Set

From Tables 7-9, we can say that the Full index with feedback provides the best results for the given set of queries.

4.2 INEX 2013 Results

From [11], we conclude that the best results are obtained by using the Full index, Title-Query-Group (TQG) combination for the query, and pseudo-feedback with **d**=10 and **t**=50. As the aim of a recommender system is to move relevant books up in rank, we utilized recall as the basis for selecting index and topic combinations.

We apply the recommender system only to topics which actually have work IDs in their corresponding creator catalogues; “similar users” for the topic user with no work IDs in his catalogue do not exist by our definition.

For our 2013 recommender system, we use the following metric $R_{ij} = \sum_{k=1}^{50/100} \frac{s_{ik}*r_{jk}}{s_{ik}*10}$, where if no similar user exists with work ID j then the average rating of j is taken as R_{ij}. Table 10 shows results obtained by applying the recommender system using specified values of λ. Our initial λ value is taken from [14]. Upon observing results from [11], we decided to use only binary values in our context matrices. We then further tuned our system using only similar users from the bin_num matrix representation.

#Similar Users	**λ**	**nDCG@10**	**MRR**	**MAP**	**P@10**
50	0.0001750	**0.0931**	**0.1754**	**0.0610**	0.0550
	0.0001800	0.0929	0.1748	0.0606	**0.0553**
	0.0001855	0.0926	0.1738	0.0602	**0.0553**
	0.0001900	0.0924	0.1736	0.0601	0.0550
	0.0001950	0.0923	0.1740	0.0599	0.0550
100	0.0001750	**0.0937**	**0.1784**	**0.0616**	**0.0558**
	0.0001800	0.0934	0.1780	0.0612	**0.0558**
	0.0001855	0.0930	0.1778	0.0608	0.0555
	0.0001900	0.0930	0.1781	0.0607	0.0555
	0.0001950	0.0928	0.1781	0.0604	0.0553

Table 10: Recommended Results (Full Index, TQG Query Set, Pseudo-Feedback [d=10, t=50])

4.3 INEX 2014 Official Results

Before INEX evaluation, we tuned our system by using the 2013 queries and QRels. We submitted 2014 results using the best λ value from 2013. Table 11 shows results from the official submission.

Run name	nDCG@10	MRR	MAP	R@1000
UMD - Full_TQG_fb.10.50_0.0000227_50.trec	**0.097**	**0.188**	**0.069**	0.328
UMD - Social_TQG_fb.10.50_0.0000222_50.trec	0.096	0.184	0.067	0.327
UMD - Full_TQG_fb.10.50_0.0000255_100.trec	0.096	**0.188**	0.068	0.328
UMD - Full_TQG_fb.10.50_traditional.trec	0.095	0.185	0.068	0.328
UMD - Full_TQ_fb.10.50_0.0000247_100.trec	0.092	0.176	0.064	0.321
UMD - Full_T_fb.10.50_0.0000260_100.trec	0.070	0.139	0.047	0.253

Table 11: 2014 Official INEX Submissions

Upon access to the 2014 QRels, we re-examined our feedback values of **d** and **t** with the aim of improving recall. R@1000 improved from 0.328 (at **d**=10 and **t**=50) to 0.380 at (**d**=10 and **t**=15), as seen in Table 12 . We used this retrieval run as the basis of our next set of results.

Run	# docs	#terms	nDCG@10	MRR	MAP	R@1000
Official INEX run	10	50	0.095	0.185	0.068	0.328
Current results	10	15	0.091	0.182	0.064	**0.380**

Table 12: Traditional Retrieval (Full Index, TQG Query Set with Pseudo-Feedback)

Table 13 shows final results of the recommender system using Metric 1 and Metric 2 (Input from “current results” of Table 12).

We further tuned our traditional retrieval system by using weighted feedback [12] and observed that best recall values are obtained for feedback weights of 0.7 and 0.8. [11, 12] show that metric 3 and metric 5 produce superior results. Table 14 shows results produced by the num_num representation with λ set to 0.0000100 in the recommender system.

Metric	Users	λ	nDCG@10	MRR	MAP	R@1000
Metric 1	50	0.0000100	0.0890	0.1848	0.0597	0.3801
	100	0.0000100	0.0820	0.1753	0.0551	0.3801
Metric 2	50	0.0000100	0.0944	0.1930	0.0639	0.3801
	100	0.0000100	0.0918	0.1904	0.0609	0.3801

Table 13: Final Results of Recommender System (Full Index, TQG, num_num Matrix, Feedback [d=10, t=15])

Table 15 shows the best results from the all experiments (see [11] for details of bin_num matrix representation and [12] for bin_bin and num_bin matrix representations).

Score type	Feedback weight	#Metric	#user	nDCG@10	MRR	MAP	R@1000
Binary	0.7	3	50	0.0913	0.1830	0.0641	0.3819
			100	0.0916	0.1881	0.0621	0.3819
		5	50	0.0936	0.1869	0.0667	0.3819
			100	**0.0969**	0.1912	0.0678	0.3819
	0.8	3	50	**0.0913**	0.1830	0.0641	0.3811
			100	0.0873	0.1846	0.0583	0.3811
		5	50	0.0862	0.1825	0.0604	0.3811
			100	0.0905	0.1875	0.0619	0.3811
Numeric	0.7	3	50	**0.0913**	0.1830	0.0641	0.3819
			100	0.0873	0.1846	0.0583	0.3819
		5	50	0.0862	0.1825	0.0604	0.3819
			100	0.0711	0.1875	0.0619	0.3819
	0.8	3	50	**0.0913**	0.1830	0.0641	0.3811
			100	0.0873	0.1846	0.0583	0.3811
		5	50	0.0862	0.1825	0.0604	0.3811
			100	0.0905	0.1875	0.0619	0.3811

Table 14: Final Results of Recommender System (Full, TQG, num_num Matrix, Feedback [d=10, t=15])

Metric	Feature	Users	λ	nDCG@10	MRR	MAP	R@1000
Metric 3	bin_num	50	0.0000075	0.0965	0.1931	0.0662	0.3801
		100	0.0000075	0.0958	0.1932	0.0661	0.3801
	bin_bin	**50**	**0.0000075**	**0.1025**	**0.2041**	**0.0715**	0.3801
		100	0.0000075	0.1004	0.1997	0.0697	0.3801
Metric 5	bin_num	50	0.0000125	0.0977	0.1946	0.0670	0.3801
		100	0.0000125	0.0978	0.1961	0.0685	0.3801
	bin_bin	**50**	**0.0000125**	**0.1058**	0.2077	**0.0746**	0.3801
		100	0.0000125	0.1053	**0.2084**	0.0722	0.3801

Table 15: Final Results of the Recommender System

5 Conclusions and Future Work

Table 15 and results from [11, 12] shows that the best features to represent context vectors are binary_binary (bin_bin), where both work IDs and tags are represented as binary values. The *similar users* are better at 50 (rather than 100). Metric 5, 50 similar users from bin_bin matrix representation and λ set to 0.0000125 produce a higher nDCG@10 result.

From R@1000 we also observe that few relevant documents are retrieved in the top 1000 during document retrieval. One reason may be because QRels are retrieved from answers in the LT forum and these users might not have knowledge of all books, whereas document retrieval retrieves all correlating books. Increasing recall at this stage may be expected to produce improvement in the final scores. Our current best result (0.1058) would rank at 17 in terms of nDCG@10 and 13 in terms of R@1000 when compared to the INEX 14 official results. As of now, no significance tests are performed by INEX on SBS track Results. Once the significance tests are performed by INEX, we can say how significant our results are compared to other participants.

Clearly there are many possibilities for improving results at different phases in both the traditional and recommender systems. Recall may improve if we use data from catalogues, such as titles of work IDs and tags as feedback in traditional retrieval. Considering the structure of documents and providing weights to query terms may also be helpful.

There are many possible improvements in our recommender system. Following are such areas. Tuning the system-generated ratings using Root Mean Square Error (RMSE) [31] and cross-validations, using book titles of work IDs and Amazon similar books as features in context vectors, using only work IDs instead of both work IDs and tags as features, using Pearson correlation coefficient [27] instead of pairwise cosine similarity in generating similar users, and applying model-based recommender systems [28] are some such areas. As there are multiple steps involved in generating final results, there are multiple options to improve results. This is our first attempt at this task, which has proved to be an excellent learning experience.

References

[1] About INEX [Internet]. Amsterdam, Netherlands. INEX: c 2008-2014 [cited 2014 June 20]. https://inex.mmci.unisaarland.de/about.html

[2] Salton, G. *The Smart Retrieval System-Experiments in Automatic Document Processing*. Prentice-Hall, 1971.

[3] Singhal, A., Buckley, C. and Mitra, M. Pivot Document Length Normalization. *ACM SIGIR*, 1996.

[4] Ganapathibhotla, M. Query Processing in a Flexible Retrieval Environment. *MS Thesis*, University of Minnesota Duluth, 2006.

[5] INEX 2014 Social book search track. Available from: https://inex.mmci.uni-saarland.de/tracks/books/

[6] Bogers, T., Wilfred, C.K. and Larsen, B. RSLIS at INEX 2011: Social Book Search Track. *INEX 2011*, LNCS 7424, pp. 45–56, 2012.

[7] Bogers, T. and Larsen, B. RSLIS at INEX 2012: Social Book Search Track. *INEX 2012 Workshop Pre-proceedings*, INEX Working Notes Series, pp.97-108, 2012.

[8] Adriaan, F., Kamps, J. and Koolen, M. University of Amsterdam at INEX 2011: Book and Data Centric Tracks. *INEX 2011 Workshop Pre-proceedings*, INEX Working Notes Series, pp.36-48, 2011.

[9] Koolen, M., Kazai, G., Preminger, M. and Doucet, A. Overview of the INEX 2013 Social Book Search Track. *CLEF 2013 Working Notes Series*, Valencia, Spain, 2013.

[10] Strohman, T., Metzler, D., Turtle, H. and Croft, W. B. Indri: A Language Model-based Search Engine for Complex Queries. *Proceedings of the International Conference on Intelligent Analysis*, 2(6), pp. 2-6, 2005.

[11] Singampalli, L.L. Social Book Search: A Methodology that Combines Retrieval and Recommendation. *MS Thesis*, University of Minnesota Duluth, 2014.

[12] Thotempudi, V.K. A Recommender System for Social-Book Search. *MS Thesis*, University of Minnesota Duluth, 2014.

[13] Jarvelin, K. and Kekalainen, J. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems*, 20(4), pp.422–446, October , 2002.

[14] Huurdeman, H., Kamps, J., Koolen, M. and Wees, J. V. Using Collaborative Filtering in Social Book Search. *INEX 2012 Workshop Pre-proceedings*, INEX Working Notes Series, pp.125-136, 2012.

[15] Lavrenko, J. and Croft, B. A language modeling approach to information retrieval. *Proceedings of the 1998 ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.275–281, 1998.

[16] About Librarything [Internet].
Available from: https://www.librarything.com/about

[17] Amazon website: http://www.amazon.com/

[18] Smart stop list [Internet]:
http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop

[19] Dewey Decimal classes [internet]. Available from:
http://bpeck.com/references/DDC/ddc.htm

[20] Cappellato, L., Ferro, N., Halvey, M. and Kraaij, W. editors (2014). CLEF 2014 Labs and Workshops, Notebook Papers. *CEUR Workshop Proceedings (CEUR-WS.org)*,ISSN 1613-0073,http://ceur-ws.org/ .

[21] Salton, G., Wong, A. and Yang, C.S. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11), pp.613-620, 1975.

[22] The Lemur Project: http://www.lemurproject.org/

[23] Krovetz, R. Viewing Morphology as an Inference Process. *Proceedings of the 16th Annual International ACM SIGIR Conference*

on Research and Development in Information *Retrieval*, pp.191-202, 1993.

[24] Zhai, C. and Lafferty, J. A study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in information Retrieval (SIGIR '01)*, pp.334-342, 2001.

[25] Zhai, C. and Lafferty, J. Two-stage Language Models for Information Retrieval. *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in information Retrieval (SIGIR '02)*, pp.49-56, 2002.

[26] Lavrenko, V. and Croft, B. Relevance-Based Language Models. *SIGIR'01*, pp.120 -127, September, 2001.

[27] McLaughlin, M. R. and Herlocker, J. L. A Collaborative Filtering Algorithm and Evaluation Metric that Accurately Model the User Experience. *Proceedings of 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '04)*, pp. 329–336, Sheffield, UK, 2004.

[28] Hofmann, T. Latent Semantic Models for Collaborative Filtering. *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 89–115, 2004.

[29] Extensible Markup Language [Internet]. World Wide Web Consortium: c 1996- 2003. Available from: http://www.w3.org/XML/.

[30] Amazon Mechanical Turks: https://www.mturk.com/mturk/welcome.

[31] Melville, P., Mooney, R. J. and Nagarajan, R. Content Boosted Collaborative Filtering for Improved Recommendations. *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI '02)*, pp. 187–192, Edmonton, Canada, 2002.