Social Book Search Reranking with Generalized Content-Based Filtering

Bo-Wen Zhang[1] Xu-Cheng Yin*[1] Xiao-Ping Cui[1] Jiao Qu[1]
Bin Geng[1] Fang Zhou[1] Li Song[2] Hong-Wei Hao[2]
[1]Department of Computer Science and Technology, School of Computer and Communication Engineering,
University of Science and Technology Beijing, Beijing 100083, China
[2]Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
bowenzhang.ustb@gmail.com, xuchengyin@ustb.edu.cn

ABSTRACT

Semantically searching and navigating products (e.g., on Taobao.com or Amazon.com) with professional metadata and user-generated content from social media is a hot topic in information retrieval and recommendation systems, while most existing methods are specifically designed as a purely searching system. In this paper, taking Social Book Search as an example, we propose a general search-recommendation hybrid system for this topic. Firstly, we propose a Generalized Content-Based Filtering (GCF) model. In this model, a preference value, which flexibly ranges from 0 to 1, is defined to describe a user's preference for each item to be recommended, unlike conventionally using a set of preferable items. We also design a weighting formulation for the measure of recommendation.Next, assuming that the query in a searching system acts as a user in a recommendation system, a general reranking model is constructed with GCF to rerank the initial resulting list by utilizing a variety of rich social information. Afterwards, we propose a general search-recommendation hybrid framework for Social Book Search, where learning-to-rank is used to adaptively combine all reranking results. Finally, our proposed system is extensively evaluated on the INEX 2012 and 2013 Social Book Search datasets, and has the best performance (NDCG@10) on both datasets compared to other state-of-the-art systems. Moreover, our system recently won the INEX 2014 Social Book Search Evaluation.

Categories and Subject Descriptors

H.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval—*Search process*

General Terms

Algorithms; Design; Experimentation

Keywords

Social book search; search-recommendation hybrid system; generalized content-based filtering; learning to rank

1. INTRODUCTION

With the development of e-commerce and social networking services, the information of online products and items is acquired, shared and published in fundamentally different ways [24, 13]. Apart from the professional metadata, i.e., information comes with products, there is also wealthy user-generated content (e.g., ratings, reviews and tags) that describes the properties, content and attributes of products. Therefore, how to utilize such information and support users in semantic searching and navigating these products from social media becomes a hot topic [11, 28, 27]. As the dominant source of information, the book is a classical type of the online products with rich professional metadata and user-generated content. As a result, in this paper, we take Social Book Search as an example for semantically searching and navigating products with rich social information (e.g., on Taobao.com or Amazon.com).

Social Book Search Instantiation

Social Book Search is a typical track in INEX Evaluation since 2011 [1]*. It is a task to satisfy the users' requests about book recommendation from the Amazon book collection with a variety of social information (see one example book document in XML 2). In this task, the input (i.e., the user's semantic request) is a query in natural language (called as "topic", see examples in XML 1) which comes from LibraryThing.com. The output is a suggested book list. In the track, the output results are evaluated with the relevance from a ground truth set (composed by manually annotated recommendations) [14].*

On one side, this suggestion task is mainly a searching problem with queries (topics), documents (book collection), and results (by ranking). However, there are a variety of rich social information (e.g., reviews, ratings, and tags) in the document collection, and how to fully use the social information in searching is challenging. Furthermore, conventional searching systems did not have an encouraged performance in Social Book Search Track [15, 16]. On the other side, the

*Corresponding author: Xu-Cheng Yin.

[1] The Social Book Search definition was proposed by INEX organizers, which can be referred to `https://inex.mmci.uni-saarland.de/tracks/books`.

CIKM'14, November 3–7, 2014, Shanghai, China.
Copyright 2014 ACM 978-1-4503-2598-1/14/11 ...$15.00.
http://dx.doi.org/10.1145/2661829.2661940.

target of this task is book recommendation, i.e., suggesting ranked books following the user's request. To some extent, it is likely a recommendation system. Some participating systems with recommendation tricks won the recent Evaluation [17]. However, conventional recommendation systems can't deal with semantic queries in natural language.

XML 1: A query XML file with two example topics.

```
<topics>
  <topic id=”1116”>
    <query>introduction book to Lisp</query>
    <title>Which LISP?</title>
    <group>Purely Programmers</group>
    <narrative>It'll be time for me to ...</narrative>
  </topic>
  <topic id=”1196”>
    <query>books about work for Peace Corps </query>
    <title>The Best Peace Corps Novel</title>
    <group>Returned Peace Corps Readers</group>
    <narrative>I'm looking for people's...</narrative>
  </topic>
</topics>
```

XML 2: A document XML file with an example book.

```
<book>
<isbn>0007175000</isbn>
<title>Mister Monday</title>
<manufacturer>Harper Collins</manufacturer>
<publisher>Harper Collins</publisher>
<dewey>823.914</dewey>
<numberofpages>361</numberofpages>
<reviews><review>
    <date>2003-10-06</date>
    <summary>So good, you can't put it down!</summary>
    <content>Now, I had...</content>
    <rating>5</rating><totalvotes>7</totalvotes>
    <helpfulvotes>7</helpfulvotes>
</review></reviews>
<tags><tag count=”240”>fantasy</tag>
<tag count=”9”>children's literature</tag></tags>
<similarproducts>
    <similarproduct>0439436559</similarproduct>
</similarproducts>
<browseNodes>
    <browseNode id=”4”>Children's Books</browseNode>
</browseNodes>
...
</book>
```

In Social Book Search, semantic understanding of the query is highly complex because the abilities and habits of expression vary from one to another. Moreover, there is great difficulty in dealing with a mixture of professional and social metadata, which differ both in quality as well as in kind. Usually professional metadata is often based on controlled vocabularies, while social metadata come in the form of reviews that vary widely in length, opinion, clarity, seriousness and in the aspects of the book they discuss [15, 16, 17].

Existing methods for Social Book Search can roughly be categorized into three groups: Query and Documents Processing methods, Sequential Dependence Model (SDM) based methods and Social Feature Reranking methods [16]. Query and Documents Processing methods focus on the frequencies of relevant words. Some techniques are proposed to remove the noise of queries, and other methods are designed to duplicate relevant information or bring meaningful information in documents [4]. These methods improve the performance of searching engine but they are still influenced by the limitation of mathematical models in searching. Sequential Dependence Model considers the linguistic model of queries [6]. According to the INEX Social Book Search Evaluation, the SDM based systems are not as stable as other participating methods. The first two categories actually construct a searching system (usually with a search engine) for Social Book Search. As we know, it is difficult to use a variety of social information in the book collection in conventional searching systems. Recently, Social Feature Reranking methods have become the focus of several recent works. These methods make use of the structure of documents collection with social information. The similarities of books are calculated on some particular fields extracted from documents XML [20]. Based on the similarities, the searching results are reranked with a mathematical model. Social Feature Reranking (the third category) can be seen as searching with recommendation tricks, but there is no unified framework to perform reranking and combination with various social information for Social Book Search.

As described before, this suggestion task is mainly a searching problem with semantic queries (topics). At the same time, the task with the user's request is likely a recommendation system for book recommendation. In this paper, we argue that the Social Book Search task can be seen as a search-recommendation hybrid system.

In order to exploit recommendation techniques for social information, we firstly propose a Generalized Content-Based Filtering (GCF) model. As we know, Content-Based Filtering model needs a set of preferable items to describe the preference of users [22]. On the contrary, in our proposed GCF model, a preference value is defined for each item which ranges from 0 to 1. Another difference is the measure of recommendation. The conventional model takes the sum of similarities with items in the preference set while GCF takes the weighted sum of the mentioned preference value and the influence value from the conventional model. This Generalized Content-Based Filtering model can adaptively use a soft preference (ranging from 0 to 1) of items, and intelligently merge different recommendation values with a weighted combination.

Based on this new GCF model, we then construct a unified Reranking Model by utilizing various social information by reranking the initial resulting list. By assuming the queries for searching act as the users in recommender systems [9], we design Social Book Search Reranking as a special case of recommender system. This new unified Reranking Model can easily utilize a variety of rich social information (e.g. ratings and tags) for semantically searching and navigating web products (books).

Next, taking Social Book Search as an example, based on this unified Reranking Model, we build a general search-recommendation hybrid system for Social Book Search. In this system, several social features are extracted from social information of documents to construct feature vectors. For each feature vector, a reranking result is obtained with the Reranking Model. Moreover, in order to make full use of all reranking results, we use learning-to-rank [7] to combine the reranking results. We select Random-Forest, Rank-Net and Coordinate-Ascent learners in the learning-to-rank strategy. The proposed searching system provides a unified framework for perform searching, reranking, merging and recommending for Social Book Search.

Finally, our proposed system is extensively evaluated on the INEX 2012 and 2013 Social Book Search datasets, and has the best performance (NDCG@10) on both datasets compared to all participating systems. Moreover, our system recently won the INEX 2014 Social Book Search Evaluation.

The rest of the paper is organized as follows. Related work is discussed in Section 2. Section 3 introduces the Generalized Content-Based Filtering (GCF) model. Our proposed Social Book Search system is presented in Section 4 in detail. Section 5 describes extensive experiments on INEX Social Book Search Track datasets. We draw a conclusion in the last section.

2. RELATED WORK

2.1 Social Book Search

As described above, existing models for Social Book Search can be mainly categorized into three groups, namely, Query and Documents Processing methods, Sequential Dependence Model based methods and Social Feature Reranking methods. Query and Documents Processing methods and Sequential Dependence Model based methods actually construct a purely searching system for Social Book Search, while Social Feature Reranking methods combine several recommendation tricks. In this section, we will review several main competitive technologies. For other related methods, interested readers can refer to the INEX Social Book Search Evaluation proceedings [15, 16, 17].

First, as referred to INEX Social Book Search Track, queries are collected from a forum LibraryThing.com and document collections are crawled from Amazon. As a result, queries always appear in the form of xml documents made up by fields <title>, <narrative>, <query> (marked manually by professional experts), <group>. Query processing methods focus on different roles of the four fields. So different combinations are used as inputs of searching engine to search for books. As mentioned above, document processing methods manage to enrich the documents. Numeric information is replaced with textual information, e.g., <xxx count="2">flag</xxx> is replaced with <xxx count="2">flag flag</xxx> [4].

Second, Sequential Dependence Model based methods focus on the sentence structure of queries. Single term feature, exact phrase feature and unordered window feature of queries are retrieved separately and the searching results are combined with Equation (1) [23], where the feature weight is set according to the default setting and f_T, f_O, f_U are the log maximum likelihood estimates of query terms in document D, computed over the target collection with a Dirichlet smoothing.

$$\begin{aligned} SDM(Q,D) = \lambda_T \sum_{q \in Q} f_T(q,D) + \lambda_O \sum_{i=1}^{|Q|-1} f_O(q_i, q_{i+1}, D) \\ + \lambda_U \sum_{i=1}^{|Q|-1} f_U(q_i, q_{i+1}, D) \end{aligned} \tag{1}$$

Third, in the Social Feature Reranking method, some user-generated content fields are defined as social features of books. Based on the fact that similar books of relevant books are useful, each social feature is extracted from the XML documents in the form of feature vectors. The fields <tags>, <ratings> and <similar-product> are concerned as social features in this model. For each social feature, the feature vectors of all documents make up to a feature matrix, in which the line vectors are the feature vectors of books. Then the similarity of each two books is calculated based on the cosine similarity of corresponding line vectors. The products of the similarity between documents i, j and corresponding initial score S_j is added to the initial score S_i, to rerank the initial results [3]. As we know, based on three social features there are three reranking results available. At last, linear combination is applied to fuse the three results if needed. According to evulation experiments, this model has a fair good performance, but suffers from two limitations: (1) how to select optimized parameters in combination and (2) how to make use of other relative fields with rich useful information. These two limitations lead to the instability of the system. For example, sometimes the combination result performs worse than the result based on single feature in the previous INEX evaluation. Consequently, a general framework is needed to be investigated for combining different searching results with various social information.

2.2 Content-Based Filtering

Content-Based Filtering model is a classical and popular model in recommendation systems. The basic idea of this model is to recommend items which match the user's preferences most [18]. Comparing with Collaborative Filtering, although extra information such as the description of items is needed, this model has a great advantage that large amount of users and their ratings of products are not necessary [1]. That is, even if there is only one user, the model can achieve the goal of recommendation.

Content-Based Filtering model focuses on descriptions of recommended items and user preferences. Usually the products in a user's purchasing record indicate the users used to like or need the products. If we assume that the preference of users is unchanged, we can define a preference set, which includes the products used to buy, as the description of the user's preference [5]. As a result, the products which meet the users' preference may be conceived as the products similar to the products in the set.

In Content-Based Filtering model, there is usually an accurate way to describe the products. Each product has a feature vector to describe the features. For textual description, usually $tf - idf$ or some numeric information in special cases is used to interpret the information. Then the similarity between two products are available based on the feature vectors. There are several methods but the cosine similarity between feature vectors are regarded as the similarities between the products in most cases [8]. In this way, the similarities between candidate products and each product in the preference set is calculated. So the products in the preference set vote for candidate products based on the sum of similarities. The more votes the candidate gets, the more possible it is recommended.

Despite the widespread application of this model, nevertheless, in this model there is a severe but not obvious pitfall that the difference of levels in preference is ignored. It is insufficient objective to mark the products in purchasing records with "like" because people always add an adverb e.g. very, a little before the word "like/dislike" in their expression. Another disadvantage of the use of record is the cold-start system. In a cold-start-system without any purchasing records, it is a difficult issue to learn the preferences of users [29]. Moreover, Content-Based Filtering model is also described as recommending items similar to items users used to like. That means if the preferences change with time, the model may recommend products irrelevant [19].

3. GENERALIZED CONTENT-BASED FILTERING (GCF)

As described above, there are some limitations in the conventional Content-Based Filtering model. In this paper, we extend the model and propose a Generalized Content-Based Filtering (GCF) model, which can adaptively and flexibly utilize the user's different preferences with a soft weight, not only using a set of preferable items. The proposed model includes the following stages:

1) *Labeling Preference Value.* We don't describe the user's preference by the purchasing record. On the contrary, we define a preference value for each product. In a recommender system, the preference value can be usually labeled by users manually, which ranges from 0 to 1. 0 stands for the default value and the larger the value, the more preferable the product is. If the users are described by user profiles or something else in the form of text, the same to the products, there is also an automatic way to label the preference value by calculating relevance between the documents and the user's feature with

$$pv(d) = \quad p(r_i|d, \mu) = \frac{f_{r_i,d} + \mu p(r_i|C)}{|d| + \mu} \tag{2}$$

where $pv(d)$ is the preference value of the document d, r_i is one of the word features of the user description, μ is Direchlet smoothing parameter and as usual, $|d|$ is the length of Document d. And $f_{r_i,d}$ represents the times that feature r_i appears in document d, while $p(r_i|C)$ shows the probability of feature r_i in the set of features. As a result, $p(r_i|d, \mu)$ stands for the polynomial distribution estimated with Dirichlet smoothing, which is a classic strategy in convetional searching engines.

2) *Extracting Features and Caculating Similarities.* In this stage, a set of features F is firstly determined according to the structure of the description of products. For each feature in F, a feature vector is extracted from the descriptions. For example, tag t is a feature in F. The tag vector of product i and j are $\overrightarrow{t_i} = [3, 0]$ and $\overrightarrow{t_j} = [0, 5]$. That means three users tag the document i with tag 1 while 5 users tag document j with tag 2. In this way, all features construct a feature matrix (Equation (3)). Each line vector in the matrix stands for the feature vector of the corresponding product.

With the matrix, we can calculate the similarity of two products i and j by calculating the similarity of the two feature vectors $f_i = [f_{i1}, \cdots, f_{im}]$ and $f_j = [f_{j1}, \cdots, f_{jm}]$. Similar to the conventional model, usually the cosine similarity is chosen, which is calculated by Equation (4) [26].

$$F = \begin{bmatrix} f_{11} & \dots & f_{1m} \\ \vdots & \ddots & \vdots \\ f_{n1} & \cdots & f_{nm} \end{bmatrix} \tag{3}$$

$$sim_{ij}(f) = \cos < \overrightarrow{f_i}, \overrightarrow{f_j} >= \frac{\overrightarrow{f_i} \cdot \overrightarrow{f_j}}{|\overrightarrow{f_i}||\overrightarrow{f_j}|} \tag{4}$$

where $\overrightarrow{f_i} \cdot \overrightarrow{f_j}$ means the dot product of two vectors, and $|\overrightarrow{f_i}|$ and $|\overrightarrow{f_j}|$ represent the norms of the two vectors.

3) *Computing Final Recommendation Value.* We defined the score calculated from the conventional Content-Based Filtering model as the influence value by Equation (5). A weight is multiplied to the similarities before they are summed together in order to emphasize the different importance of products caused by preference values.

$$inv(d_i) = \sum_{j=1}^{N} sim_{ij} \cdot pv(d_j)(j \neq i) \tag{5}$$

Considering the effect both from the preference value (pv) and from the influence value (inv), we design a general weighting formulation for the measure of recommendation inspired by Social Feature Reranking method. The weighted sum of the preference value and the influence value is defined as the final recommendation value (rv), which is calculated by Equation (6) to decide which products to be recommended.

$$rv(d_i) = \alpha \cdot pv(d_i) + (1 - \alpha) \cdot \frac{1}{N - 1} \cdot inv(d_i) \tag{6}$$

In Equation (6), a normalization coefficient $\frac{1}{N-1}$ is multiplied in order to unify the dimension. Note that only products in the top N list of the preference value are considered in Stage 2) and Stage 3).

Obviously, the proposed GCF model can be easily applied in most product recommendation systems, and the only difference is the way of feature extraction for various structures of the product descriptions. Moreover, we design a recommendation algorithm with the Generalized Content-Based Filtering model, which is shown in Algorithm 1.

Algorithm 1: GCF Algorithm.

```
Input:
    $P$: preference value set, empty set allowed.
    $D$: products set.
Output:
    $R$: set of recommended products.
Parameter:
    $pv(i)$: preference value of each product $i$.
    $inv(i)$: content-based influence value of product $i$.
    $rv(i)$: final recommendation value of product $i$.
    $vec_i, vec_j$: feature vectors of product i,j.
    $sim_{ij}$: similarity of product i,j.
    $N$: threshold to be reranked.
    $k$: number of product to be recommended.
Procedure:
    For $i = 1, 2, \cdots, N$;
      If $P$ is not empty
        set $pv(i) = p_i$default 0
      Else
        Calculate $pv(i)$ for product $i$ by Equation (2).
    End
    Choose top N products.
    For $i = 1, 2, \cdots, N; j = i, i + 1, \cdots, N$;
        Extract feature vectors $vec_i$,$vec_j$ of product $i, j$
        Calculate $sim_{ij}$ by Equation (4) with $vec_i, vec_j$.
    End
    Calculate $rv$ for all products by Equation (6).
    Choose top $k$ products added into $R$.
    Output $R$.
```

REMARK 1. *In stage 1) with GCF, we choose the manual labeling method and label the preference values of all items in purchasing records with the number* 1, *otherwise with* 0. *Then we follow the steps of the model, and set* $\alpha = 0$, *i.e.,*

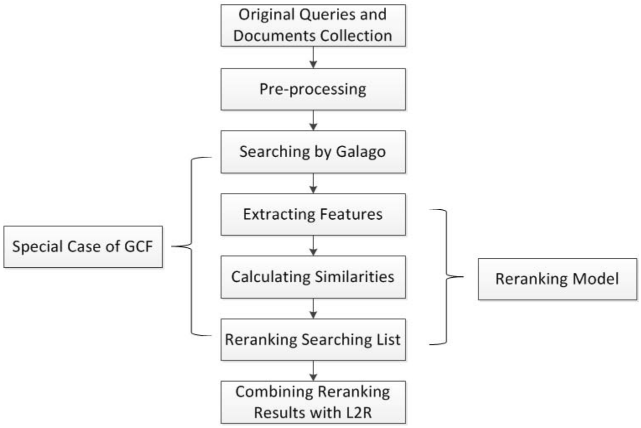

Figure 1: Flowchart of our search-recommendation hybrid system for Social Book Search.

we ignore the influence of the preference value. It is found out that in this case the procedure and the result of GCF are the same to the conventional (kNN) Content-Based Filtering model. Consequently, the conventional Content-Based Filtering model is a special case of our proposed GCF model.

4. SOCIAL BOOK SEARCH

4.1 System Overview

Based on the proposed GCF algorithm, by assuming each query act as a user of the recommender system, we construct a Reranking Model to solve the semantic searching and navigating problem of various online products by reranking the resulting list from the searching engine. According to the assumption, the content of a query can be regarded as the description of the user's preferences. In most cases, the preference values of queries cannot be easily labeled manually. Consequently, in our system, the scores given by the searching engine based on the query are regarded as the user's preference values of books. Moreover, on the basis of the assumption, we design a general strategy to use rich social information for books by the Reranking Model with GCF, which reranks the initial resulting list. Then, we construct a general search-recommendation hybrid framework for Social Book Search. Here, learning-to-rank is used to combine all reranking results on different social features and results. The flowchart of our search-recommendation hybrid system is presented in Figure 1.

1) *Pre-Processing.* As described in Section 2, Query and Document Processing methods focus on the frequencies of relevant words. According to the experiments in [4], these methods do help to improve the performance. Processing on queries may influence the description of users, while processing on documents mentioned does not influence the features of documents but improves the accuracy of the scores from the searching engine. Here, these scores are used as the preference values of documents. According to the structure of Amazon documents collection (see Section 5.1, the Experimental Data section), similar to [4], we enrich the XML documents with the professional metadata by utilizing the fields <tags>, <count> and <deway>.

2) *Searching by Galago.* In our system, we use the open-source Galago [2] as the search engine. The probability of the query content appearing in the documents is used to rank the documents with language models. The priori probabilities of documents are assumed the same so documents can be ranked by the conditional probability $P(Q|D)$. With Dirichlet Smoothing, the estimated probability is calculated by Equation (7),

$$P(Q|D) = \prod_{i=1}^{n} p(q_i|D) = \prod_{i=1}^{n} \frac{f_{q_i,D} + \mu\frac{c_{q_i}}{|C|}}{|D| + \mu} \qquad (7)$$

where $f_{q_i,D}$ stands for the amount of times the word/phrase q_i appears in document D. In this way, documents are scored and ranked by

$$\log P(Q|D) = \log \prod_{i=1}^{n} p(q_i|D) = \sum_{i=1}^{n} \log \frac{f_{q_i,D} + \mu\frac{c_{q_i}}{|C|}}{|D| + \mu} \qquad (8)$$

3) *Extracting Features.* After Stage 2), each document has a preference value for the user-query. Considering both the complexity and the accuracy, we select top N(1000 in our experiments) books to be reranked. Several features (single-feature and pairwise-feature) are selected and extracted from the particular fields in XML documents, i.e., various social information. For each feature, a feature matrix is constructed with the extracted feature vectors of all documents. The selection process and the extraction process are described in details in Section 4.2.

4) *Calculating Similarities.* As mentioned above, Equation (4) is used to calculate similarities between two documents, i.e., two corresponding line vectors in feature matrix from stage 3). In this way, the similarities between each two-books can be computed. The calculation of similarities for different (social information) features is shown in Section 4.2 in detail.

5) *Reranking with GCF.* According to the Reranking model with GCF, Equation (5) is used to calculate the content-based influence value of documents. With the preference value and the influence value, we can get the final recommendation score for each book by Equation (6). The selection of the optimized parameter α is usually decided by cross-validation on the training set. The reranking process is described in Section 4.2.

6) *Combining reranking results with L2R.* As different features are extracted from specific kinds of social information, we use the Reranking Model several times (11 times in our experiments) and get several corresponding reranking results. L2R is used as a learning tool to combine all the results. Three typical learners, i.e., Random-Forest, Rank-Net and Coordinate-Ascent, are selected as the learning models. More details are introduced in Section 4.3.

4.2 Reranking with GCF

As described before, in order to improve the results of semantic searching with a variety of rich social information, we proposed a Reranking Model (RM) based on GCF model. In this paper, we take Social Book Search as an typical example. We regard the scores given by searching engine as the preference values(pv) of books, as Equation (2) mentioned, while the scores based on reranking are the influence

[2] `http://www.galagosearch.org`

values(inv). According to the Amazon document collections, we design 11 different reranking methods: Tag-Rerank (T), Item-Rerank (I), Deep-Rerank (D), Node-Rerank (N), RatingBayes-Rerank (B), RatingReview-Rerank (R), Tag-Node-Rerank (TN), Item-Tag-Rerank (IT), Deep-Tag-Rerank (DT), Item-Tag-Node-Rerank (ITN), and Deep-Tag-Node-Rerank (DTN).

We categorize these methods into three groups. The first group contains T, N and TN. These methods are involved in the feature extraction stage. T, N match the fields <tag> and <browse-node>, which can be used as the column name of the feature vectors and the amount of the corresponding field is counted as the number of columns. The values of elements in the vector are extracted from the attribute *count*. The example of the feature matrix extraction can be referred to the example in Section 3. Moreover, the feature matrix of TN method is the connection matrix of T, N, like $[T\ N]$. With the feature matrices, Equation (4) can be used to calculate the similarities of two documents. Furthermore, the influence value can be calculated according to Equation (5) and the recommendation value can be finally calculated by (6).

The second group includes the I, D and the combined methods of I, D, T and N except TN. These methods match the field <similar-product>, which can provide a unique way to calculate similarities of two books. The field <similar-product> of a book contains a list of books marked with similar books by some particular algorithms of Amazon program, and there are at most 10 books in the list. Then for the method I, the similarity between two books is calculated by

$$sim_{ij}(I) = \begin{cases} 1, & \textit{i is j's similar product or} \\ & \textit{j is i's similar product} \\ 0, & \textit{else} \end{cases} \tag{9}$$

Considering the transitivity of similarities, the method D is related to similar products of similar products. As a result, the similarity for the method D is calculated by

$$sim_{ij}(D) = \begin{cases} 1, & sim_{ij}(I) = 1 \ \ or \\ & \exists \quad k \neq i,\ k \neq j, \\ & s.t.\ sim_{ik}(I) = sim_{jk}(I) = 1. \\ 0, & else \end{cases} \tag{10}$$

As we known, most of the similarities for the methods I and D are 0 because of the upper limit of similar products. The sparsity may weaken the performance of Reranking Model. So we propose the methods like IT, DT, ITN, DTN to solve this problem. We take the method IT as an example by Equation (11) and the other combination methods are constructed in the same way. For all methods in this group, we calculate the influence value by Equation (5) and the recommendation value by

$$sim_{ij}(IT) = \begin{cases} 1, & sim_{ij}(I) = 1. \\ sim_{ij}(T), & else \end{cases} \tag{11}$$

The third group includes R and B, which integrate two classical methods into the Reranking Model. These methods are separated from other methods because they have special ways to calculate the influence value. The method R is based on the idea that if a book has a lot of reviews and if its ratings are generally good, then it must be a very good book. The influence value of this method is calculated by

$$inv(d_i) = log(|reviews(d_i)|) \times \frac{\sum_{r \in R_{d_i}} r}{|reviews(d_i)|} \times pv(d_i) \tag{12}$$

where R_{d_i} is the set of all ratings given by users for the book d_i, and $|reviews(d_i)|$ is the number of reviews.

At the same time, B focuses on the idea that as more users rate the same work, the average becomes more reliable and less sensitive to outliers [12, 21]. The Bayesian averaging of ratings takes how many users have rated a book into account. The BA of a book is calculated by Equation (13) [8].

$$BA(d_i) = \frac{\hat{n} \cdot \hat{m} + \sum\limits_{r \in R_{d_i}} r}{n + \hat{n}} \tag{13}$$

where $R(d_i)$ is the set of ratings for d_i, $\hat{m}$ is the average unweighed rating and $\hat{n}$ is the average number of ratings over all the books in the top N list. With the Bayesian averaging, the influence value is computed with

$$inv(d_i) = \frac{1 + BA(d_i)}{1 + BA_{max}} \times pv(d_i) \tag{14}$$

The final recommendation values for these two methods are available through the Equation (6). In Equation (6), the parameter α is selected by cross-validation on the training set. At last, the top N candidates of the searching list is reranked by this recommendation value.

4.3 Combination with Learning-to-Rank

In the previous ranking strategies, usually several sub-strategies are investigated. Generally, these sub-strategies should be combined and merged. Howeve, in most conventional systems, the combination and the parameter adjustment is usually based on manual or semi-manual rules, which is time-consuming and error-pruning. Learning-To-Rank (L2R) [25], a supervised learning method, is to solve such problems. In L2R, the first step is about Data Labeling, i.e., preparing the training data. Then the features, sometimes the outputs from the initial ranking strategies, are extracted to represent each document. Afterwards, a learning model is chosen to learn the training data. Finally, we score the documents with the learned model. In this way, different strategies are easily and adaptively combined.

Following the step of the Reranking Model, we have 11 different reranking results. To make full use of these results, Learning-to-rank is used to combine them through some typical learning models [10]. In our system, Random-Forest, Rank-Net and Coordinate-Ascent learners are chosen as the training models. The recommendation score of each reranking method is regarded as a feature of the documents. Then the previous queries are used to label the training set. Afterwards, the learner parameters are optimized and selected with the K-fold cross-validation on the training set.

5. EXPERIMENTS

5.1 Experimental Data

In this section, we focus on the document collections from Amazon.com and the queries and "qrels" (professional recommendations) from LibraryThing.com, provided by INEX Social Book Search (SBS) Track organizers [14] [3]. The eval-

[3] `https://inex.mmci.uni-saarland.de/tracks/books`

uation measure is NDCG@10 and the tool we use is trec_eval 9.0 [4], the same with the INEX official evaluation.

The documents collection has a total of about 2.8 million book records. Each book record is represented in the format of a separate XML file with an ISBN number to be uniquely identified. According to the overview of the INEX 2011 Social Book Search track, the books in the collection contain large amount of professional metadata and wealthy user-generated data. We analyze all fields of the XML and the information is broadly divided into two categories: one is to describe the publication information of a book, such as <title> and <publisher>; the other is to describe the user-generated data, such as <tags> and <ratings>. A list of all element names in the collections is shown in Table 1.

Table 1: A list of all fields.

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

LibraryThing.com (LT) is a social book cataloguing website. LT is a platform where users can post a topic in the forum to seek for recommendations of books they desired. The descriptions of the requests are diverse because of the different habits and skills of the expression of topic creators. Some users give their suggestions of the posted topics. Since the year of 2011, INEX Social Book Search Track organizers have been investigating the LT forum requests and recommendations of books. They define the relevance values to distinguish between books that are mentioned in suggestions as positive recommendations, negative recommendations, neutral suggestions and books mentioned for some other reasons. Based on the behaviors of the topic creators on the suggestions, they build a decision tree [5] to help to label the relevance values of the suggestions manually. There are respectively 211, 96, 386, 680 posted topics collected in INEX 2011, 2012, 2013 and 2014 Social Book Search Task respectively. The sample parts of a topic XML are shown in XML 1.

As shown in XML 1, each query topic mainly consists of four types of fields: (1) **Title** The <title> field contains the general description headings. The title is given by topic creators, sometimes useless for suggestion. (2) **Narrative** The first message of recommendation request posts, posted by the topic creators, described in field <narrative>. The narrative is a detailed description of user requests of the book, with varying lengths and characters. It reflects the user's information needs in many ways, such as writing style, whether it is popular, etc, but may also draw some useless information. (3) **Group** The <group> field tells what kind of books needed. It's in the format of a brief description, and usually reflects the user's information needs. And (4) **Query** The <query> field is provided by an annotator to explain the exact topic in the format of a brief description, the length of more unified. But according to [21], this field is difficult to make a difference itself. The way we choose the training set and the test set is shown in Table 2.

Table 2: Training and testing sets in experiments.

Group	Training Set	Testing Set
1	2011-211 topics	2012- 96 topics
2	2011-211 topics 2012- 96 topics	2013-386 topics
3	2011-211 topics 2012- 96 topics 2013-386 topics	2014-680 topics

5.2 Experimental Setup

In our experiments, first we compare our system with several state-of-the-art models (the top systems won in the INEX Social Book Search Evaluation), e.g., Data Processing, Sequential Dependence Model , Bayes with Query Field Selection, and Social Feature Reranking Model based methods. Then we present experimental results with different components and stages (Initial Ranking, Reranking Model and Learning-to-rank) of our system. For each method, all parameters are set according to the INEX workshop papers. In our system, as mentioned in Section 4, all parameters α are selected by the cross-validation on the training set. Specifically, for the learning-to-rank [6], 5-fold cross-validation is performed to select the optimized parameters on each experimental dataset.

5.3 Experimental Results

5.3.1 Comparison on INEX 2012 and 2013 Social Book Search Task

In INEX 2012 Social Book Search (SBS) Track, it is the first time to define a relevant value for suggested books to distinguish the difference of recommendations. The training set is the official database of INEX 2011, which contains 211 topics, while the testing set contains 96 topics.

The performance of our model and the high-ranked runs in the year (2012) are presented in Table 3. In our experiments, the official evaluation tool Trec-eval and the official measure NDCG@10 are used for fair comparison.

In Table 3, runs of $p54$ focused on the indexing strategies and used the Social Feature Reranking Model, which is introduced in Section 2. Team $p4$ used a Bayesian rating

[4] `http://trec.nist.gov/trec_eval/`

[5] `https://inex.mmci.uni-saarland.de/tracks/books/INEX14_SBS_results.jsp\#mapping`

[6] We use the learning-to-rank tool, RankLib, which is available at `people.cs.umass.edu/˜vdang/ranklib.html`.

Table 3: Comparison results on SBS 2012.

Run	NDCG@10	P@10	MRR
Our run with Random Forest	0.2023	0.1526	0.4128
p54.run2 .all-topic-fields .all-doc-fields	0.1492	0.1198	0.3069
p54.run3 .all-topic-fields .QIT.alpha0.99	0.1488	0.1198	0.3066
p4.inex2012SBS .xml social.fb.10.50	0.1460	0.1380	0.370
p62.B_IT30_30	0.1339	0.1260	0.3410
p4.inex2012SBS .xml social	0.1297	0.1135	0.3256
p62.mrf-booklike	0.1295	0.1250	0.3584
p54.run5 .title.II.alpha0.94	0.1173	0.1073	0.2558
p62.IOT30	0.1141	0.1240	0.2933
p62.IT30	0.1082	0.1187	0.2999
p54.run6 .title.II.alpha0.97	0.0958	0.0823	0.2392

Table 4: Comparison on SBS 2013.

Run	NDCG@10	P@10	MRR
Our run with Random Forest	0.1659	0.0834	0.2475
RSLIS-run3 .all-plus-query .all-doc-fields	0.1361	0.0653	0.2286
UAms_ILLC .ti_qu_gr_na .bayes_avg.LT_rating	0.1331	0.0771	0.2342
UAms_ILLC .ti_qu .bayes_avg.LT_rating	0.1331	0.0771	0.2342
RSLIS-run1 .all-topic-fields .all-doc-fields	0.1295	0.0647	0.2190
UAms_ILLC .ti_qu_gr_na	0.1184	0.0555	0.2075
UAms_ILLC .ti_qu	0.1163	0.0647	0.2091
ISMD-run_ss _bsqstw_stop_words _member	0.1150	0.0479	0.1839
UAms_ILLC .qu.bayes_avg .LT_rating	0.1147	0.0661	0.1997
ISMD-run_ss _bsqstw_stop_words	0.1147	0.0468	0.1843
UAms_ILLC .ti.bayes_avg.LT_rating	0.1095	0.0634	0.2005

method based on Collaborative Filtering. Sequential Dependence Model was used by the team *p*62. Other submitted runs have a low evaluation value. As can be seen from Table 3, the run of our search-recommendation hybrid system performs much better than the others.

Moreover, we also compare our method with competitive systems on the official INEX 2013 Social Book Search Dataset. In this database, the field <query> is firstly used in query, which added a query processing problem to the task and the relevance value calculated through the decision tree built with a complex step. Several strategies are proposed to improve the performances of the models in 2012. The training set contains more than 300 topics (211 + 96, including some duplicated topics) and the testing set contains 386 topics. The performance results of our run and the high-ranked runs of the 2013 evaluation are shown in Table 4.

In table 4, *RSLIS* improved the previous Social Feature Reranking method by expanding the XML document. *UAms* tried to choose different fields of queries to improve the Bayesian Rating method. *ISMD* is not available in the workshop paper. The NDCG@10 value of all runs with SDM is lower than 0.10 [2]. From the table we can see although the evaluation value is lower, the runs of our search-recommendation hybrid system preform the best of all.

Overall, our proposed search-recommendation hybrid system for Social Book Search provides a unified framework for perform searching, reranking and merging, and has the best performance (NDCG@10) on both datasets compared to state-of-the-art systems.

5.3.2 Comparison of System Components

In this section, we present the performance comparison of initial ranking result, all reranking results and the learning-to-rank results on the INEX Social Book Search Task 2013.

We selected a total of 307 query topics provided by INEX Social Book Search 2011 and 2012 as the training set to train the proposed model. The best reranking parameters α of all methods are selected according to the best performances of training set. The reranking results are calculated based on the proposed model with the selected parameters. The initial and reranking results on training set and testing set are shown in Table 5 and Table 6.

Table 5: Reranking results on the training set.

Method	NDCG@10	α
Initial	0.1635	-
Tag	0.1724	0.93
Item	0.1701	0.94
Deep	0.1700	0.96
Node	0.1689	0.99
RatingBayes	0.1645	0.97
RatingReview	0.1712	0.98
Tag-Node	0.1699	0.97
Item-Tag	0.1697	0.96
Deep-Tag	0.1696	0.95
Item-Tag-Node	0.1698	0.98
Deep-Tag-Node	0.1694	0.95

Table 6: Reranking results on the testing set.

Method	NDCG@10	α
Initial	0.1383	-
Tag	0.1456	0.93
Item	0.1422	0.94
Deep	0.1425	0.96
Node	0.1407	0.99
RatingBayes	0.1404	0.97
RatingReview	0.1429	0.98
Tag-Node	0.1418	0.97
Item-Tag	0.1414	0.96
Deep-Tag	0.1415	0.95
Item-Tag-Node	0.1418	0.98
Deep-Tag-Node	0.1410	0.95

As can be seen from Table 5 and Table 6, all the reranking methods are well-performed except RatingBayes-Rerank method. The best performance is obtained by Tag-Rerank method with a greatest improvement of 5.2% compared with initial ranking. The average improvement of all reranking methods is 2.5%, which proves the effectiveness of the methods with recommendation techniques (GCF model). The unified Reranking Model with GCF can utilize various social information.

The reranking results of all methods on the training set are labeled with Qrels of 2011 and 2012, to be used as the training set of learning-to-rank. On the other hand, the reranking results on the testing set are generated into the testing set of learning-to-rank. Random Forest, Coordinate Ascent and Rank Net are used as training models. Several groups of parameters are used to train the model and we select the best parameters based on the stability performance of the validation set. The results of Random Forest, Coordinate Ascent and Rank Net learners are shown in Table 7, Table 8 and Table 9 respectively.

Table 7: Results with Random Forest.

Bag	S-rate	F-rate	tc	Tree	NDCG@10
200	1	0.1	11	5	0.1620
200	1	0.2	10	5	0.1597
200	1	0.2	11	4	0.1633
200	1	0.2	11	5	**0.1658**
200	1	0.2	12	5	0.1596
200	1	0.2	13	5	0.1575
200	1	0.3	11	5	0.1611
200	1	0.4	11	5	0.1586
300	1	0.2	11	5	0.1645
300	1	0.3	11	5	0.1605
300	1	0.2	12	5	0.1590
500	1	0.2	11	5	0.1620
500	1	0.3	11	5	0.1597
500	1	0.2	12	5	0.1575

Table 8: Results with Coordinate Ascent.

Tolerance	Restarts	Iterations	NDCG@10
0.0005	12	11	0.1553
0.001	12	11	0.1571
0.005	12	10	0.1526
0.005	12	11	0.1551
0.005	12	12	0.1563
0.01	11	11	0.1569
0.01	12	10	0.1596
0.01	12	11	**0.1614**
0.01	12	12	0.1587
0.01	12	13	0.1554
0.01	13	11	0.1565
0.05	12	10	0.1552
0.05	12	11	0.1569
0.05	12	12	0.1561
0.1	12	11	0.1575
0.5	12	11	0.1461

According to Table 7, Table 8 and Table 9, with appropriate parameters, the performances of these 3 learning models are better than all the reranking results. Random Forest has a highest improvement of 13% comparing with the best reranking result, while the improvement of Coordinate Ascent and Rank Net are 10.8% and 6% respectively. These experimental results verify that our merging strategy with learning-to-rank is effective.

Table 9: Results with Random Net.

Epochs	Layers	Nodes	NDCG@10
100	3	10	0.1472
200	3	10	0.1525
300	2	10	0.1480
300	2	16	0.1505
300	3	10	**0.1545**
300	3	11	0.1496
300	3	12	0.1353
400	2	10	0.1520
400	2	13	0.1516
400	3	10	0.1413

By comparison, we can see that T is the most effective reranking method mainly because the users always tag the books with relevant words without affective factors. Moreover, Random Forest gives out a best-performed model, which indicates that Random Forest is more stable and effective in this case.

5.3.3 Results on INEX 2014 Social Book Search

We participated in INEX 2014 Social Book Search Suggestion Task and submitted 6 runs based on the proposed system (with the prefix "USTB-"). According to the evaluation results of the updated version [7], the NDCG@10 results of 18 runs among 40 submitted runs are over 0.10. Our system won the INEX 2014 Social Book Search Evaluation. Moreover, all our 6 submitted runs are in the top 7. The evaluation results are shown in Table 10 [8].

Table 10: Participating results on SBS 2014.

Run	NDCG@10	MRR	MAP	R
USTB-run5 .L2R_RandomForest	0.142	0.258	0.102	0.390
HAFSI-326	0.142	0.275	0.107	0.426
USTB-run3 .L2R_Coordinate	0.138	0.256	0.101	0.390
USTB-run5 .L2R_RankNet	0.133	0.246	0.098	0.390
USTB-run2 .rerank_T	0.131	0.246	0.096	0.390
USTB-run1 .feedback	0.128	0.246	0.095	0.390
LSIS-InL2	0.128	0.236	0.101	0.441
AAU-run1	0.127	0.239	0.097	0.444
AAU-run3	0.120	0.227	0.090	0.425
CYUT-Type2QTGN	0.119	0.246	0.086	0.340
CYUT-0.95Average Type2QTGN	0.119	0.243	0.085	0.332
HAFSI-328	0.117	0.226	0.088	0.392
HAFSI-329	0.116	0.217	0.088	0.392
HAFSI-325	0.115	0.214	0.087	0.392
LSIS-InL2Feedback	0.114	0.230	0.094	0.434
HAFSI-324	0.112	0.214	0.086	0.392
LSIS-InL2tag Feedback	0.102	0.212	0.075	0.388

[7] `https://inex.mmci.uni-saarland.de/tracks/books/INEX14_SBS_resultsv2.jsp`

[8] In Table 10, we present results of 5 runs of our participated system. Another one, the **USTB-run6.simQuery**, is our run based on the topic similarity model for estimating similar topics between different-years evaluations [30].

As we can see from Table 10, unsurprisingly the runs based on our search-recommendation hybrid system have a better performance compared to our runs only based the searching system and most of the runs of other participating systems. Moreover, our run with the merging strategy by Random Forest is best-performed in the evaluation.

6. CONCLUSION

This paper presents a general search-recommendation hybrid system for semantically searching and navigating products (books) with a variety of social information. First, we propose a Generalized Content-Based Filtering (GCF) model. Then a reranking model based on the model is proposed to solve the semantic searching problem with social information. Afterwards, the model is applied in Social Book Search and uses learning-to-rank to combine the results of different reranking models. Moreover, our novel system is evaluated on INEX 2012 and 2013 Social Book Search Track and the runs perform much better than the other state-of-the-art systems. Finally, we participated in INEX 2014 Social Book Search Suggestion Task and won the first place.

In our paper, the proposed Generalized Content-Based Filtering model can be easily applied in most product recommendation systems, and the only difference is the way of feature extraction for various structures of the product descriptions. Consequently, in the near future, we will investigate this GCF model for applications in general recommendation systems. Another future issue is to expand the general Reranking Model and the general search-recommendation hybrid framework not only for Social Book Search but also for searching and navigating other web products with rich social information (e.g., on Taobao.com or Amazon.com).

7. ACKNOWLEDGMENTS

The research is partly supported by National Natural Science Foundation of China (61105018, 61175020).

8. REFERENCES

[1] M. Balabanović and Y. Shoham. Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[2] C. Benkoussasy and P. Belloty. Book recommendation based on social information. In *INEX'13 Workshop Pre-proceedings*, 2013.

[3] T. Bogers and B. Larsen. RSLIS at INEX 2012: Social Book Search Track. In *INEX'12 Workshop Pre-proceedings*, pages 97–108, 2012.

[4] T. Bogers and B. Larsen. RSLIS at INEX 2013: Social Book Search Track. In *INEX'13 Workshop Pre-proceedings*, 2013.

[5] T. Bogers and A. Van den Bosch. Collaborative and content-based filtering for item recommendation on social bookmarking websites. In *ACM'09 Workshop on Recommender Systems and the Social Web*, pages 9–16, 2009.

[6] L. Bonnefoy, R. Deveaud, and P. Bellot. Do social information help book search? In *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes*, 2012.

[7] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *ICML'07*, pages 129–136, 2007.

[8] E. Chang, K. Goh, G. Sychay, and G. Wu. CBSA: Content-based soft annotation for multimodal image retrieval using Bayes point machines. *IEEE Trans. Circuits and Systems for Video Technology*, 13(1):26–38, 2003.

[9] C. Djeraba. Association and content-based retrieval. *IEEE Trans. Knowledge and Data Engineering*, 15(1):118–135, 2003.

[10] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl. Combining collaborative filtering with personal agents for better recommendations. In *AAAI'99*, pages 439–446, 1999.

[11] I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel. Social media recommendation based on people and tags. In *SIGIR'10*, pages 194–201, 2010.

[12] M. K. Hugo Huurdeman, Jaap Kamps and J. van Wees. Using collaborative filtering in social book search. In *INEX'12 Workshop Pre-proceedings*, pages 109–113, 2012.

[13] G. Kazai and N. Milic-Frayling. Trust, authority and popularity in social information retrieval. In *CIKM'08*, pages 1503–1504, 2008.

[14] M. Koolen, J. Kamps, and G. Kazai. Social book search: Comparing topical relevance judgements and book suggestions for evaluation. In *CIKM'12*, pages 185–194, 2012.

[15] M. Koolen, G. Kazai, J. Kamps, A. Doucet, and M. Landoni. Overview of the INEX 2011 Social Book Search Track. In *Focused Retrieval of Content and Structure, Lecture Notes in Computer Science*, pages 1–29. 2012.

[16] M. Koolen, G. Kazai, J. Kamps, M. Preminger, A. Doucet, and M. Landoni. Overview of the INEX 2012 Social Book Search Track. In *INEX'12 Workshop Pre-proceedings*, 2012.

[17] M. Koolen, G. Kazai, M. Preminger, and A. Doucet. Overview of the INEX 2013 Social Book Search Track. In *INEX'13 Workshop Pre-proceedings*, 2013.

[18] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2(1):1–19, 2006.

[19] P. Lops, M. de Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pages 73–105. 2011.

[20] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: Social recommendation using probabilistic matrix factorization. In *CIKM'08*, pages 931–940, 2008.

[21] H. H. Marijn Koolen and J. Kamps. Comparing topic representations for social book search. In *INEX'13 Workshop Pre-proceedings*, 2013.

[22] P. Melville and V. Sindhwani. Recommender systems. In *Encyclopedia of Machine Learning*, pages 829–838. Springer, 2010.

[23] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *SIGIR'05*, pages 472–479, 2005.

[24] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *JCDL'00*, pages 195–204, 2000.

[25] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *KDD'09*, pages 727–736, 2009.

[26] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW'01*, pages 285–295, 2001.

[27] J. Tang, X. Hu, and H. Liu. Social recommendation: A review. *Social Network Analysis and Mining*, 3(4):1113–1133, 2013.

[28] F. Wang, P. Cui, G. Sun, T.-S. Chua, and S. Yang. Guest editorial: Special issue on information retrieval for social media. *Information Retrieval*, 15:179–182, 2012.

[29] M. Zanker, A. Felfernig, and G. Friedrich. *Recommender Systems: An Introduction*. Cambridge University Press, 2011.

[30] B.-W. Zhang, X.-C. Yin, X.-P. Cui, J. Qu, B. Geng, F. Zhou, and H.-W. Hao. USTB at INEX 2014: Social Book Search. In *INEX'14 Workshop Pre-proceedings*, 2014.