

# Assessing the Significance of Incorporating User Profiles in Social Book Search

Bothhokang Apadile  
University of Botswana  
P/Bag UB 00704  
Gaborone, Botswana

Edwin Thuma  
University of Botswana  
P/Bag UB 00704  
Gaborone, Botswana

Gontlafetse Mosweunyane  
University of Botswana  
P/Bag UB 00704  
Gaborone, Botswana

## ABSTRACT

In this article, it is hypothesized that personalizing the book search application by incorporating user profiles such as background of personal tastes, interests and previously seen books. can issue or produce a more effective query result set as well as an effective book recommendation. To meet this end, experiments were carried out to explore which topic representation gives the best result. Four different query representations, which are title, request, group and a combination of title-request-group were used. It was observed that the title-request-group query representation was best. In addition, an investigation was conducted to determine whether a learning to rank framework that incorporates topical relevance by exploiting user profiles for document re-ranking according to individual preference will issue a more effective result set. Moreover, an investigation was conducted to determine whether the use of keywords from profiles for query expansion and reformulation improves the search results. The results of these investigations suggest that a more effective query result set as well as an effective book recommendation can be attained by incorporating user profiles such as background of personal tastes, interests and previously seen books into the social book search application.

## General Terms

Query Expansion, Learning to Rank

## Keywords

Social Book Search, User Profiles

## 1. INTRODUCTION

We are living in the era of information age where access and control of information is the defining key in human civilization. Documents are now digitized, hence more and more users are now looking to the web to find their next read. With the development of the Internet and storage devices, on-line document servers are abound with enormous quantities of documents, so that finding the right and useful information becomes a very difficult task [1]. Shardanand & Maes [2] corroborate by stating that recent years have seen the explosive growth of the sheer volume of information. The number of documents, movies, news, advertisements, and in particular on-line information, is staggering. The volume of things is considerably

more than any person can possibly filter through in order to find the ones that he or she will like.

Due to the high volume of information on-line, users find that they are exposed to a lot of information and are unable to make quality decisions due to their fairly limited cognitive processing capacity. This is what is referred to as an 'information overload' - the input exceeds the human mind's processing power. In many cases when a user submits a query to a search engine he/she must wade through hundreds of results most of them irrelevant. Users handle this information overload through their own effort and the effort of others [2]. Readers are now interacting via the web to get or give recommendations on books. They do this by giving reviews on the books they have previously read. Information regarding which books they have searched is recorded [3]. With every comment that the readers write and every rating that they give they leave a footprint. Just like the documents on-line, the trail that users leave is increasing with every click and every review they leave on-line. This information is rarely ever used in information retrieval (IR). As it stands the Internet has a one size fits all to IR, this presents the user with an information overload. The core of the problem is that whether a user is a primary school student or a masters graduate, if users submit the same query, the query results will be identical discarding the obvious gap levels between the two users. As Arezki et al. [1] points out "Two users can formulate the same request for different needs- for example, the results awaited by an expert in Java language formulating the request "Java course" are different from the results awaited by a non-expert with the same request". Therefore in order to increase the relevance of the results, the use of user profile information should be taken into consideration in the IR process. Relevance simply denotes how well a retrieved set of documents meets the information needs of the user. There are two main types of relevance which are prominent in the IR field; topical relevance and user relevance. Topical relevance refers to relevance to a subject (topic), in its most simple form, matching words in documents and queries, whereas user relevance is a user-oriented view of relevance which is based on the user's evaluation of the usefulness of the documents.

It is important that users find documents that meet their expectations without searching through a ton of documents before satisfying their information need. Queries in general are very short and produce an incomplete specification of the users' information need. The burden of how to phrase a query in order to find documents that are relevant should be taken off the user. Hence, in this work, a Social Book Search (SBS) system is developed in which not only the

topical relevance is used but also information from the user profile is also taken into consideration in order to return results that meet the user's information need. The intuition is that this would greatly benefit the user as their information need is weighed against their background of personal tastes, interests and previously seen books. Koolen et al. [4] envisions that book search will grow to more than using traditional metadata. It will be more personalized as books are no longer described using professional catalogues but are also described using the users own vocabulary. The user's search context or background may provide information that disambiguates their information need in order to automatically predict and issue a more effective query [5].

## 2. RELATED WORK

### 2.1 User Profiling

Adomavicius & Tuzhilin [6] defines a user profile as a collection of information that describes a user and can be specified with sets of rules learned from user transactional histories using various data mining techniques. According to Gauch et al. [7], user profiles may include demographic information, e.g., name, age, country, education level, etc., and may also represent the interests or preferences of either a group of users or a single person. These profiles are then used to find or recognize other documents that are likely to be of interest [8]. However, since users interests are ever changing and ever growing, user profiling is not an easy task. Gauch et al. [7] corroborates this by differentiating between two different kinds of profiles namely dynamic - those that can be modified or augmented in contrast to static profiles that maintain the same information over time. They also mention that dynamic profiles take time into consideration and may differentiate between short-term and long-term interests. Short-term profiles represent the users current interests whereas long-term profiles indicate interests that are not subject to frequent changes over time.

In addition Middleton et al. [9] identifies user profiling as either knowledge-based or behaviour-based. Knowledge-based user profiling centres on a user being asked about their interests in the current moment, which allows learning about users characteristics. This approach is static, intrusive and time consuming. Questionnaires and interviews are often employed to obtain this user knowledge. Behaviour-based user profiling on the other hand uses the users behaviour to build and improve the profile dynamically, commonly using machine-learning techniques to discover useful patterns in the behaviour. Therefore it is much desirable to focus on behaviour-based profiling since the user would normally want a quick response from the system without having to fill out any questionnaire forms. Even though the behaviour-based user profiling is desirable for the user, the knowledge-based user profiling is much easier to create. Table 1 presents a distinction between the traits of a user profile that has been built through knowledge-based approach and the behaviour-based approach.

## 2.2 TRENDS IN THE SOCIAL BOOK SEARCH

### 2.2.1 The 2011 Social Book Search Task.

The Social Book Search task began in 2011 with the primary goal of investigating the complex nature of relevance in book search and the relative value of traditional and user-generated book metadata in retrieval. Koolen et al. [10] state that in the 2011 Social Book Search Task, the best run for the 211 topics and associated relevance judgments was by the University of Amsterdam. Their results showed that forum suggestions are not drawn from a much larger set of equally relevant books, but from a more or less com-

Table 1. DIFFERENCE BETWEEN KNOWLEDGE-BASED AND BEHAVIOUR-BASED USER PROFILE

KNOWLEDGE-BASED PROFILE	BEHAVIOUR-BASED PROFILE
Asking a user to rate an item	Observing the items that a user views in a particular site
Asking a user to rank a collection of items from favourite to least favourite	Analysing item / user viewing time
Presenting two items to a user and asking him/her to choose the better one of them	Keeping a record of the items that a user purchases online
Asking a user to create a list of items that he / she likes	Analysing the users social network and discovering similar likes and dislikes

plete set of the best or most popular books for the requested topic. University of Amsterdam used pseudo relevance feedback on an index with only reviews and tags in addition to the basic title information. Their system proved to be good at finding topically relevant books; this influenced the decision to look more carefully at different aspects of relevance, such as topical relevance, recommendation, reading level and whether a book looks interesting or engaging.

### 2.2.2 The 2012 Social Book Search Task.

The 2012 task changed focus from the relative value of professional and user-generated metadata to the complexity of book search information needs. The goal was to evaluate approaches used to support users in reading, searching and navigating metadata, full texts and user-generated content. The best performing run was by the Royal School of Library and Information Science; it used all topic fields combined against an index containing all available document fields [3]. The official measure was nDCG@10, which considers graded relevance and concentrates on top retrieved results. The best performing run did not use any information from the user profiles. However the second best performing run which was also by the Royal School of Library and Information Science did incorporate user profiles. It did so by using all topic fields against all document fields but then re-ranked the results list based on the profile of the topic creator [3]. This means that the retrieved books that shared a lot of tags associated with books already present in the users catalogue are regarded as a more appropriate match. In summary, systems that incorporate user profile information have not so far been able to improve upon a plain text retrieval baseline.

### 2.2.3 The 2013 Social Book Search Task.

The goal of the 2013 SBS task remained the same as the one for the previous year; to investigate search and navigation using professional metadata and user-generated content for retrieval and recommendation of books on the Web. What was different from previous years was that there was a much deeper understanding into the nature of the book search information needs and book suggestions from the forums. As stated by Koolen et al. [11] the results showed that the most effective systems incorporated the full topic statement, which includes the title of the topic thread, a query provided by the annotator, the name of the discussion forum and the full first message that elaborates on the request. From the above mentioned items, it is clear that the system employed full text retrieval that ignored all user profile information. The best performing system by the Royal School of Library and Information Science used all topic

fields combined against an index containing all available document fields.

#### *2.2.4 The 2014 Social Book Search Task.*

The 2014 SBS task continued with the goal from the prior years and yet again none of the best performing groups used user profile information for the runs submitted. As stated by Koolen et al. [12], the best performing system used all topic fields combined against an index containing all available document fields. The run is then re-ranked with 12 different re-ranking strategies, which are combined adaptively using Learning to Rank. 2014 was the first year in which systems included Learning to Rank. Zhang et al. [13] who had the best performing system used the probability of the query content produced by language models to rank the documents. They noted that in order to improve the initial ranking, the results had to be re-ranked using 11 different strategies after analyzing the structure of XML: 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). In order to choose the most effective feature and select the optimized parameter  $\alpha$ , in the first round, they trained their re-ranking model on Social Book Search 2011-2012 and tested on Social Book Search 2013. They outlined that all the features showed improvements of different degree. So they used all features to combine the results. Random Forest was used to re-rank the documents. The runs submitted were evaluated using graded relevance judgments. The relevance values were labelled manually according to the behaviours of topic creators. The authors noted that on both the training and the testing set the best results were from combining all re-ranking results in Random Forest. This suggests that good use of social information can improve the results of Social Book Search. The high evaluation value of Similar Query method indicates the amount of similar topics not the effectiveness of the model. The second best performing system by the group UJM, used BM25 on the title, mediated query and narrative fields, with the parameters optimised for the narrative field.

#### *2.2.5 The 2015 Social Book Search Task.*

The 2015 SBS task which also had the same goals as the tracks from the previous year still had the best performing systems making no use of profile information [4]. Thus the additional value of user profiles still had not been established. However it was observed that several of the best performing systems made use of Learning to Rank approaches. The best performing system namely the MIIB - Run6 described that they generated queries from all topic fields and applied a BM25 index with all textual document fields merged into a single field. They also outlined that they used Learning to rank, a machine learning technique, utilising random forest on 6 result lists as well as the price, the book length and the ratings. Results were then re-ranked based on tags and ratings. The system which was ranked second to MIIB- CERIST [14] outlined that they tackled the problem through two contributions. They started off by introducing the tf-idf function to assign weights to value terms which are significant to the topic and a low weight to those appearing in many different topics. They also outlined that they optimized the parameters of the BM25 term weighting model using the 2014 topics. Secondly, and to better represent the topic, they added other terms by expanding the original query using Rocchio technique. One of their main findings was that using the tf-idf function to weigh all the topic terms improves the result more than using the frequency of the terms.

This gives insight that in the book search domain systems need to learn from user behaviour what the right balance is for the multiple and diverse sources of information, both from the collection and the user side.

### **2.3 Summary**

From the trends of the Social Book Search Track, it is evident that there is no personalisation in the retrieval systems as there has not been any system that has successfully used the user profiles. To date the importance of whether learning about the implicit and explicit preferences of individual users can improve the relevance of a query is yet to be proven. Therefore the aforementioned overall aim of using forums to shed light on complex search requests, their context and relevance aspects and relating it to a wealth of studies in information seeking still stands.

## **3. EVALUATION DATASET AND THE CRANFIELD EVALUATION METHODOLOGY**

### **3.1 THE CRANFIELD EVALUATION METHODOLOGY**

In this work, the less expensive system evaluation methodology referred to as the Cranfield paradigm is used [15, 16]. Experiments conducted in this way require a resource known as a test collection and an evaluation measure [16]. Test collections are re-usable and standardised resources that can be used to measure the retrieval effectiveness of an information retrieval system [16]. The main components of an information retrieval test collection are the document collection, topics and the relevance judgments. The following is a description of each component:

- Document Collection: A static set of documents to be searched.
- Topics (Queries): A set of statements that describe typical users' information needs. These might be expressed as queries that are submitted to an IR system, questions or longer written descriptions.
- Relevance Judgments: For each topic in the test collection, a set of relevance judgments must be created indicating which documents in the collection are relevant to each topic. The notion of relevance used in the Cranfield approach is commonly interpreted as topical relevance: whether a document contains information on the same topic as the query. In addition, relevance is assumed to be consistent across assessors and static across judgments. The author is the person best qualified to judge relevance.

With an appropriate test collection, and a chosen evaluation measure, an IR researcher can assess and compare the effectiveness of different retrieval strategies when deployed in an IR system [17].

### **3.2 DOCUMENT COLLECTION**

The document collection used in this study consists of 2.8 million book records from Amazon, extended with social metadata from LibraryThing (LT) [3, 4, 10, 11]. The records contain title information as well as a Dewey Decimal Classification Code (DDC) and category and subject information supplied by Amazon. The books are identified by an International Standard Book Number (ISBN). It should however be noted that different editions of the same work have different ISBN's. Each book record is represented as an XML file with fields like isbn, title, author, publisher, dimensions, numberofpages and publicationdate which are extracted from Amazon as a form of formal metadata. As for the social or user generated

metadata, it consists of fields such as tag, rating and review from both Amazon and LT. The full list of fields that have been identified as relevant to this work is shown in Table 2. From LT, there are user tags and user-provided metadata on awards, book characters and locations and blurbs. There are additional records from the British Library and the Library of Congress; this ensures that there is enough high quality metadata. Both the British Library and the Library of Congress records are in MARCXML format (an XML version of the well-known MARC format). Even though there is no single library catalogue that covers all books available on Amazon, it has been found that these combined library catalogues can improve both the quality and quantity of professional metadata. With only the Amazon and LT data, 57% of the book descriptions have at least one subject heading, but with the addition of the Library of Congress and British Library, it increases to 80%.

Table 2. Document Collection Field Explanation

FIELD	EXPLANATION
Title	This refers to the name/title of the book
Reviews	This refers to the lump sum critical appraisals of the book by readers
Review	This refers to the critical appraisal of the book by one reader
Authorid	This refers to the id that uniquely identifies the reader writing the critical appraisal of the book
Date	This refers to the date in which the reviewer made the review
Summary	This refers to a brief statement of what the reviewer thought of the book
Content	This refers to detailed statement of what the reviewer thought of the book
Rating	This refers to an evaluation given on a 5 point Likert scale
TotalVotes	This refers to the total number of people who viewed the review
HelpfulVotes	This refers to the number of people who found the review helpful
EditorialReviews	This refers to the lump sum reviews by the editors of the book
EditorialReview	This refers to the review by one editor
Content	This refers to a detailed statement of the editors review of the book

### 3.3 Topics

The topic set consists of 120 topics, where each topic has a narrative description of the information need as well as one or more example books provided by the topic creator. These topics are taken from the LT forum. Many of the topic threads are initiated with a request from a member for interesting, fun new books to read. In these topic threads users make mention of which types of books they are looking for, giving examples of what they like and dislike, indicating which books they are familiar with and ask for recommendations. Each topic has a title and is associated with a group on the discussion forums. It as well has at least one example book provided by the requester that helps other forum members understand the direction in which the requester is thinking. For each example, there is a book ID from LT (which can be mapped to the Amazon/LT collection using a mapping file), and information on whether the topic creator has read the book and whether s/he is positive, negative or neutral about that example.

### 3.4 Relevance Judgments

The books mentioned by users replying in the topics are used as relevance judgments. It is assumed that those suggestions that the information seeker later adds to his personal catalogue are most relevant, and thus are assigned a higher relevance value than judgments that are just topically relevant or already in the library of the topic starter. Non-relevant suggestions are those works the information seeker already mentions and are assigned a relevance score of zero. Forum members can mention books for many different reasons. The relevance values distinguish between books that were mentioned as positive recommendations, negative recommendations (books to avoid), neutral suggestions (mentioned as possibly relevant but not necessarily recommended) and books mentioned for other reasons (not relevant at all). There is also a differentiation between recommendation from members who have read the book they recommend and those who have not. A recommendation from someone who has actually read the book is regarded as more valuable. The mapping of relevance values is defined using the has read tag, if a forum member has read the book it is reflected as a yes, else if the member has not read the book it is denoted as a no. This is done bearing the following assumptions in mind:

- When it is not clear whether the person mentioning the book has read it or not, it is taken as a has not read.
- If it is not clear whether the suggestion was positive, negative or neutral it is treated as neutral.
- A work with only negative suggestions has no value for the requester when found in the search results.
- Has read recommendations overrule not read recommendation, this is because someone who has read the book is in a better position to judge a book than someone who has not.

### 3.5 User Profiles

This set consists of almost over 94,000 user profiles anonymised from LT with 34 million cataloguing transactions, and can be used to derive recommendations based on collaborative filtering. The user profiles file contains five columns separated by tabs: `<user_id><TAB><book_id><TAB><add_date><TAB><user_rating><TAB><user_tags>`

- The user\_id field is an anonymised id instead of the LT username.
- The book\_id field contains book IDs from LT, which can be mapped to their ISBNs via the ISBN-LT\_id mapping file that's also available on the website. These IDs are similar to the ones in the topic files.
- The add\_date is the month in which the user added that book to their own catalogue
- The ratings and reviews are those added by the user at some point.

## 4. BASELINE SOCIAL BOOK SEARCH RETRIEVAL SYSTEM

To meet the objectives of this research, a baseline of the Social Book Search Retrieval System was first built. The baseline system served as a basis for defining change, where the change reflects a movement from the baseline state to the next. The basis of the baseline system for the Social Book Search Retrieval System was topical relevance where the system retrieves results that are relevant to the subject topic. Topical relevance works in such a way that it matches the words in the query to the words in the document, i.e. it

is the relation between the subject of the document and the subject of the query. From thereon user profile data was incorporated in this baseline system and an evaluation was conducted to assess the retrieval effectiveness of the SBS system after incorporation user profiles.

#### 4.1 MAIN INTEGRANTS OF THE SOCIAL BOOK SEARCH RETRIEVAL BASELINE SYSTEM

This section describes the main integrant of the Social Book Search Retrieval Baseline System. First, a description of the document collection and how it was stored and prepared for retrieval is provided. This is then followed by how the topics/ set of queries were prepared for matching and retrieving relevant documents from the collection. This is followed by an ISBN to WorkId conversion as it has been specified that work refers to the intellectual work of an author. Therefore, each record in the collection corresponds to an ISBN and each ISBN corresponds to an intellectual work. That is when the evaluation will then take place.

##### 4.1.1 Data Cleaning.

Data cleaning refers to removing noise or outliers, collecting relevant information for modelling noise, deciding on strategies on missing data, and accounting for time sequence information and known changes [18]. Saravanan et al. [18] further outlines that clean data implies relevant data. Hence as not all data provided as part of the document collection was particularly useful in this research, it was therefore vital that the document collection be cleaned before the index process could take place. Table 3 presents a list of tags that were deemed as not useful and therefore removed from the documents. Only the tags in Table 2 were left in each document to be indexed.

Table 3. DOCUMENT TAGS REMOVED

---

ean, binding, label, listprice, manufacturer, publisher, readinglevel, releasedate, publicationdate, studio, edition, dewey, numberofpages, dimensions, images, creators, blurbers, dedications, epigraphs, firstwords, lastwords, quotations, series, awards, characters, places, subjects, tags, similarproducts, browseNodes
---

---

##### 4.1.2 Query Processing.

120 queries were provided from the LT forum. The queries contained the following:

- The name of the group where the request was initially posted (group)
- The title of the entry (title)
- Actual requests in the form on natural language (request)
- Potential book examples (example)
- Catalogue of the already read books of the user (catalogue)

Four combinations of these tags were considered as input for the retrieval process, namely: title, request, group, title-request-group. For a more efficient way to process the queries, they were written to a separate file in the following format:

```
Topic_id /tab topic_title  
/tab topic_request /tab topic_group
```

##### 4.1.3 INDEXING.

The platform chosen for indexing was Terrier-4.2<sup>1</sup> [19]. Terrier is

<sup>1</sup><http://terrier.org>

a highly flexible and transparent open source IR platform for research experimentation in text retrieval which is written in Java. It implements the state of the art indexing and retrieval functionalities and provides an ideal platform for the rapid development and evaluation of large-scale retrieval applications. The basic indexing setup for the document collection after cleaning the data was as follows:

- The marker tag for document boundaries was set as BOOK
- The tag that contains the DOCNO of the document was set as ISBN
- The tags that should be parsed in the collection were set as TITLE, REVIEWS, CONTENT
- The tags that should not be parsed in the collection were set as RATING, TOTALVOTES, HELPFULVOTES

##### 4.1.4 TREC EVALUATION.

After the indexing phase was done, a rank-ordered list of 1000 documents for each query was retrieved. Since the books in the document collection were identified using the ISBN number, it was necessary to convert the ISBN number that appears in the result set to its respective work Id. It is therefore important to note that one book can have many editions thus rendering those different editions with different ISBN numbers. In the case of the work Id, different editions of the same book have the same work Id thus an ISBN to a work Id mapping is a one to many mapping. To convert the ISBN numbers to their corresponding work Ids a java code was written, from thereon it removed any duplicates that occurred in respect to the combination of the query id and work Id. Once the conversion was done, trec\_eval<sup>2</sup> tool was deployed taking the QRels and the retrieved results as input to obtain the final set of evaluation results.

#### 4.2 Empirical Evaluation: Choosing a Suitable Baseline Term Weighting Model

This section presents a set of experiments which were conducted to determine a suitable baseline term weighting model for the Social Book Search Retrieval System. The term weighting models chosen for this experiment are:

- BM25: The BM25 probabilistic model
- DPH(DFR): A different hyper-geometric DFR model using Popers normalization
- Hiemstra LM: Hiemstra language model
- PL2 (DFR): Poisson estimation for randomness, Laplace succession for first normalisation and Normalisation 2 for term frequency normalisation.

From the above mentioned term weighting models, the model that returns the highest evaluation score was carried forward to the next experiments. The topics and each of the above-mentioned term weighting models were used to rank and retrieve the documents that were indexed as specified in Section 4.1.3. From there on the results were evaluated using the MRR and NDCG@10 evaluation measures for each of the denoted term weighting models. These evaluation measures assess how well the search result satisfied the users query intent. Looking at the differences between the evaluation measures as well as the nature of the SBS IR System, the chosen official measure was the NDCG@10 as it takes graded relevance values into account and is designed for evaluation based on the top retrieved results. The weighting model that produces the highest score for the evaluation measures was then chosen as the

<sup>2</sup>[http://trec.nist.gov/trec\\_eval/](http://trec.nist.gov/trec_eval/)

baseline model. This baseline model formed the basis on which the system was to be improved to meet the objectives of the study. The first step in the experiments was to determine which weighting model yields the highest score in relation to the different query formats. For better retrieval the queries were processed into the format below:

- Topic-Title: Only the title of each topic
- Topic-Request: Only the request field
- Topic-Group: Only the group field
- Topic-All-Fields: Contains title, request, group

The results of the experiments for topical relevance are highlighted in the Table 4:

Table 4. PERFORMANCE OF QUERIES WITH VARIOUS TERM WEIGHTING MODELS

Weighting Model	Evaluation Measure	Topic-All-Fields	Topic-Title	Topic-Request	Topic-Group
BM25	MRR	0.190	0.0578	0.1902	0.0046
	NDCG@10	0.0833	0.0266	0.0782	0.0009
DPH	MRR	<b>0.2520</b>	0.0867	0.1966	0.0262
	NDCG@10	<b>0.1075</b>	0.0442	0.0843	0.0030
Hiemstra-LM	MRR	0.1621	0.0001	0.1537	0.0015
	NDCG@10	0.0686	0.0091	0.0571	0.0008
PL2	MRR	0.1342	0.0419	0.1274	0.0046
	NDCG@10	0.0595	0.0239	0.0470	0.0010

With respect to which query format is most effective, Table 4 shows that the combination of the title, request, group which is the Topic-All-Fields is most effective as it depicts the highest score of 0.1075 (NDCG@10) for DPH term weighting model as compared to when the title, request and group are used singularly with the highest being request at 0.0843 (textbfNDCG@10) for DPH. Therefore, the chosen query format is the Topic-All-Fields. As can be seen in Table 4, the DPH term weighting model is the best as compared to the other term weighting models as it yields the highest scores for MRR and NDCG@10 at 0.2520 and 0.1075 respectively. Therefore DPH is the chosen baseline term weighting model.

## 5. INCORPORATING USER PROFILES IN SOCIAL BOOK SEARCH

In this section, the following research questions were investigated:

- Research-Question One (RQ1)** : Will a learning to rank framework that incorporates topical relevance by exploiting user profiles for document re-ranking according to individual preference issue a more effective result set?
- Research-Question Two (RQ2)** : Does the use of keywords from profiles for query expansion and reformulation improve the search results?

### 5.1 LEARNING TO RANK EXPERIMENTATION

Learning to rank techniques refers to algorithms that use machine learning techniques and an appropriate combination of features to learn an effective ranking model [20]. These features can be classified as query dependent features and query independent features. Query dependent features are those features which depend both on

the contents of the document and the query [21]. Query Independent features are those features which depend only on the document, but not on the query [21].

The steps for learning to rank are as follows [21, 22]:

- (1) Top K Retrieval: For a set of training queries, generate a sample of documents using an initial retrieval approach.
- (2) Feature Extraction: For each document in the sample, extract a vector of feature values. A feature is a binary or numerical indicator representing the quality of a document, or its relation to the query.
- (3) Learning: Learn a model by applying a learning to rank technique. Each technique deploys a different loss function to estimate the goodness of various combinations of features. Once a learned model has been obtained from the above learning steps, it can be deployed within a search engine as follows:
- (4) Top K Retrieval: For an unseen test query, a sample of documents is generated in the same manner as in step (1),
- (5) Feature Extraction: As in step (2), a vector of feature values is extracted for each document in the sample. The set of features should be exactly the same as for step (2).
- (6) Learned Model Application: The final ranking of documents for the query is obtained by applying the learned model on every document in the sample, and sorting by descending predicted score.

#### 5.1.1 TRAINING, TESTING AND VALIDATION QUERY SETS.

The 120 queries outlined in Section 4.1.2 were divided into three parts, i.e. 1/3 training set, 1/3 testing set and 1/3 validation set. This sums up to 40 queries each for the training, testing and validation.

#### 5.1.2 EXTRACTING FEATURES FOR LEARNING TO RANK.

In this section, a description of the features extracted for the training, testing and validation sets is provided. These training, testing and validation sets were subsequently used in the learning to rank experiments. The DPH term weighting model was used to retrieve a sample of the SBS documents for each query in the training, testing and validation set. Due to the small nature of the collection, all the SBS documents that matched the query were retrieved. Several query dependent and query independent features were extracted from this sample of documents. Ten term weighting models were used as query dependent features and eleven query independent features taken from the document collection. Such features can be precomputed in off-line mode during indexing. They may be used to compute the documents static quality score (or static rank), which is often used to speed up search query evaluation [21]. Query independent features were extracted from the user profiles in the document collection. Table 5 provides a list of features used in the learning to rank experimentation.

#### 5.1.3 EXPERIMENTAL SETUP.

**FAQ Retrieval Platform:** Terrier-4.2 [19], an open source IR platform was used to extract all the query dependent features. All the documents used in this study were first pre-processed before indexing and this involved tokenising the text and stemming each token using the full Porter stemming algorithm.

**Training Learning to Rank Techniques:** For the learning to rank approach, RankLib<sup>3</sup>, a library of learning to rank algorithms was used. In this work, two state-of-the-art listwise approaches were deployed. Prior work has indicated that listwise approaches are often effective compared to the other approaches [20]. In particular,

<sup>3</sup><http://people.cs.umass.edu/vdang/ranklib.html>

Table 5. All query-dependent (QD) features used in this work.

Features	Type	Total
DPH hypergeometric weighting model	QD	1
BM25 weighting model	QD	1
DirichletLM hypergeometric weighting model	QD	1
Hiemstra_LM - Hiemstra LM weighting model	QD	1
DLH weighting model	QD	1
PL2 weighting model	QD	1
TF_IDF weighting model	QD	1
DFree weighting model	QD	1
DFRDependenceScoreModifier	QD	1
MRFDependenceScoreModifier	QD	1
Number of Reviews	QI	1
Average Number of Ratings	QI	1
Minimum Number of Ratings	QI	1
Maximum Number of Ratings	QI	1
Variance of the Ratings	QI	1
Standard Deviation of the Ratings	QI	1
Average Number of Helpful Votes	QI	1
Minimum Number of Helpful Votes	QI	1
Maximum Number of Helpful Votes	QI	1
Variance of the Helpful Votes	QI	1
Standard Deviation of the Helpful Votes	QI	1
Total		21

Coordinate Ascent [23], which is a linear-based learner and LambdaMART [24], which is a tree-based learner were deployed. To train and test LambdaMART and Coordinate Ascent, the default RankLib parameter values of the algorithms were used. In all experiments, NDCG was used as the objective function [21].

## 5.2 EXPERIMENTAL RESULTS AND EVALUATION

### 5.2.1 INCORPORATING QUERY DEPENDENT FEATURES IN A LEARNING TO RANK APPROACH.

In the first experiment, the focus was on the role of query dependent and query independent features for attaining an effective learned model. This section investigates the role played by the multiple document weighting models and user profiles outlined in Table 5 and how they bring different evidence within the learning to rank process. Table 6 presents the results of using these query dependent (QD) and query independent (QI) features.

Table 6. Query Dependent and Query Independent Features Results.

	Coordinate Ascent	LambdaMART
<b>MRR</b>	0.2857	0.2283
<b>NDCG@10</b>	<b>0.1338</b>	0.1075

For the first research question *RQ1*, an investigation on whether a learning to rank framework that incorporates topical relevance by exploiting user profiles for document re-ranking according to individual preference can issue a more effective result set was conducted. The results of this investigation suggests that indeed a learning to rank framework that incorporates topical relevance by exploiting user profiles for document re-ranking according to individual preference does produce a more effective result set when Coordinate Ascent is deployed as depicted by an increase

in NDCG@10 from 0.1075 in Table 4 to 0.1338 in Table 6. However, there was no improvement in the retrieval performance when LambdaMART in deployed.

### 5.2.2 INCORPORATING KEYWORDS FROM USER PROFILES FOR QUERY EXPANSION.

Query expansion (QE) is the process of reformulating a query to improve retrieval performance in information retrieval operations, particularly in the context of query understanding. In this experiment, an investigation on whether expanding the original query with keywords from profiles can improve the retrieval performance was conducted. Table 7 presents the results of using an expanded query in.

Table 7. Query Expansion Results.

<b>MRR</b>	0.2769
<b>NDCG@10</b>	<b>0.1400</b>

For the second research question *RQ2*, an investigation on whether the use of keywords from profiles for query expansion and reformulation can improve the search results is conducted. The results of this investigation suggests that expanding the original query with keywords from user profiles does indeed improve the retrieval results as shown by an increase in NDCG@10 from 0.1075 in Table 4 to 0.1400 in Table 7.

## 5.3 CONCLUSIONS

The goal of this research was to develop an IR system to suggest books based on rich search requests combining several topical and contextual relevance signals, as well as user profiles. The article hypothesized that personalizing the book search by incorporating user profiles such as a background of tastes, interests, previously seen books, etc. can issue or produce a more effective query result set as well as an effective book recommendation. To meet this end, a baseline SBS IR system was developed and an investigation conducted to determine a query format that can give the best retrieval performance. With respect to which query format is most effective, the results of this investigation suggest that the combination of the title, request, group which is the Topic-All-Fields is most effective (Table 4). In addition, an investigation was conducted to determine whether a learning to rank framework that incorporates topical relevance by exploiting user profiles for document re-ranking according to individual preference will issue a more effective result set; The results for this investigation suggests that it does indeed issue a more effective result set (Table 6). Moreover, an examination on whether the use of keywords from profiles for query expansion and reformulation improves the search results was carried out. To this end the experiments suggests that query expansion does indeed improve the search results as hypothesized (Table 7). Therefore, conclusions can be derived that personalizing the book search application by incorporating user profiles such as background of personal tastes, interests, previously seen books, etc. does issue a more effective query result set as well as an effective book recommendation. In future works it will be worthwhile to use features such as recency of the books, example books, and the personal catalogue of the user which may help in retrieving a much more relevant result set. In addition it would be interesting to use sentiment analysis to determine the opinion or attitude of the user through the reviews they give for previously read books so that they are not returned a book similar to one which they have negatively reviewed.

## 6. REFERENCES

- [1] Arezki R., Poncelet P., Dray G., and Pearson D.W. *Information Retrieval Model Based on User Profile*, pages 490–499. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [2] Shardanand U. and Maes P. Social information filtering: Algorithms for automating “word of mouth”. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 210–217, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
- [3] Koolen M., Kazai G., Kamps J., Preminger M., Doucet A., and Landoni M. Overview of the INEX 2012 social book search track. In *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012*, 2012.
- [4] Koolen M., Bogers T., Gäde M., Hall M.M., Huurde-man H.C., Kamps J., Skov M., Toms E., and Walsh D. Overview of the CLEF 2015 social book search lab. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France, September 8-11, 2015, Proceedings*, pages 545–564, 2015.
- [5] Shokouhi M., Sloan M., Bennett P.N., Collins-Thompson K., and Sarkizova S. Query suggestion and data fusion in contextual disambiguation. In *Proceedings of the 24th International Conference on World Wide Web, WWW '15*, pages 971–980, Republic and Canton of Geneva, Switzerland, 2015.
- [6] Adomavicius G. and Tuzhilin A. User profiling in personalization applications through rule discovery and validation. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '99*, pages 377–381, New York, NY, USA, 1999. ACM.
- [7] Gauch S., Speretta M., Chandramouli A., and Micarelli A. *User Profiles for Personalized Information Access*, pages 54–89. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [8] Krulwich B. LIFESTYLE FINDER: intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2):37–45, 1997.
- [9] Middleton S.E., Shadbolt N.R., and De Roure D.C. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88, January 2004.
- [10] 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, 10th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2011, Saarbrücken, Germany, December 12-14, 2011, Revised Selected Papers*, pages 1–29, 2011.
- [11] Koolen M., Kazai G., Preminger M., and Doucet A. Overview of the INEX 2013 social book search track. In *Working Notes for CLEF 2013 Conference, Valencia, Spain, September 23-26, 2013*, 2013.
- [12] Koolen M., Bogers T., Kamps J., Kazai G., and Preminger M. Overview of the INEX 2014 social book search track. In *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014*, pages 462–479, 2014.
- [13] Zhang B.-W., Yin X.-C., Cui X.-P., Qu J., Geng B., Zhou F., and Hao H.-W. USTB at INEX2014: social book search track. In *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014*, pages 536–542, 2014.
- [14] Chaa M. and Nouali O. CERIST at INEX 2015: Social book search track. In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015*, 2015.
- [15] Harman D. Is the Cranfield Paradigm Outdated? In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1–1, New York, NY, USA, 2010. ACM.
- [16] Sanderson M. Test Collection Based Evaluation of Information Retrieval Systems. *Foundations and Trends in Information Retrieval*, 4(4):247–375, June 2010.
- [17] Voorhees E.M. The Philosophy of Information Retrieval Evaluation. In *Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, pages 355–370, London, UK, UK, 2002. Springer-Verlag.
- [18] Saravanan M., Raj P.C.R., and Raman S. Summarization and categorization of text data in high-level data cleaning for information retrieval. *Applied Artificial Intelligence*, 17(5-6):461–474, 2003.
- [19] Ounis I., Amati G., Plachouras V., He B., Macdonald C., and Johnson. Terrier Information Retrieval Platform. In *Proceedings of the 27th European Conference on IR Research*, volume 3408 of *Lecture Notes in Computer Science*, pages 517–519, Berlin, Heidelberg, 2005. Springer-Verlag.
- [20] Liu T.-Y. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, June 2009.
- [21] Macdonald C., Santos R.L.T., Ounis I., and He B. About learning models with multiple query-dependent features. *ACM Transactions on Information Systems (TOIS)*, 31(3):11:1–11:39, August 2013.
- [22] Macdonald C., Santos R.L.T., and Ounis I. The whens and hows of learning to rank for web search. *Information Retrieval*, 16(5):584–628, October 2013.
- [23] Metzler D. and Croft W.B. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, June 2007.
- [24] Burges C.J.C., Ragno R., and Le Q.V. Learning to rank with non-smooth cost functions. In *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, January 2007.