Social Book Search: A Methodology that Combines both Retrieval and Recommendation

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Lakshmi Lavanya Singampalli

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Dr. Carolyn J. Crouch

August 2014

Acknowledgements

I would like to thank Dr. Carolyn Crouch and Dr. Donald Crouch for their constant guidance and advice. Their support in every aspect of my research was crucial to my completion of this degree. I would like to thank Dr. Marshall Hampton for serving on my committee and taking the time to talk to me on several occasions. A special thanks to Dr. Ted Pedersen for offering me one of the best courses on Natural Language Processing that helped me a great deal in delivering this thesis.

Thanks to my fellow students Vamshi Krishna Thotempudi and Ravva RaviKiran for their constructive feedback and help with the experiments.

To Jim Luttenin, Lori Lucia, Clare Ford for their cordial support and valuable help. Lastly, I thank my family and friends without whose support I would not have come this far. I am and will be, ever so indebted to them.

Dedication

I would like to dedicate this thesis to my father who gave me the courage and inspiration I needed at every walk of my life. He showed me that anything is possible with faith, hard work and determination. It is impossible to thank him adequately for everything he has done for me. I could not have asked for a better parent or role-model.

Abstract

Information Retrieval as an area of research aims at satisfying the information need of a user. Retrieval in the Information Age has expanded exponentially as its underlying technologies have expanded. Traditional IR systems that give response to a user's natural language search query are combined with recommendation through collaborative filtering [6]. This research focuses on a methodology that combines both traditional IR and recommender systems. It is done as part of the Social Book Search (SBS) Track, Suggestion task of INEX (*INitiative for the Evaluation of XML Retrieval*) 2014 [3].

The Social Book Search Track was introduced by INEX in 2011 with the purpose of providing support to users in terms of easy search and access to books by using metadata. One complexity of the task lies in handling both professional and social metadata which are different in terms of both kind and quantity. Methodology and experiments discussed are inspired by background research [1,2,4,5,6] on the Social Book Search track. Our IR team submitted six runs for the track to the INEX 2014 competition, five of which use a recommender system that re-ranks the otherwise traditional set of results. Background work done to establish a good foundation for the methodology used in the SBS 2014 task includes experiments performed on both the 2011 and 2013 Social Book Search tracks. This research focuses on the 2013 experiments and their impact on results produced for SBS 2014.

Contents

List of Tables

List of Figures

1 Introduction

Information Retrieval (IR) is the process of extracting information, (either structured or unstructured), from large collections of data to satisfy the need of a user. The information extracted in response to a query comes from a variety of domains and takes many forms. The domain of a retrieval system may be open or closed. With closed domains, information is extracted from a relatively small set of constrained data, unlike the open domain which is highly general. The response can be factoid (facts), making retrieval easier, or it can be non-factoid (short summaries or snippets), which is harder to produce than the former. Information retrieval systems can also be classified based on the amount of information they access [7]. The Web provides access to billions of documents and is the most prominent and largest scale of retrieval. Personal information retrieval has access to the smallest number of documents. On a scale between these two is enterprise information retrieval, which is domain-specific and has access to all documents within an organization.

Information Retrieval as an area of research is challenging as there are limitations with respect to the ability of computers or any technology to understand human language. In the traditional process of retrieval, given natural language queries and a corpus, the IR system retrieves and returns a ranked list of documents that satisfy the need of the user posting the query. Various models and tools may be incorporated to facilitate retrieval.

This thesis discusses the design and implementation of experiments proposed by INEX (*INitiative for the Evaluation of XML Retrieval*) [8], which also aims at advancing the state of knowledge with respect to XML [17] retrieval. Every year INEX provides different tasks in the field. According to the official 2014 INEX website, the active tracks are the Social Book Search, Linked Data and Tweet Contextualization. The objective of this research is to provide a robust and efficient system for the Suggestion task, which is a part of the Social Book Search Track (SBS) [9] in INEX14.

In the Social Book Search Track, participant teams are provided with an Amazon Corpus [11] (consisting of around 2.8 million book descriptions) and extra metadata from LibraryThing [10]. The track aims at answering queries posted on LibraryThing by various users, who are in search of books they might be interested in. The answers are presented as a ranked list of documents that best fit the query (i.e., are "most relevant" to it). The book descriptions from the Amazon Corpus are in structured XML format and have both Social and Professional content included in them. User-generated content from LibraryThing is also included in these XML documents.

Any IR system depends heavily on indexing. Indexing of documents identifies their content and determines the documents produced as a result. There are various state-of-the-art IR tools that do exceptionally well at indexing [12,13,14]. The tool used as part of this research is the Indri search engine [14]. Indri is a tool that is part of the Lemur project [18] and uses probabilistic Language Modeling (LM) along with inference networks. In this tool, each document has a language model created for it [15]. This language model helps determine how many query terms can be extracted from the document. The inference network assumes that the more the query terms it contains, the more relevant a document is to the query [16].

This year (2014) the Social Book Search track focuses more on the recommender system rather than the traditional system. The recommender system uses content generated from user profiles and produces a re-ranked list of recommended documents based on users similar to (i.e., with similar interests to) the user who posted the query. The primary objective of this thesis is to generate a method that combines both retrieval and recommendation to generate a possibly better ranked list of documents than the list produced by retrieval alone.

The evaluation measures that help determine the outcome of the retrieval system are calculated using prescribed metrics from Text REtrieval Conference (TREC) [19]. Normalized Discounted Cummulative Gain (NDCG) [20] is the main evaluation metric, along with others such as precision and recall.

The thesis is organized as follows. Background information on the SBS tracks (2011-2014), the tools and techniques used for retrieval on this track, and evaluation measures are described in Chapter 2. The approach and methodology are discussed in Chapter 3. An in-depth view of the experiments conducted along with the results obtained through INEX evaluation are given in Chapter 4. Chapter 5 presents conclusions and recommendations for future work.

2 Background

This chapter gives an overview of INEX [8], its associated tracks and the prescribed TREC [19] evaluation measures. It also discusses the papers that we referenced in beginning our work on the SBS Track [1,2,4,5,6].

2.1 INEX

The INitiative for the Evaluation of XML Retrieval (INEX) [8], a global forum, was established in 2002 with the aim of improving the retrieval of relevant elements (i.e., pieces of information). Every year it provides a platform for various organizations to compete on XML-related tasks and compare their results. The data collections and the evaluation measures used in this competition are distributed in common to every organization participating in the tracks.

In 2012, the active tracks were Snippet Retrieval, Tweet Contextualization, Linked Data, Relevance Feedback and Social Book Search. INEX 2013 continued all of the tracks except for Relevance Feedback. Over 100 universities have participated in tracks presented by INEX over the last three years. We chose the 2011, 2012 and 2013 Social Book Search Tracks as the source of our experiments for this year.

2.2 Related Work

It is better to understand the architecture of the system and get a general idea of how the system works in a traditional environment before delving deeper into the details and methodology. This section presents related work in Information Retrieval as a research area before going forward with the Social Book Search track. Many methodologies and tools were studied to understand how they work.

Retrieval in the Information Age has expanded exponentially as its underlying technologies have expanded. In traditional retrieval, the primary goal is to retrieve relevant documents. In XML retrieval, relevant information from relevant documents may be retrieved as a non-factoid or factoid response.

Focused Retrieval was pursued by our IR team for quite some years before delving into the Snippet Retrieval track. Basic IR models are incorporated to facilitate indexing and retrieval. The model used in previous research in the Snippet track was the Vector Space Model [21].

2.2.1 Snippet Retrieval Track

The Snippet retrieval track aims at generating snippets in response to queries posted by a user. These snippets come from highly correlated documents that are, in our work, dynamically retrieved using Flex [22] and Smart [13], which is based on the Vector Space Model.

The Vector Space Model (VSM) is arguably the most popular traditional information retrieval model. As the name suggests, the model creates a space wherein both the documents and queries are represented as vectors. It has three main stages in its implementation: document indexing, term weighting and retrieval. Document indexing involves filtering words with significance to the document and producing a term vector from them. (Words that do not add meaning to a document such as function words are removed.) The term weighting stage involves adding value to the terms by recognizing their frequency both in the document and across the collection as a whole. The rank ordering takes place when the terms in the document are correlated with the terms in the query, based on a predefined similarity function.

The Smart retrieval system, which automatically indexes and retrieves text, is based on the VSM. Thus we have the documents and queries represented as vectors in a vector space. This data is further structured into a tree of XML elements for use by Flex [22] which performs dynamic retrieval of elements that form the basis of snippets.

2.2.2 Reference Run Generation

Generating a good reference run for the snippet track is essential to its success. The reference run is a result set of retrieved documents for the query set. It is used by the teams during development of their methodology to evaluate their current results. The teams may utilize the reference run generated by INEX or they may choose to generate their own.

2.3 Social Book Search Task

The Social Book Search task was introduced by INEX in 2011; it has since had many active participants. Its primary goal is to facilitate easy access to and search for books that a user might be interested in, based on the query he posted. Book search and retrieval can be done traditionally using indexing methodologies and IR models and tools. This is a successful and classic approach since the early years. However, many suggestion tasks at present include using a recommender system which is a more state-of-the-art enhancement to the traditional ways. Background research was essential on the methodologies and recommender systems before coming up with our own system for recommendation.

Our background research centered on the 2011, 2012 and 2013 Social Book Search tracks. The 2011 track involved generating a ranked list of books and determining the effect of social and professional data on the result. The social data included tags, reviews and other user-generated content added to the XML from LibraryThing and Amazon. The 2012 track was an extension of the 2011 track with more focus on a recommender system. User preference information that came from the user profiles listed on LibraryThing were used as part of the retrieval process to see if there were improvements over the traditional system. The 2013 task was again like the 2011 task with more focus on identifying which XML tags were more important than others.

The results generated in 2011 and 2013 were consistent and as expected. In 2012, the top-ranked participant teams did not produce consistent outcomes. Some teams found the use of a recommender system improved results whereas some teams felt otherwise. In 2014, the track was introduced with emphasis on how the recommender system improves traditional results. The data in 2014 is much cleaner than the previous years and the number of queries (680) doubled from 2013. The evaluation metrics used to evaluate the runs are stipulated by TREC.

2.3.1 Document and Query Collections

The document collection for the Social Book Search Track is huge with 2.8 million book descriptions that contain XML tags from both Amazon and LibraryThing. Each book has its own XML document. The XML document includes, from Amazon, professional data (eg., publisher, title, creator, subject) and social data (eg., editorial reviews and tags) along with user-generated content in the form of user reviews and ratings. The data from LibraryThing is also user-generated and is user-preferred data. The tags and structure of the documents are described in Chapter 3.

Topics or queries posted by a user are also given and are produced from LibraryThing. A sample topic is presented in Figure 1. Each topic associated with a user has his/her user profile included as part of the topic. The user profile contains information about his/her interests. These interests are usually found by analyzing the catalogue of a user (which is part of his/her profile). The genre (e.g., fiction) included in his catalogue is based on that of the majority of books in the catalog. The topics are all in structured format and hence follow a specific definition (see Figure 2).

```
<topics>
 <topic id="1116">
        <title>Which LISP?</title>
        <mediated_query>introduction book to Lisp</mediated_query>
        <group>Purely Programmers</group>
        <narrative>        It'll be time for me to shake things up and learn a new language soon. I had started on Erlang a while back and getting back to it might be fun. But I'm starting to lean toward Lisp--probably Common Lisp rather than Scheme.   Anyone care to recommend a good first Lisp book?
</narrative>
        <catalog>
            <book>
                <LT_id>859035</LT_id>
                <entry_date>2006-04</entry_date>
                <rating>0.0</rating>
                <tags>livingry, hunger, poverty, policy</tags>
            </book>
            <book>
                <LT_id>569191</LT_id>
                <entry_date>2005-11</entry_date>
                <rating>0.0</rating>
                <tags>law, the firm, economics</tags>
            </book>
            <book>
                . . .
            </book>
          . . .
        </catalog>
 </topic>
</topics>
```

Figure 1 Sample Topic XML from the Corpus of SBS 2014 (1116.xml)

```
<!ELEMENT topics (topic)>
<!ELEMENT topic (title, mediated_query, group, narrative, catalog)>
<!ATTLIST topic id ID #REQUIRED>
<!ELEMENT title (#PCDATA)>
<!ELEMENT mediated_query (#PCDATA)>
<!ELEMENT group (#PCDATA)>
<!ELEMENT narrative (#PCDATA)>
<!ELEMENT catalog (book*)>
<!ELEMENT book (LT_id, entry_date, rating, tags)>
<!ELEMENT LT_id (#CDATA)>
<!ELEMENT entry_date (#CDATA)>
<!ELEMENT rating (#PCDATA)>
<!ELEMENT tags (#PCDATA)>
```

Figure 2 DTD for a Topic (SBS 2014)

The mediated query is an expanded version of the title. The narrative states what the user is looking for in a descriptive manner. The group refers to the group in LibraryThing in which the query was posted.

Apart from the topics and the corpus, a set of 94,000 anonymized user profiles from LibraryThing are also provided. This is used to help generate recommendations for topics (using collaborative filtering) in conjunction with a recommender system developed by the teams. Each profile contains information by the anonymous user about catalogued books.

2.3.2 INEX Run Submission Format

The runs submitted to INEX have a specific format. INEX allows six runs from each participating team where one run is based solely on indexing the title tag. The submission format is as follows: topic_id, Q0, ISBN number, rank, RSV score, run_id. The fields are defined in Table 1.

Field	Description
topic_id	The topic number, which is based on the LT forum thread number
Q0	Query number, it is used for evaluation purpose by organizers and is always fixed as Q0
ISBN	ISBN number of a book; unique for each book
Rank	The rank at which the document is retrieved for a particular topic
RSV	Retrieval Status Value; A score that ranks results in a decreasing order
Run_id	A code that identifies the participating group and run

Table 1: Definition of Fields in the SBS Track Submission Format

2.3.3 Evaluation

The retrieved ranked list of results for the track are evaluated based on metrics by TREC (TExt Retrieval Conference) [19]. The official evaluation metric is nDCG@10 (Normalized Discounted Cummulative Gain) [20] where 10 refers to the top ten documents of the list on which the metric is calculated. The other required metrics are: Mean Reciprocal Rank (MRR), Mean Average Precision (MAP) and Recall@1000.

The measures used for evaluation tend to give evaluation scores based on the quality of the top-ranked documents. For NDCG, a gain is accummulated from top to bottom in the ranked list. Documents at lower ranks tend to have discounted gains (gain is reduced) (hence the name, Discounted Cummulative Gain [DCG]) . This discounted gain is calculated by giving a penalty of logarithmic value that is proportional to the position of the document. The equation can be seen in Table 2. This supports the assumption that the higher the position of the document in the list, the more relevant it is.

The DCG value at each position should be normalized until a defined position ‘p’ is realized across queries to get the NDCG value. The NDCG value can be calculated as DCG value over the maximum DCG value that can be obtained till position ‘p’. The maximum NDCG value is called ‘Ideal DCG’ (IDCG). All these equations are given in Table 2.

The Mean Reciprocal Rank (MRR) is an evaluation measure used to find the level of correctness of a ranked list of documents. The reciprocal rank of a query is the 'multiplicative inverse' of the position of the first correct document in the retrieved list. The mean of all the reciprocal ranks across queries is MRR (Table 2).

In information retrieval, Precision and Recall are the two basic metrics. Precision refers to the fraction of documents retrieved that are relevant and recall refers to the fraction of relevant documents retrieved. The Mean Average Precision (MAP) is the mean of the average precisions of a set of documents that are retrieved as relevant and Recall @1000 is the recall at 1000 documents retrieved for each query. (see Table 2)

Metric	Equation	Description
NDCG@10	$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$ NDCG = DCGp/IDCGp	p = particular position IDCG =Ideal DCG NDCG = Normalized DCG NDCG@10 => p = 10
MRR	$(1/\lvert Q\rvert) * \sum_{n=1}^{\lvert Q\rvert} 1/rank_n$	Q = sample of queries Q n = number of query
MAP	$(\sum_{q=1}^{Q} AveP(q)) / Q$	AveP = Average Precision Q = Sample of queries Q q = number of query
Recall@1000	$\frac{\lvert\{rel@1000\} \cap \{ret@1000\}\rvert}{\lvert\{rel@1000\}\rvert}$	rel = relevant documents ret = retrieved documents

Table 2. Evaluation Metrics Prescribed by TREC for Evaluation of SBS 2014

These are the TREC evaluation measures stipulated to be used to compare the results of participating teams. In Chapter 3, our methodology and the design of our recommender system are presented.

3 Methodology

Our approach to the Social Book Search (SBS) Track that forms the basis of this thesis uses both Information Retrieval (IR) and Natural Language Processing (NLP) tools. It uses a method that combines traditional retrieval with recommendation. The literature review, tools used, and flow of events (architecture) of both the traditional and recommender systems are discussed in this chapter.

3.1 Traditional System

This section presents background from the SBS 2011 track [23], work performed by the author on the improved SBS 2013 track and work on the latest SBS 2014 track by [24]. Each year the data that evolves is cleaner and the topic sets larger.

3.1.1 Background - SBS 2011 track

The 2011 SBS track included four tasks: Social Search for Best Books (SB), Prove It! (PI), Structure Extraction (SE), and Active Reading (AR). Our focus is the SB task. The SB task goal is evaluating the efficiency of professional data, such as controlled keywords and publishing details, versus social data such as tags, ratings and reviews for retrieving a ranked list of relevant documents. Professional data is used for highly accurate retrieval and the classification of books in a large catalogue but requires a fair amount of expertise and training to use, whereas social data lacks vocabulary control but is helpful in reflecting general search terminology. Each has its own advantages and disadvantages.

The data set provided for the 2011 task included 2.8 million book descriptions from both Amazon and Librarything. Each book had its own XML document with both social and professional tags included. The topic set had 211 topics generated by users of LibraryThing. A set of training topics were also provided for the participant teams to work on. The operations conducted on the data (scrubbing, parsing, indexing) and the results can be found in detail in [23]. It was observed by both the participants and organizers that, on increasing the quality of data (both topic set and data set), a better set of results could be obtained.

In the 2011 data set, there is a DDC code for 61% of the descriptions and 57% of the collection has at least one subject heading [25]. More than 1.2 million descriptions (43%) have at least one review and 82% of the collection has at least one LibraryThing tag [25]. The experiments on the 2011 track helped us achieve a good understanding of which tags to use for efficient traditional retrieval. User profiles were not yet a part of the data in 2011. (Experiments on recommender systems started as early as 2012.)

Apart from helping us know how to evaluate the traditional system of the track, the 2011 experiments also proved to be a base for some other interesting results. The touchstones (links to documents posted in response to a query by other members of LibraryThing) which were part of the topic set proved to be a good source of recommendations for the final set of relevant documents. Needing improvement were the quality of the dataset and the topic set. On inclusion of user profiles in 2012, the focus became the recommendation part of the system that took as input the traditional results.

3.1.2 Architecture of the Traditional System (2013 SBS track)

Our traditional system has a methodology that is enriched by a set of IR and NLP tools. Figure 3 shows the architecture of the traditional system over the 2013 SBS track. Details follow.

3.1.2.1 Amazon and LibraryThing Corpus

The dataset consists of 2.8 million XML documents from Amazon and LibraryThing. The tags include both professional and social metadata. A sample XML document is shown in Figure 4.

3.1.2.2 Scrubbing

Before the XML documents can be used for indexing, they must be scrubbed and parsed. Not all the data provided as part of the document is useful. Some is redundant and some is unimportant or of no use. Scrubbing removes unwanted tags that are part of the XML document. A sample XML document is shown in Figure 4. Tags removed as part of the scrubbing process are shown in Table 3.

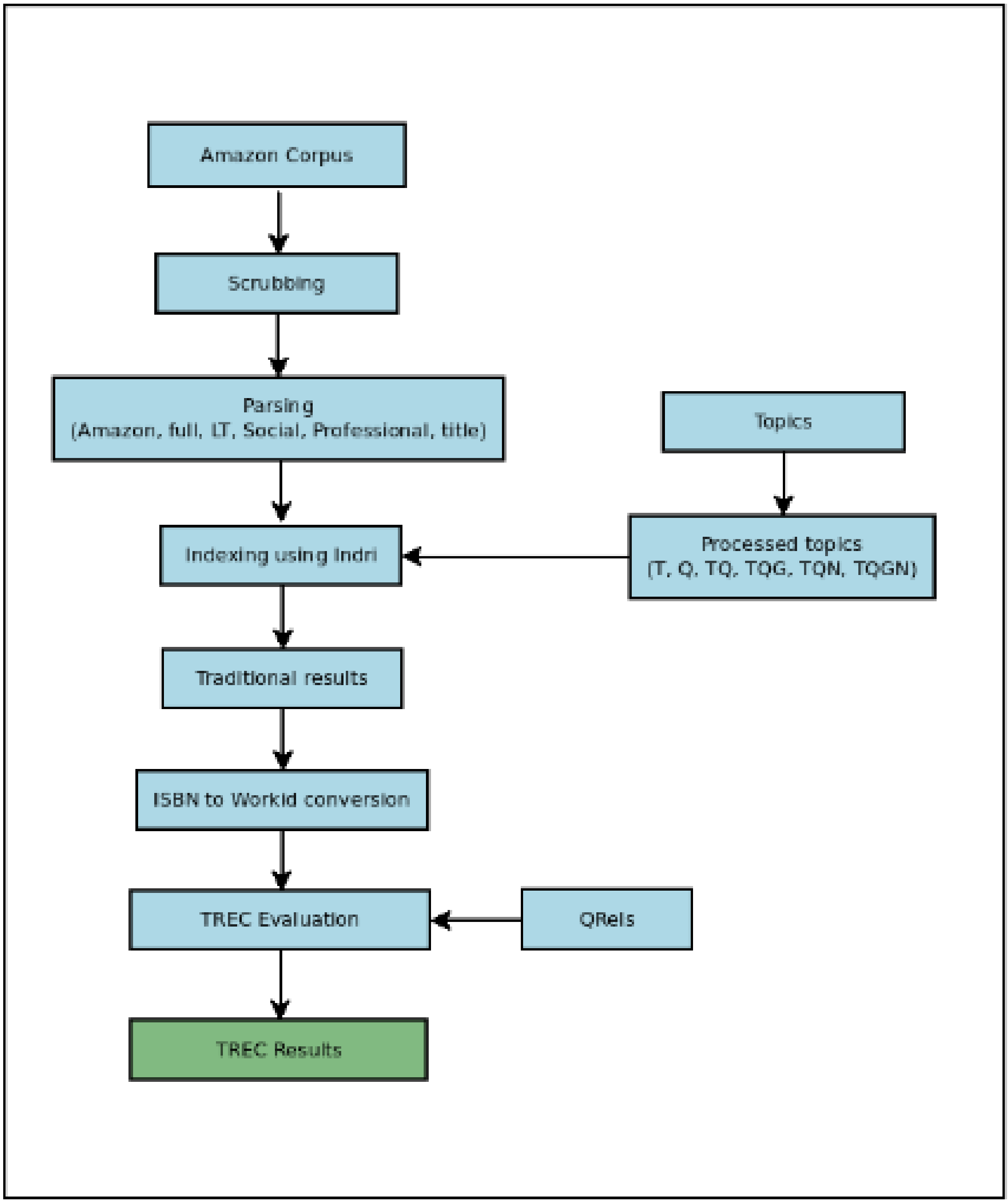

Figure 3 Traditional System Architecture (SBS Track 2013)

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- version 1.0 / 2009-11-04T13:44:03+01:00 --><!DOCTYPE book SYSTEM "books.dtd">
<book>
  <isbn>0001360000</isbn>
  <title>Mog's Kittens</title>
  <ean>9780001360006</ean>
  <binding>Board book</binding>
  <label>HarperCollins UK</label>
  <listprice>$6.99</listprice>
  <manufacturer>HarperCollins UK</manufacturer>
  <publisher>HarperCollins UK</publisher>
  <readinglevel>Ages 4-8</readinglevel>
  <releasedate/>
  <publicationdate>1994-09-01</publicationdate>
  <studio>HarperCollins UK</studio>
  <edition/>
  <dewey/>
  <numberofpages>16</numberofpages>
  <dimensions>
    <height>55</height>
    <width>461</width>
    <length>469</length>
    <weight>22</weight>
  </dimensions>
  <reviews>
    <review>
    <date>2007-11-27</date>
    <summary>Cute Book</summary>
    <content>An very cute book for the young child. Great for the cat or animal lover.
      &lt;br /&gt;Author of "Hobo Finds A Home"
      &lt;br /&gt;
    </content>
    <rating>5</rating>
    <totalvotes>0</totalvotes>
    <helpfulvotes>0</helpfulvotes>
    </review>
  </reviews>
  <editorialreviews>
    <editorialreview>
      <source>Product Description</source>
      <content>One of the &lt;B&gt;Mog the Cat&lt;/B&gt; Board Books.</content>
    </editorialreview>
  </editorialreviews>
  <images>
    <image>
      <url>http://ecx.images-amazon.com/images/I/514Y3NCWBTL._SL160_.jpg</url>
      <height>160</height>
      <width>159</width>
      <imageCategories>
        <imagecategory>primary</imagecategory>
      </imageCategories>
    </image>
  </images>
  <creators>
    <creator>
      <name>Judith Kerr</name>
      <role>Author</role>
    </creator>
  </creators>
  <blurbers/><dedications/><epigraphs/><firstwords/><lastwords/><quotations/>
  <series><seriesitem>Mog (13)</seriesitem></series>
  <awards/><characters/><places/>
  <subjects><subject>Mog (Fictitious character : Kerr)</subject></subjects>
  <tags>
    <tag count="2">p</tag><tag count="1">children's</tag>
    <tag count="1">picture book</tag><tag count="2">pz</tag>
    <tag count="1">mog</tag>
  </tags>
  <similarproducts><similarproduct>0001374761</similarproduct></similarproducts>
  <browseNodes>
    <browseNode id="4">Children's Books</browseNode><browseNode id="1000">Subjects</browseNode>
  </browseNodes>
</book>
```


Figure 4. A Sample XML Document from the SBS Corpus

/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

Table 3. Tags Removed as Part of Scrubbing Process

3.1.2.3 Parsing

Once scrubbing is done, we are left only with tags that have potentially useful content. The process of structuring the document such that the correct content is connected to each node on a per tag basis is called parsing. In parsing, the entire document is represented as a tree with each tag identifying the node. We use XPath, which identifies each tag's location in an XML document, and the LibXML parser to gain access to the content associated with that respective tag.

Experiments were conducted on six different parses to see which parses were effective in retrieving the most relevant documents. The six parses generated were: full, Amazon, LibraryThing, professional, social, and title parses. The full parse had content from all the tags in the document after scrubbing. The Amazon parse included tags that came from the Amazon website; the LibraryThing parse included tags that came from the LibraryThing website. The Professional parse include all the professional tags like classification labels and Dewey codes [26]. Social tags included user-generated content like reviews, ratings and tags. The title parse included only the title of the document. The parses and the tags included in each of them are presented in Table 4.

XPath	Title	Professional	Social	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/characteritem	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. Details of Tags Present in Six Different Parses

3.1.2.4 Query Processing

The queries that are a part of the dataset have a specific format (see Section 2.3.1). The 2013 SBS track had 380 queries in the query set. The XML tags, title, mediated query, group, and narrative contain rich content for retrieval purposes (see Section 3.1.2.5.1). Six combinations of these tags were considered as input for indexing. The process of analyzing and deciding upon these six query sets required some background research [1,5]. The six query sets considered and processed were: title (T), query (Q), title-query (TQ), title-query-group (TQG), title-query-narrative (TQN), title-query-group-narrative (TQGN). These different query sets are given as one of two inputs to the indexing phase.

3.1.2.5 Indexing using Indri

The third stage of the architecture takes as input the six different parses (Section 3.1.2.3) and the six different query sets (Section 3.1.2.4). Indexing produces indices used for retrieval. We use Indri for this purpose.

3.1.2.5.1 The Indri Search Engine

Indri [14] is an open source search engine, a part of Lemur project maintained by Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts, Amherst, and the Language Technologies Institute (LTI) at Carnegie Mellon University. It uses probabilistic indexing. It is flexible and can parse documents in the PDF, HTML, XML, and TREC formats. It also provides a framework for field and passage retrieval.

Language modeling and inference-based probabilistic networks form the basis of the retrieval model in Indri [15,16]. The relevance of a document to a query is calculated by the number of related query concepts that occur in the document. Retrieval in Indri can also be restricted to a particular field (helpful to us in using the "text" field to index all the document content as an entity).

3.1.2.6 TREC Evaluation

Once the indexing phase is done, Indri retrieves a rank-ordered list of 1000 documents for each query. Results are converted to TREC format and evaluated using a script provided by INEX, thus enabling the teams to compare results. (Metrics and evaluation details are presented in Section 2.3.3.)

The TREC metrics evaluation script takes as input the documents retrieved along with their associated work IDs. A work ID for a document is used to identify the book (but is not unique to that book). Different editions of the same book have the same work ID. However, results produced by Indri are represented by ISBN numbers. An ISBN number unlike the work ID is unique to the book. An ISBN to a work-id mapping is a one-to-many mapping. To convert the ISBN numbers to their corresponding work-ids, INEX provides a Perl script called "deduplicate.pl" that takes as input a file containing the mappings provided by INEX (as in Figure 5).

ISBN	Work-id
0030843278	6
0675076455	7
1582099855	14
0681047992	14
0843111577	16
0440428130	17
0330308297	17
0027116905	17
0590072242	17
0395732387	17
0439044588	17
0434930210	17
1561378224	17
0316490059	18
0153536004	19
1562470817	23
0590677292	23
1562470825	23
0785740910	23
….	

Figure 5. Excerpt from an ISBN to Work ID Mapping File

Once the conversion is done, TREC evaluation can be run using the QRels provided by INEX as input to obtain the final set of traditional TREC evaluation results. A team's position in the INEX official rankings is determined by the nDCG@10 metric. Experiments conducted and results obtained are presented in detail in Chapter 4.

3.1.3 Traditional system - 2014 SBS Track

Modifications were required as we moved to 2014. The 2014 dataset has almost twice as many topics (680). These topics include the profile of the user posting the topic along with other related data. Including profiles as part of the topics increases the amount of metadata associated with a topic, making it favorable for the recommender system; (comparison of 2012 SBS results with 2013 SBS results shows an improvement with the same methodology and corpus but richer and larger topic set [5,25].) Information about the methodology applied, the experiments performed and results obtained on the 2014 data can be found in [24].

3.2 Architecture of the Recommender System

The second part of our system is the recommender system, in which results obtained from the traditional system are re-ranked based on "similar users" (collaborative filtering) to produce recommended results. The recommender system was designed to make use of information from the users "similar to" the user who posted the query. This idea stems from assuming that similar users tend to have the same preferences and taste in books. The architecture of the recommender system is shown in Figure 6. Generation of matrices, similar users and new recommended scores are the three main stages in the recommender system. These are followed by tuning of the λ parameter. Each stage is presented in the sections that follow.

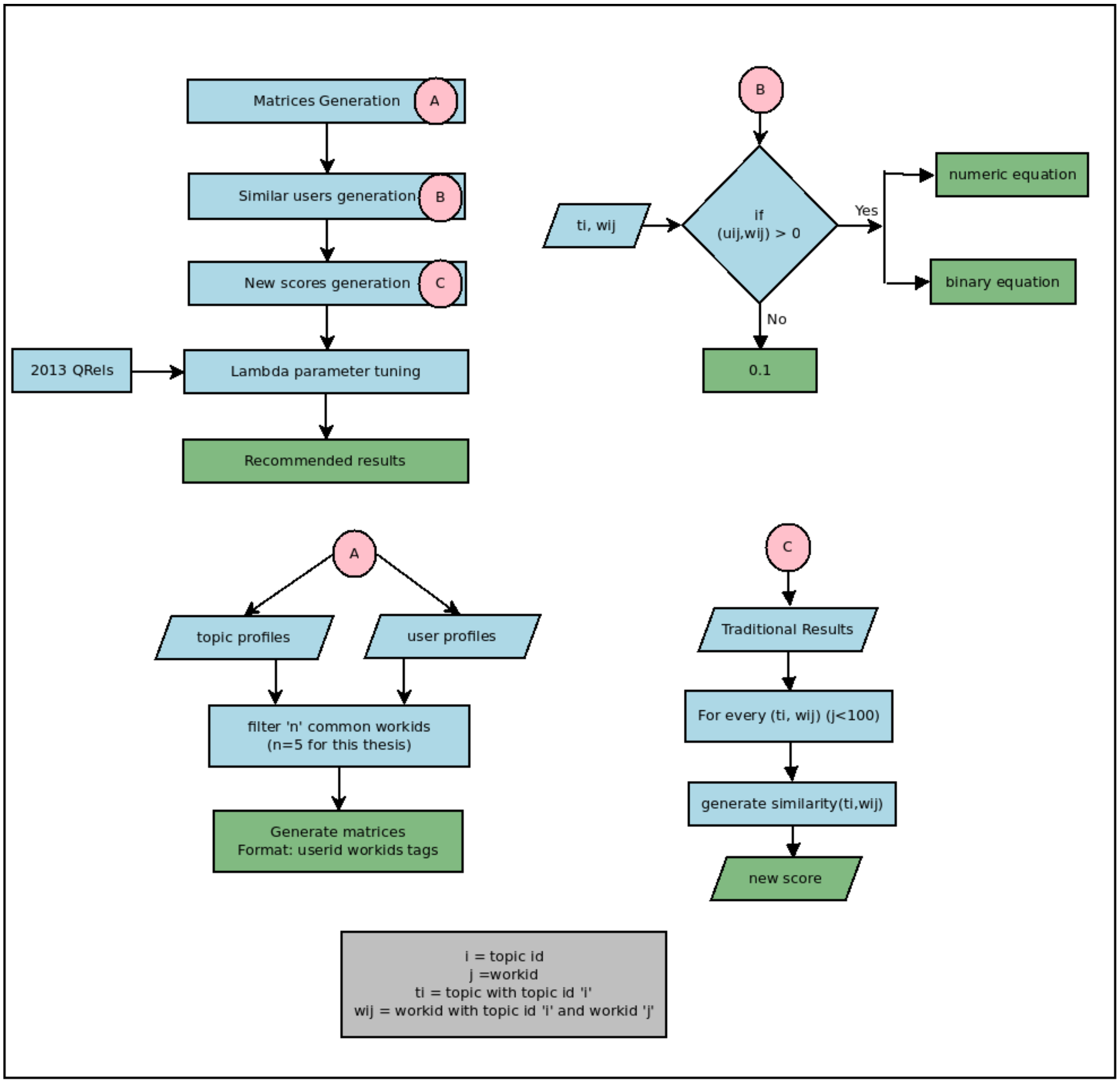

Figure 6. Architecture of Recommender System

3.2.1 Matrix Generation

The first step of the recommender system involves generating matrices with features (work IDs and tags) for each topic in the topic set. The features are selected with the idea that the same books (workids) and same genres (tags) are a measure of similarity between users. Four different matrices (see Table 5) are generated.

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. Matrix Representation

The focus of this research is the bin_num matrix. (Details of the other three can be found in [23,24].) A matrix is generated for each topic in the topic set. In the bin_num matrix, work IDs that occur in common between the user (his profile) and the profiles of other anonymous people [94,000 user profiles are provided by INEX for experimental purposes] are given a value of 1. If the work ID is not common to both, it is given a value of zero. Only if there are at least 5 work IDs in common between the user and anonymous profiles, are they compared (considered ”similar”). (I.e., n=5 as shown in Figure 6.)

A major obstacle faced during the generation of matrices for the 2013 SBS track was the poor quality of the dataset. For example, data in the XML tags ‘author’ and ‘publishing date’ were switched (placing them in the wrong tags altogether). The content had to be cleaned, formatted and then retrieved for better quality. This obstacle was overcome using the power of regular expressions (a part of the scripting language ‘Perl’).

3.2.2 Generating similar users

Once the matrices are generated, the next step is to generate a list of similar users based on the context vectors. Pairwise cosine similarity (as in Figure 6) is used for this purpose. Once the similarity score is calculated using different equations for both binary and numeric values (see Figure 6), the top-ranked 50 and 100 “similar users” are considered the sets of interest.

3.2.3 Generating Recommender Scores

After the similar users are calculated for each topic, we are ready to generate new scores (Part C, Figure 6.). In this stage, we calculate the score, i.e., the contribution made by the recommender system. Two metrics proved to be worthy after conducting experiments with different versions of equations. The two successful metrics are presented in Table 6.

Metric	Binary Score	Numeric Score
Metric 1 (DCG-style)	$R_{ij} = \sum_{k=1}^{50/100} \frac{S_{ik}+1}{\log_2^{rank}+1}$	$R_{ij} = \sum_{k=1}^{50/100} \frac{S_{ik}+r_{jk}}{\log_2^{rank}+1}$
Metric 2 (MRR- style)	$R_{ij} = \sum_{k=1}^{50/100} \frac{S_{ik}+1}{rank}$	$R_{ij} = \sum_{k=1}^{50/100} \frac{S_{ik}+r_{jk}}{rank}$
i = topic id j = work ID 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 the Contribution of the Recommender System

Metric 1 employs a DCG-style of calculating the new score and Metric 2 employs the MRR style. This calculation of the contribution by the recommender system takes as input 1) the rank-ordered list of similar users, 2) the similarity score of each user, 3) the user rating for each work ID as identified by traditional retrieval and, 4) the count of similar users that have the same work ID in their respective catalogs. Other metrics were considered but proved unsuccessful or inferior to these two metrics.

3.2.4 Generating the Final Score

Once the traditional scores and the recommender scores are calculated, a linear combination of the two scores produces a final score. The parameter used to produce a linear combination has to be tuned effectively for good results. Tuning requires QRels provided by INEX. The 2013 QRels that were already available to us were used for tuning λ. The parameter was given an initial seed value of 0.0001855 based on [1]. This value was later fine tuned to produce better results. 2013 λ value was used in 2014 experiments because the 2014 QRels were unavailable prior to official submission of results to INEX. Details about our 2014 experiments can be found in Chapter 4.

4 Experiments and Results

This chapter presents details of experiments based on the combination of the traditional and recommender system methodology. The experiments conducted on the 2011 SBS track can be found in [23]. The 2011 track helped formulate a good foundation for the architecture of our traditional system. This thesis focuses on experiments conducted on the 2013 dataset that form the basis for experiments on the 2014 dataset [9,10,11]. We also present the results obtained from the 2014 dataset. These results (2014) are again presented in two parts: the official results submitted to INEX 2014 and the improved results produced upon access to the 2014 QRels.

4.1 SBS 2013 Experiments and Results

After running experiments on the 2011 dataset [23], we developed a good understanding of the traditional system. A problem, however, was the quality of the dataset provided. 2013 data was much cleaner than that in 2011.

4.1.1 Retrieval Methodology

Our retrieval methodology combines aspects of retrieval and recommendation; an overview is presented here. A more detailed explanation follows.

Step 1: Scrubbing the SBS 2013 corpus (containing 2.8 million documents from Amazon and LibraryThing [see Section 3.1.2.2])

Step 2: Parsing the scrubbed documents (six combinations [see Section 3.1.2.3])

Step 3: Processing the XML query files (six different combination sets [see Section 3.1.2.4])

Step 4: Indexing the documents using Indri both with and without feedback (36 combinations without feedback and 36 combinations with feedback for a total of 72 retrievals).

Step 5: Retrieve the 1000 top ranked documents for each index and query set.

Step 6: Redirect the ranked lists into unique topic files of their own
E.g.: The file containing ranked lists for topic-ids 1, 2 and 3, upon redirection, produces three files (1.txt, 2.txt and 3.txt) with each file containing its corresponding ranked lists.

Step 7: Produce the context matrices used to determine similar users. Four different matrix representations are considered with cosine as the similarity measure.

Step 8: Those redirected files from Step 6 with no common tags or work IDs in their matrices (generated in Step 7) are removed. These files are not re-ranked due to lack of commonality.

Step 9: Generate "similar users" (50/100 per query), each correlated with the query via cosine to produce a rank ordered list of "recommender scores."

Step 10: Produce the final scores by combining the traditional scores (Step 5) in a linear combination (using parameter λ) with the recommender scores (Step 9).

Step 11: Submit to TREC evaluation the final re-ranked list of documents and INEX Qrels to produce nDCG@10 values.

This algorithm is simple yet complicated in its own way. The number of parameters, values and thresholds set at every step make the process difficult to optimize. (Changing just one value may lead to a very different result.) Yet the number of experiments must be feasible. We address this issue below.

4.1.2 Traditional System

For the first experiments conducted in the 2013 traditional system, the data was scrubbed, parsed and indexed. The six different indices used for this purpose were inspired by [1]. These six indices against six different

```
<!--Ref:http://lemur.sourceforge.net/indri/IndriRunQuery.html -->
<parameters>
   <index>/smart/IR2014/Indexes/Amazon</index>
   <count>1000</count>
   <rule>method:dir,mu:2500</rule>
   <runID>SBS_Indri_mu_2500</runID>
   <inex>
         <particpantID>65</particpantID>
         <task>social book search track</task>
   </inex>
   <fbDocs>10</fbDocs>
   <fbTerms>50</fbTerms>
</parameters>
```

Figure 7. Sample Indri Document Index File (with Pseudo Feedback Parameters)

query combinations (Section 3.1.2.4) produced 36 retrievals. For all 36 combinations, pseudo feedback was added as an option (in all cases with 10 documents and 50 terms [1,16]). Pseudo or blind feedback expands the query, using terms [here 50] from the top ranked documents [here 10] returned by the query. This expanded query is used with the aim of retrieving documents that are more similar to the query than those from the first pass. Indri allows this option. A sample Indri document indexing file is shown in Figure 7.

Indri includes a parameter for smoothing [27] (removing zero probabilities) in the result. The smoothing by default is the Dirichlet method with a Mobius function value of 2500 (Figure 7, XML tag rule). Based on papers from earlier, successful research [1,2,4,6], we chose Dirichlet smoothing for our framework.

By adding the option of feedback, the total number of combinations reached 72. When TREC evaluation was performed on all 72 combinations, it was observed that the full and social indices performed best and second best, respectively. This result conforms with the results obtained by the best performing team in 2012 [1]. Thus 24 cases (full and social parses, six query combinations, with and without pseudo feedback [2x6x2]) were chosen for future experimentation. A sample file showing the ranked list of documents and their relevance scores (produced by Indri) is shown in Table 8. The format of the file is shown in Table 1.

```
89218   Q0   2007755   1    -4.3973    UMD_2013_SBS_Indri_dir_2500
89218   Q0   3760      2    -4.85693   UMD_2013_SBS_Indri_dir_2500
89218   Q0   6906698   5    -4.91525   UMD_2013_SBS_Indri_dir_2500
89218   Q0   7177288   6    -4.91525   UMD_2013_SBS_Indri_dir_2500
89218   Q0   29874     9    -5.14775   UMD_2013_SBS_Indri_dir_2500
89218   Q0   7592742   10   -5.14817   UMD_2013_SBS_Indri_dir_2500
……
……
131328   Q0   66332     144   -5.83928   UMD_2013_SBS_Indri_dir_2500
131328   Q0   5686164   145   -5.84236   UMD_2013_SBS_Indri_dir_2500
131328   Q0   7235416   146   -5.84285   UMD_2013_SBS_Indri_dir_2500
131328   Q0   1101277   147   -5.84695   UMD_2013_SBS_Indri_dir_2500
……
```

Figure 8. Sample Traditional Results File (1000 Documents per Topic_id)

4.1.3 Recommender System

The second stage of our system, the recommender system, takes as input the top 1000 documents retrieved by each query. The top 100 of these results are now re-ranked, based on a linear combination of the document score with the newly generated recommender score.

The first step in generating the recommender scores is to generate the context matrices (see Figure 6) in order to find "similar users." Two feature sets (work IDs and tags) were selected. Work IDs and tags were chosen because people with the same books and genres in their catalog may be considered "similar." The context vectors thus generated may have either binary or numeric values (see Table 5 for matrix representations). So a row in the matrix corresponds to a profile, and the columns represent the values of the work-ids and tags in that profile.

We used cosine to generate the similarity scores between the user and the profiles of others. A threshold value of 5 is selected to reduce the number of rows in the matrices; i.e., at least 5 work IDs must be in common between the user and the profile before that profile is added as a row in the matrix. The threshold value is set at 5 because at a higher level, important profiles were being lost and at a lower level, trivial information was being added.

Based on the similarity of the user's profile with each row of the matrix a new score reflecting the contribution of the recommender system is calculated. Here we used a self-designed metric. The metric used in 2013 was changed in 2014 to improve the results. (See Table 6 for these metrics). The two metrics applied to calculate the contribution of the recommender system are used for both binary and numeric values. Two classes of "similar users" are considered, namely, 50 and 100. Decreasing this value below 50 did not produce a substantive difference in the re-ranking process, and increasing the value above 100 produced the same or worse results. Once the recommender scores are generated, a linear combination of these scores with the traditional scores is formed. The parameter used to produce the linear combination is λ. This parameter must be tuned to get good results. Because the 2014 QRels were unavailable, the value of λ was entirely dependent on experiments conducted on the 2013 dataset. λ was given an initial seed value of 0.0001855 which was based on the results of the winning teams at INEX SBS 2012 [1,2]. This seed value, however, did not produce the expected results for our methodology. In tuning λ, the first interval taken was from 0.0001755 to 0.0001955 in steps of 0.00001. The results produced for both 50 and 100 similar users were recorded. Tuning continued until the "best" value of λ was determined. Results of these experiments, including λ, the

Indexing type	Query combination	Pseudo feedback (terms =50, documents =10)	λ value	Similar users	nDCG@10
Full	title (T)	applied	0.000260	100	0.0823
Full	title-query (TQ)	applied	0.0000247	100	0.0985
Full	title-query-group (TQG)	applied	0.0000227	50	**0.1044**
Full	title-query-group (TQG)	applied	0.0000255	100	0.1043
Social	title-query-group (TQG)	applied	0.0000222	50	0.0985

Table 7. Value of λ that Produces Best nDCG@10 Value

number of similar users, and the corresponding nDCG@10 values are shown in Table 7.

It was observed that the full and social indices along with T, TQ, and TQG query combinations produced the best results. Based on these experiments and the λ values obtained, the runs for the INEX 2014 SBS track were submitted. λ could not be tuned to the 2014 dataset because QRels were not available before the submission of the results.

4.2 SBS 2014 Experiments and Results

Our IR team participated in the 2014 INEX SBS track, for which background work for the 2011-2013 SBS tracks was required. The official 2014 submissions allowed participant teams to submit up to six runs including one based only on the title (T). The 2014 data set was larger and richer in content; it contained 680 queries, a corpus of 2.8 million documents from Amazon and LibraryThing, and a set of 94,000 profiles for experimental purposes. See [3,24] for details.

4.2.1 Official INEX Results 2014 SBS Track

Since we did not have access to the 2014 QRels, we used the λ value that was tuned for the 2013 dataset. This was a risk, because the 2014 data is much different from that in 2013. However, all other results and outcomes associated with the 2014 data set conform with those produced using the 2013 data set. E.g., the full and social indices along with the TQG query combination again produced the best results. These results (presented in

Table 7) were submitted as the official runs of our IR team to the 2014 INEX SBS track. In the official results we placed 20 in terms of nDCG@10 and 19 in terms of R@1000. Our official results are presented in Table 8.

Indexing type	Query combination	λ value	Similar users	nDCG@10	MRR	MAP	R@1000
Full_fb	T	0.000260	100	0.070	0.139	0.047	0.253
Full_fb	TQ	0.0000247	100	0.092	0.176	0.064	0.321
Full_fb Full_fb	TQG TQG	0.0000227 0.0000255	50 100	0.097 0.096	0.188 0.188	0.069 0.068	0.328 0.328
Social_fb	TQG	0.0000222	50	0.096	0.184	0.067	0.327
Traditional_full_fb	TQG	-	-	0.095	0.185	0.068	0.328

Table 8. Official INEX Results SBS Track 2014

4.2.2 Improved Results 2014 SBS Track

When given access to the 2014 QRels, we re-examined our methodology to improve both our traditional and recommender systems by changing and tuning the parameters for the 2014 data set.

4.2.2.1 Revisiting Feedback Parameter Values (2014)

Feedback values for the number of documents and terms were re-visited with the aim of improving the recall of the traditional system. We changed the values of **t** (number of terms) and **d** (number of documents), with **d** ranging from 5 to 15, and **t** ranging from 15 to 50. R@1000 improves from 0.328 (at **d**=10 and **t**=50) to 0.380 (at **d**=10 and **t**=15) as seen in Table 9. We use this retrieval run as the basis of our improved current 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 9. Results of Traditional Retrieval (Full Index, TQG Query Set with Pseudo-feedback)

4.2.2.2 Revisiting the λ parameter values (2014)

Since there was improvement in the traditional system (seen in Table 9), we decided to change parameters in the recommender system as well to see if we could improve our official results. Since λ was tuned to the 2013 dataset (due to unavailability of 2014 QRels), upon access to the 2014 QRels we tuned λ again.

Some observations are: (1) The matrix representation bin_bin (Table 5) is found to produce a better result than bin_num that was used for the official results. (2) Two new metrics, successfully used here, are defined in Table 6. The results obtained for the newly tuned 2014 λ values are presented in Table 10. Our current best result (0.1058) ranks 17 in terms of nDCG@10 and 13 in terms of R@1000 when compared to the INEX 2014 official results. The significance results for the track have not been released yet so it is hard to estimate our correct position in the official INEX results.

Metric	Feature	Users	λ	nDCG@10	MRR	MAP	R@1000
Metric 1	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 2	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 10. Final (Improved) Results of the Recommender System

5 Conclusions and Future Work

The 2014 results (both official and improved) show that even though the nDCG@10 values were fairly good, improvement in the traditional system may yield as good or better results. It is observed that among all the indices generated as part of the indexing phase in the traditional system (Section 3.1.2.5), full indexing, including content from all the parsed tags, produces the best results. This shows that the more content, the more relevant documents are captured. The best query combination is the title-query-group (TQG) combination. [see in Table 7]. The matrix feature considered here (bin_num in Table 5) proved second best when compared to the bin_bin feature combination [24]. Considering binary values for both tags and work IDs in the context vectors produced the better result. Of the number of similar users considered, 50 vs. 100, results for 50 users were better. Of the two metrics used to find the contribution of the recommender system (Table 6), metric 2 produces a better result.

Upon analysis, we found that at least two thirds of the relevant documents present in the QRels provided by INEX were not being retrieved by the traditional system. If we increase the number of relevant documents retrieved, the entire system should produce a better nDCG@10 value. As a first attempt at the Social Book Search Track, we found the background and experimental work done on this task an excellent learning experience.

For future work, improvements can be made on both the traditional and recommender systems. Possibilities for the traditional system include: (1) Using feedback from data in the topic user's catalog (eg., title corresponding to work IDs and tags assigned by the user) to expand the query. With a longer and more informative query, one can expect to retrieve more relevant books. (2) We can provide structure-based weights to tags while retrieving data from Indri. For example, the tag ‘review content’ has information that is richer in content than a tag like ‘publishing date.’ So the former may be given more weight than the latter. (3) We can use phrase detection and Named Entity Recognition (NER) techniques on the query to detect content bearing terms and can provide extra weight to those terms.

Suggested changes to the recommender system are: (1) When generating similar users, include Amazon “similar books” as the features of matrices for each book cataloged. (2) We could also include terms that occur in the titles of cataloged books in the feature vectors of the matrices. Improving retrieval in these areas holds promise for improving the final nDCG@10 values and hence overall track results.

References

[1] 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.

[2] 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.

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

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

[5] 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.

[6] 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.

[7] Salton, G. and McGill, M. *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1983.

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

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

[10] About LibraryThing [Internet]. Available from: https://www.librarything.com/about

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

[12] Accomazzi, A., Murtagh, F., & Rasmussen, B. F. Information retrieval tools and techniques, *Vistas in Astronomy*, 39, pp. 235-241, 1995.

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

[14] 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.

[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] Lavrenko, V. and Croft, B. Relevance-Based Language Models, *SIGIR'01*, pp.120 -127, September 9-12, 2001.

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

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

[19] TREC: Text REtrieval Conference [Internet]. To encourage research in information retrieval from large text collections. Available from: http://trec.nist.gov/

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

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

[23] Ravva, V.R. Personalized Book-Retrieval System Using Amazon-LibraryThing Collection, MS Thesis, University of Minnesota Duluth, 2014.

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

[25] Koolen, M., Kazai, G., Kamps, J., Doucet, A., and Landoni, M. Overview of the INEX 2011 books and social search track, *In Focused Retrieval of Content and Structure*, pp. 1-29, Berlin, Heidelberg.

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

[27] 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.