Personalized Book Recommendations Created by Using Social Media Data

Maria Soledad Pera, Nicole Condie, and Yiu-Kai Ng

Computer Science Department, Brigham Young University, Provo, Utah, U.S.A.
{soledadpera@gmail.com, nc2000@hotmail.com, ng@compsci.byu.edu}

Abstract. Book recommendation systems can benefit commercial websites, social media sites, and digital libraries, to name a few, by alleviating the knowledge acquisition process of users who look for books that are appealing to them. Even though existing book recommenders, which are based on either collaborative filtering, text content, or the hybrid approach, aid users in locating books (among the millions available), their recommendations are not personalized enough to meet users' expectations due to their collective assumption on group preference and/or exact content matching, which is a failure. To address this problem, we have developed *PBRecS*, a book recommendation system that is based on *social interactions* and *personal interests* to suggest books appealing to users. *PBRecS* relies on the *friendships* established on a social networking site, such as LibraryThing, to generate more personalized suggestions by including in the recommendations solely books that belong to a user's friends who share common interests with the user, in addition to applying *word-correlation factors* for partially matching book tags to disclose books *similar* in contents. The conducted empirical study on data extracted from LibraryThing has verified (i) the effectiveness of *PBRecS* using *social-media data* to improve the quality of book recommendations and (ii) that *PBRecS* outperforms the recommenders employed by Amazon and LibraryThing.

1 Introduction

A sophisticated and effective recommendation system that suggests items (such as books, movies, or news articles) that match users' interests can enhance the users' satisfaction and confidence in the performance of the system. These recommenders must ensure that users are presented with personalized contents that are tailored towards their particular interests. Even though suggestions provided by a recommendation system can lead users to access items that they are not aware of, existing approaches adopted for generating recommendations are not personalized enough to meet users' expectations [11].

To further enhance the performance of existing recommendation systems, we consider the premises given in [2, 3, 10]. Andersen et al. [2] claim that the quality of recommendations given to a user U can be improved by considering opinions of other users whom U trusts, whereas Carmel et al. [3] suggest that relying on users who belong to U's social network is a good practice in identifying U's preferences, since they share "common interests." More importantly, Guy et al. [10] assert that social media[1] can benefit personalized recommendation systems, since social media introduces new types of public data and metadata, such as ratings, comments, social connections, and user-defined tags, which can be employed to enhance the quality of recommendations.

While recommenders introduced in [18, 23] incorporate social-media data to increase the quality of tag and news article recommendations, respectively, to the best of our knowledge, there are no recommendation systems that consider *users' relationships* within a social network, in addition to *user-generated* data, such as tags, to recommend books. Pertinent recommendations on newly disclosed books that might be highly regarded by a user can aid the user in reducing the time and minimizing the efforts in identifying them (among the millions available), which can easily be left out. Book recommendation have been adopted by commercial websites, social media sites, and digital libraries, to name a few, to further enhance the search experience of customers (patrons, respectively) and facilitate their knowledge acquisition process by offering alternative choices to books they are interested in purchasing or have purchased (examining or have examined, respectively). In this paper, we introduce *PBRecS*, a personalized book recommendation system. *PBRecS* offers recommendations based on (i) the *relationships* of a user with other members of a social network, which is LibraryThing (Librarything.com)[2] in our case, (ii) *social data*, i.e., user-provided book tags in our case, and (iii) *word-correlation factors* [13], to develop a highly effective, personalized book recommendation system. *PBRecS* relies on an obvious assumption that each person P favors recommendations made by "trusted" friends more than recommendations provided by others who are unknown to P. Therefore, *PBRecS* locates, among the books belonged to the friends of a user posted on a social media site[3], books *similar* in content to a given one that the user is interested in which are determined by *word-correlation factors* on the same or different tags assigned to distinct books.

We have conducted an empirical study to validate the quality of recommendations created by *PBRecS*. The study has verified that by using *social-media data* the quality of books recommended by *PBRecS* is significantly higher than the ones created by Amazon(.com)'s and LibraryThing's recommender.

The remaining of this paper is organized as follows. In Section 2, we discuss existing (book) recommendation systems. In Section 3, we detail the design on *PBRecS*. In Section 4, we present the results of the empirical study conducted for assessing the performance of *PBRecS*. In Section 5, we give a conclusion.

---

[1] In recent years, social media sites, such as Facebook (facebook.com), Twitter (twitter.com), and Delicious (delicious.com), have become increasingly popular [10].

[2] LibraryThing was founded in 2006 for aiding users in cataloging and referencing books. LibraryThing users can rate and review books, add tags to available books to describe their content, and establish friendships with other LibraryThing users.

[3] The data required by *PBRecS* in performing the book recommendation task can be extracted from **any** social media site, providing that users' relationships and book tags can be obtained from the site, where *PBRecS* serves as its book recommender.

2 Related Work

Machine learning, information retrieval, natural language processing, and probabilistic techniques have been adopted to develop systems that recommend (web) documents [9], song/music tracks [4], videos [17], and movies [14], to name a few.

As defined in [7], a recommendation system suggests items (i.e., products or actions) to an end-user U. *Content-based* and *collaborative filtering* are two well-known recommendation methods [20]. The former creates a user profile to represent the preferences of U using words, phrases, or features, which defines the items of interest to U, whereas the later identifies the group of people who have similar (items) preferences as U's and recommends items that the group is interested in. More importantly, recent publications [7, 20] present various hybrid approaches that exploit the benefits of using both content-based and collaborative filtering methods to improve the quality of recommendations. An in-depth discussion of various content-based, collaborative filtering, and hybrid recommendation systems can be found in [1].

There exist a number of book recommendation systems [16, 20, 24], among which the one used by Amazon is a popular one [8]. Amazon's recommender, as presented in [16], suggests items to a user that are similar to other users' past purchased and/or rated items, i.e., items that appear in the purchase patterns of various users.

Yang et al. [24] rely on a ranking-oriented collaborative filtering approach, which considers users' preferences on digital library resources extracted from users' access logs to perform the recommendation task. This filtering approach overcomes the problem that arises due to the sparseness of explicit users' ratings, i.e., lack of initial information to perform the recommendation task, in predicting digital library materials of interest to a user.

Park and Chang [20] create a user-profile P based on individual and group behavior information, such as clicks, shopping habits, purchases, and interest fields, for generating book recommendations. Using P, the authors compute the Euclidean distance between P and each product profile, which describes product features, and select products for which their Euclidean distances are the closest to P. Additional references on book recommendation systems can be found in [15].

The authors of [10, 18, 23] use social-media data to enhance the performance of recommendation systems. Wang et al. [23] consider an original news posting, along with the comments made by users on the posting, to generate a list of recommended news articles for a particular news thread. The authors of [18] develop a personalized collaborative filtering algorithm that combines collaborative information extracted from social tagging systems, such as Delicious, and the user's personalized tag preferences for tag recommendation. The graph-based approach in [18] captures the social relations among users and the similarities between resources, such as bookmarks, and applies a random-walk method that explores the structure of the created graph to generate tag recommendations. Guy et al. [10] present a personalized recommendation system on social items (such as blogs posts and bookmarks), which considers the relationships between people, items, and (item) tags in making recommendations.

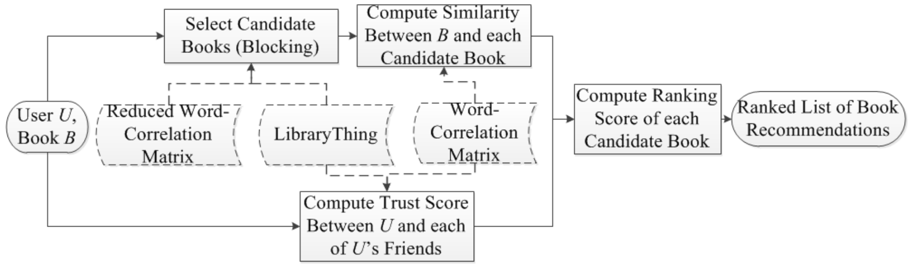

Fig. 1. Processing steps of the proposed book recommender, *PBRecS*

3 Our Proposed Recommender

In this section, we present our proposed recommender, *PBRecS*, which generates personalized book recommendations by (i) applying word-correlation factors (as defined in Section 3.1) on the data extracted from a social networking site to determine books with similar content and (ii) considering social interactions among users. The overall process of *PBRecS* is illustrated in Figure 1.

PBRecS processes data extracted from *LibraryThing*, which is a library social network site. LibraryThing is an innovative, well-designed, and (to the best of our knowledge) the most popular social application that was set up solely for cataloging books [22]. As of September 28, 2010, LibraryThing archives 5,667,984 unique records (on books), and approximately 1,197,659 users have added more than 68.6 million tags to different book records at LibraryThing, according to the Zeitgeist Overview (librarything.com/zeitgeist) which provides official statistical data of LibraryThing. Each LibraryThing user has a *personal catalog* that includes books (s)he owns, or is interested in. In addition, a user can assign individual tags to books included in his/her personal catalog, which are treated as personalized book identifiers of the contents of books. Each LibraryThing user U also has a *profile* which includes personal information such as a list of other LibraryThing users who were explicitly chosen by U to be his/her friends and links to book reviews created by U. Moreover, each book B on LibraryThing is associated with a *tag cloud*, which is a global visual representation of tags assigned to B by LibraryThing users, in addition to their frequencies of occurrence.

Given a particular LibraryThing user LT_User and a book, denoted $Source\_Book$, which has either been added by LT_User to his/her personal catalog or browsed by him/her on LibraryThing, *PBRecS* (i) identifies LT_User's friends and (ii) determines the set of books, denoted $Candidate\_Set$ (among those included in the personal catalogs of LT_User's friends), that are similar to $Source\_Book$ (as detailed in Section 3.2). Hereafter, *PBRecS* computes a *ranking score* (as defined in Section 3.3), which is based on (i) the *degree of resemblance* between (the tags representing) $Source\_Book$ and (the tags representing) a book in $Candidate\_Set$, and (ii) the *degree of closeness* between LT_User and each of LT_User's friends who possesses the book in his/her personal catalog, to select books among $Candidate\_Set$ to be recommended to LT_User. The top-N

($N \geq 1$) books in $Candidate\_Set$ for which their ranking scores are the highest are recommended to LT_User.

3.1 Word Correlation Factors

PBRecS relies on the pre-computed word-correlation factors in the word-correlation matrix [13] to determine the similarity between (the content of) books using their corresponding tags. The *word-correlation factors* were generated using a set of approximately 880,000 Wikipedia documents (wikipedia.org), and each correlation factor indicates the *degree of similarity* of the two corresponding words[4] based on their (i) *frequency of co-occurrence* and (ii) *relative distances* in each Wikipedia document. Wikipedia documents were chosen for constructing the word-correlation matrix, since they were written by more than 89,000 authors (i) with different writing styles, (ii) using various terminologies that cover a wide range of topics, and (iii) with diverse word usage and content. Furthermore, the words in the matrix are common words in the English language that appear in various online English dictionaries, such as 12dicts-4.0 (prdownloads.sourceforge.net/wordlist/12dicts-4.0.zip), Ispell (cs.ucla.edu/geoff /ispell.html), and BigDict (packetstormsecurity.nl/Crackers/bigdict.gz).

The word-correlation matrix is a 57,908 × 57,908 *symmetric* matrix, and the word-correlation factor of any two words, i and j, denoted $wcf(i, j)$, is defined as

$$wcf(i,j) = \frac{\sum_{w_i \in V(i)} \sum_{w_j \in V(j)} \frac{1}{d(w_i,w_j)+1}}{|V(i)| \times |V(j)|} \tag{1}$$

where $d(w_i, w_j)$ is the *distance* between any two words w_i and w_j in any Wikipedia document D, $V(i)$ ($V(j)$, respectively) is the set of words that includes i (j, respectively) and its stem variations in D, and $|V(i)| \times |V(j)|$ is the *normalization factor* of $wcf(i, j)$.

Compared with synonyms and related words compiled by WordNet (wordnet.princeton.edu) in which pairs of words are not assigned similarity weights, word-correlation factors provide a more sophisticated measure of word similarity.

3.2 Selecting Candidate Books

Since the number of books in the personal catalogs of LT_User's friends can be large, i.e., in the thousands, it is not practical to compare each of these books with $Source\_Book$ to identify the ones to be recommended to LT_User, which could significantly prolong the processing time. To minimize the time for performing the comparisons, *PBRecS* applies a *blocking strategy*[5] on the

[4] Words in the Wikipedia documents were *stemmed* after all the *stopwords*, such as articles, conjunctions, and prepositions, which do not play a significant role in representing the content of a document, were removed. From now on, unless stated otherwise, (key)words refer to nonstop, stemmed words.

[5] A *blocking strategy* [12] is a filtering technique which reduces the potentially very large number of comparisons to be made among records [5].

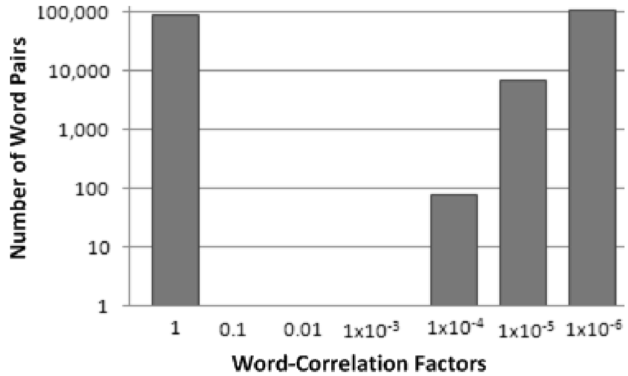

Fig. 2. Distribution of word-correlation factors in the reduced word-correlation matrix

books posted under the personal catalogs of LT_User's friends to retrieve a subset of potential books, i.e., $Candidate\_Set$, to be recommended. Books in $Candidate\_Set$ are represented by tags, such that at least one of their tags either *exactly matches* or is *highly similar* to one of the tags of $Source\_Book$ assigned by LT_User. In case when there are no personal tags assigned to $Source\_Book$ by LT_User, *PBRecS* relies on the top-3 tags, i.e., the tags with the highest frequency of occurrence, in the tag cloud of *Source_Book* to perform the blocking task. The top-3 tags are chosen, since we have observed that LibraryThing users have assigned on the average three tags to each book in their personal catalogs.

To select books to be included in *Candidate_Set*, *PBRecS* relies on a *reduced* version of the word-correlation matrix (introduced in Section 3.1) which contains 13% of the most frequently-occurring words (based on their frequencies of occurrence in the Wikipedia documents), and for the remaining 87% of the less-frequently-occurring words only the exact-matched correlation factor, i.e., 1, is used. The distribution of the word-correlation factors among different word pairs in the *reduced* matrix is illustrated in Figure 2 . As shown in the figure, the word-correlation factors of non-identical words are in the range of 1×10^{-4} and 1×10^{-6}, and word pairs with correlation factors *closer* to (*lower*, respectively) 1×10^{-4} are treated as *relatively* (*less*, respectively) *similar*. Adopting a reduced word-correlation matrix, instead of using the word-correlation matrix, in selecting a subset of books, the overall processing time can be significantly reduced without affecting the matching accuracy [21].

3.3 Ranking Score

$PBRecS$ ranks the books in $Candidate\_Set$ to prioritize them for recommendations. In accomplishing this task, $PBRecS$ considers not only the degree of resemblance of each book CB in $Candidate\_Set$ with respect to $Source\_Book$, but also the relative degree of interest on $Source\_Book$ that LT_User's friends, who include CB in their personal catalogs, exhibit.

Similarity Among Books To determine the similarity between *Source_Book* and *CB*, *PBRecS* computes their *degree of resemblance* by adding the word-correlation factors between each tag in the tag cloud (provided by LibraryThing) of *Source_Book* and *CB*, respectively. In computing the resemblance score, *PBRecS* relies on the word-correlation matrix introduced in Section 3.1, instead of the reduced word-correlation matrix employed in Section 3.2, since the former provides a *more accurate* similarity measure between (tags representing) *Source_Book* and *CB* than the reduced matrix. The *degree of resemblance*, denoted *Resem*, between *Source_Book* and *CB* is defined as

$$Resem(Source\_Book, CB) = \sum_{i=1}^{n} Min\left\{\sum_{j=1}^{m} wcf(Source\_Book_i, CB_j), 1\right\} \quad (2)$$

where n (m, respectively) is the sum of the *frequency of occurrence* of each distinct tag in (the tag cloud of) *Source_Book* (*CB*, respectively), $Source\_Book_i$ (CB_j, respectively) is a tag in the tag cloud of *Source_Book* (*CB*, respectively), and $wcf(Source\_Book_i,\ CB_j)$ is the correlation factor of $Source\_Book_i$ and CB_j in the word-correlation matrix. By considering the frequency of occurrence of tags, *PBRecS* ensures that if tags assigned to *Source_Book* are similar to the most descriptive, i.e., frequent, tags representing *CB*, then the corresponding resemblance score should be higher than tags that occur infrequently.

The *Min* function in Equation 2 imposes a constraint on summing up the word-correlation factors of tags representing *Source_Book* and *CB*. Even if a tag in the tag cloud of *CB* (i) matches exactly one of the tags in the tag cloud of *Source_Book* and (ii) is similar to some of the remaining tags describing *Source_Book*, which would yield a value greater than 1.0, *PBRecS* limits the sum of their word-correlation factors to 1.0. This constraint ensures that if *CB* contains a dominant tag T in its tag cloud, i.e., T is highly similar to (or the same as) a couple of tags in the tag cloud of *Source_Book*, T alone cannot significantly impact the resemblance value of *Source_Book* and *CB*, i.e., "one" does not represent "all". Tags assigned to *CB* that are similar to most of the tags of *Source_Book* yield a higher degree of resemblance of *Source_Book* and *CB* than tags assigned to *Source_Book* that are similar to only one dominant tag representing *CB*.

Interests Among Friends LibraryThing friends of the owner of *Source_Book*, *LT_User*, might be interested in books on various subject areas, e.g., religion, politics, fiction, science, etc., and not all the friends should be given the same "weight" (or the same level of trust) in providing recommendations for books (highly) similar to *Source_Book*, which belongs to a particular subject area. *PBRecS* measures the relative degree of interest (closeness) of *LT_User*'s friends on *Source_ Book* whose personal catalog include a book in *Candidate_Set*. The *degree of closeness*, denoted *Close*, between *LT_User* and one of his/her friends, denoted *LT_Friend*, measures the interest of *LT_Friend* on *Source_Book* based on the number of closely related tags on books in his/her personal catalog with respect to the tags assigned by LT_User to *Source_Book*.

$$Close(Source\_Book,\ LT\_Friend) = \sum_{i=1}^{r} Min\left\{\sum_{j=1}^{s} wcf(Source\_Book_i, LT\_Friend_j), 1\right\} \quad (3)$$

where r (s, respectively) denotes the sum of the frequency of occurrence of each tag assigned by LT_User to *Source_Book* (by *LT_Friend* to a book in his/her personal catalog, respectively), $Source\_Book_i$ (LT_Friend_j, respectively) is a tag assigned by LT_User to *Source_Book* (by *LT_Friend* in describing some books in his/her personal catalog, respectively), and $wcf(Source\_Book_i,\ LT\_Friend_j)$ is the correlation factor between $Source\_Book_i$ and LT_Friend_j.

Recommendations With the *degree of resemblance* (*closeness*, respectively) between *Source_Book* and each book CB in *Candidate_Set* (*LT_Friend*, who posts CB in his/her personal catalog, respectively), *PBRecS* computes the *rank-(ing) score* of CB using the Joint Product [19] as follows.

$$Rank(CB) = Resem(Source\_Book, CB) \times Close(Source\_Book, LT\_Friend) \quad (4)$$

The Top-N ($N \geq 1$) books with the *highest ranking score* are recommended to LT_User. *PBRecS* sets $N = 10$, which follows the number of recommendations presented by LibraryThing to its users. Note that *PBRecS* does not include duplicate books in the generated recommendations. If multiple friends of LT_User possess a book CB (in *Candidate_Set*) in their personal catalogs, only the highest ranking score of CB will be considered during the recommendation process.

Example 1 Consider the book "Emma" by Jane Austen and a LibraryThing user, Soleenusa, who is one of the independent appraisers of *PBRecS* interested in "Emma". Based on the books included in the personal catalogs of Soleenusa's LibraryThing friends, *PBRecS* suggests 10 books that might also be of interest to Soleenusa. As shown in Figure 3, except for the 9^{th} recommended book, all the remaining (in bold) recommendations are considered closely related to "Emma", since books 1 to 7 are also written by Jane Austen and are in the same subject area of "Emma", which is a classical novel, whereas the stories portrayed in books 8 and 10 occur in the same time period as "Emma" and include characters from Jane Austen's popular novels. Compared with the books recommended by Amazon and LibraryThing for "Emma", only 4 and 5 of the recommendations generated by Amazon and LibraryThing, respectively can be treated as closely related (as shown in Figure 3). The remaining recommended books, such as "The Odyssey", which is a Greek epic poem written by Homer, "Treasure island", which is an adventure novel by Robert L. Stevenson, and "Jane Eyre" and "Wuthering Heights" by Charlotte and Emily Brontë, respectively, which are dramatic classical novels written in a style significantly different from Jane Austen, are dissimilar to "Emma", as claimed by Soleenusa. □

"Emma" by Jane Austen		
PBRecS	*Amazon*	*LibraryThing*
1. **Persuasion**	1. **Sense and Sensibility**	1. **Northanger Abbey**
2. **Northanger Abbey**	2. **Pride and Prejudice**	2. **Lady Susan/ Sandition/The Watsons**
3. **Mansfield Park**	3. Alice's Adventures in Wonderland	3. **Mansfield Park**
4. **Sense and Sensibility**	4. **Great Expectations**	4. Villette
5. **Pride and Prejudice**	5. The Odyssey	5. Jane Eyre
6. **The Oxford Illustrated Jane Austen** (Six Volume Set)	6. Lady Chatterley's Lover	6. Wuthering Heights
7. **Minor Works of Jane Austen**	7. Treasure Island	7. The Tenant of Wildfell
8. **Duty and Desire: A Novel of Fitzwilliam Darcy, Gentleman**	8. Jane Eyre	8. **Vanity Fair**
9. The Meaning of Night: A Confession	9. Wuthering Heights	9. Tess of D'Ubervilles
10. **Darcy's Story**	10. **Cranford**	10. **Middlemarch**

Fig. 3. Recommendations generated by *PBRecS*, Amazon, and LibraryThing, respectively in response to the book "Emma", by Jane Austen

4 Experimental Results

In this section, we first introduce the data and metrics in Sections 4.1 and 4.2, respectively which were used for assessing the performance of *PBRecS*. Thereafter, we detail the results of the empirical study conducted for evaluating *PBRecS*, in addition to comparing its performance with other existing book recommenders in Section 4.3.

4.1 Experimental Data

To analyze the performance of *PBRecS*, we rely on data extracted from LibraryThing that contain personal information of a group of independent appraisers who are LibraryThing users, which include (i) (tags of) books in their personal catalogs, (ii) lists of their friends, and (iii) (tags of) books posted under their friends' personal catalogs. In addition, the extracted data include the tag cloud of each book listed in (i) and (iii) above created using tags specified by LibraryThing users.

To the best of our knowledge, there is no dataset available for assessing the performance of personalized book recommendation systems, and thus we rely on the independent appraisers who manually examined the relatedness of each one of the top-10 recommendations generated by *PBRecS* with respect to each of the books in their personal catalogs, yielding a set of 100 books, denoted $Test\_Books$, used in our empirical study.

4.2 Evaluation Metrics

To evaluate the effectiveness of *PBRecS* in generating high-quality, personalized book recommendations, we apply three well-known information retrieval metrics, the (overall) *Accuracy*, *Precision at K*, and *Mean Reciprocal Rank* [6].

$$Accuracy = \frac{\sum_{i=1}^{N} \frac{Number\_of\_Related\_Recommendations_i}{10}}{N} \quad (5)$$

where N is the number books in $Test\_Books$, i is a book in $Test\_Books$, 10 is the number of book recommendations generated by $PBRecS$[6] for book i, and $Number\_of\_Related\_Recommendations_i$ is the number of recommendations out of 10 that are evaluated as *related* to book i by a particular appraiser who owns i.

The $P@K$ value quantifies the top-K ranked recommended books for a particular book in $Test\_Books$, which measures the overall user's satisfaction with the top-K recommendations (generated by $PBRecS$).

$$P@K = \frac{\sum_{i=1}^{N} \frac{Number\_of\_Related\_Recommendations_i}{K}}{N} \quad (6)$$

where K is the (pre-defined) number of book recommendations to be considered, and N, i, and $Number\_of\_Related\_Recommendations_i$ are as defined in Equation 5. Note that in our study, we set $K = 1$, 5, and 10, which evaluate the relatedness of the recommendations positioned at the *top*, *middle*, and *overall* in the ranking, respectively.

The *Mean Reciprocal Rank* (MRR) of the ranked book recommendations generated by $PBRecS$ is the averaged sum of the ranking values for the recommendations computed for each book in $Test\_Books$ such that each ranking value is either the reciprocal of the ranking position of the *first* related recommendation among the top-10 recommendations, if there is any, or 0, otherwise.

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{r_i} \quad (7)$$

where r_i is the (position in the) rank of the *first related* recommendation with respect to book i in $Test\_Books$, if it exists, and N and i are as defined in Equation 5.

While the *accuracy* measures the overall user's satisfaction of the recommendations created by $PBRecS$, $P@K$ and MRR evaluate the ranking strategy of $PBRecS$, since the higher the related recommendations are ranked, the higher their corresponding $P@K$ and MRR scores should be.

4.3 Performance Evaluation and Comparisons

In this section, we present the experimental results achieved by $PBRecS$ and compare its performance with the recommendation systems of Amazon and LibraryThing[7], which are two well-known, commercial book recommenders. While

[6] As stated in Section 3.3, we only evaluate the top-10 book recommendations generated by $PBRecS$.

[7] From now on, unless stated otherwise, whenever we mention Amazon (LibraryThing, respectively), we mean Amazon's (LibraryThing's, respectively) book recommender.

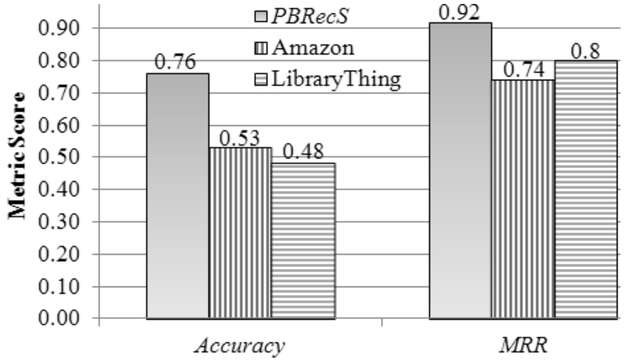

Fig. 4. (Average) Accuracy ratios and MRR scores based on (not) related books recommended by $PBRecS$, Amazon, and LibraryThing for the books in $Test\_Books$

the recommender of Amazon has been introduced in Section 2, the recommendation system of LibraryThing (i) compares books in a user's personal catalog with thousands of books in other users' catalogs, (ii) considers common tags assigned to (the tag clouds of) books, and (iii) identifies books with similar Library of Congress Subject Heading and/or Classification to provide a list of books a user might be interested in. (A detailed discussion on LibraryThing's recommender system can be found in http://www.librarything.com/wiki/index.php /Automatic_recommendations).

In comparing $PBRecS$ with Amazon and LibraryThing, we rely on the same group of independent appraisers (as discussed in Section 4.1) who determine which one of the top-10 books recommended by $PBRecS$, Amazon, and LibraryThing, respectively for each book B in $Test\_Books$ is related to B. Note that since $PBRecS$ is based on the premise that a user U tends to trust recommendations made by his/her friends, books recommended by $PBRecS$ to U are books in the personal catalogs of U's friends, whereas books recommended by Amazon (LibraryThing, respectively) are extracted from the entire collection of books available at Amazon (LibraryThing, respectively).

Assessment To assess the overall performance of $PBRecS$ (Amazon and LibraryThing, respectively), in terms of the (average) *accuracy* of the top-10 recommendations generated by $PBRecS$ (Amazon and LibraryThing, respectively) for each book B in $Test\_Books$, we rely on the recommended books labeled as (*not*) *related* to B by each independent appraiser. As shown in Figure 4, $PBRecS$ achieves an accuracy ratio of 76%, which surpasses the accuracy ratio of Amazon (LibraryThing, respectively) by 23% (28%, respectively).

Besides accuracy, we have also computed the $P@K$ scores on the top-10 book recommendations generated by $PBRecS$, Amazon, and LibraryThing, respectively for each book in $Test\_Books$, again based on independent appraisers' evaluations. As shown in Figure 5, the $P@1$ score of $PBRecS$, which is 0.87,

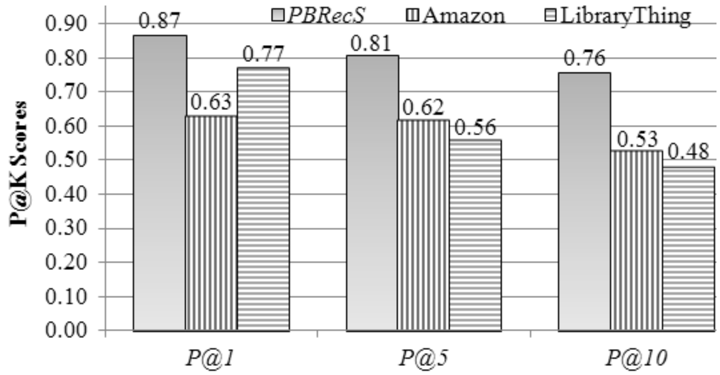

Fig. 5. *Precision*@K (K = 1, 5, and 10) scores on the (top-10) recommendations achieved by *PBRecS*, Amazon, and LibraryThing for the books in *Test_Books*

indicates that among the 87 out of 100 books in *Test_Books*, their first recommended books generated by *PBRecS*, i.e., the books with the highest ranking score, were treated as *related*. A high P@1 score implies that the ranking strategy of *PBRecS* is highly effective in presenting the first recommended books that users are interested in. On the other hand, the P@1 scores achieved by Amazon and LibraryThing on the top-10 recommendations generated for books in *Test_Books* are 0.63 and 0.77, respectively, which are at least 10% lower compared with *PBRecS*'s P@1 score.

As previously stated, P@5 measures the overall user satisfaction with respect to the top-5 recommended books. Figure 5 shows that the P@5 score of *PBRecS* is at least 19% higher than the P@5 scores of Amazon and LibraryThing, respectively. The outcome demonstrates that *PBRecS*, in general, positions higher in the list of recommendations books that are related to a given book for a particular user than Amazon and LibraryThing, respectively. The P@10 scores of *PBRecS*, Amazon, and LibraryThing are 0.76, 0.53, and 0.48, respectively, as shown in Figure 5. Based on the P@10 values, on the average, close to 8 out of the 10 books recommended by *PBRecS* are perceived as related recommendations, as opposed to the five recommended by Amazon and LibraryThing. Note that since we only evaluated the top-10 recommendations generated by a book recommendation system, its P@10 score is the same as its accuracy score.

Besides the *accuracy* and P@K scores, we have also assessed the performance of *PBRecS* (Amazon and LibraryThing, respectively) based on the *MRR* metric. As shown in Figure 4, the *MRR* scores computed for *PBRecS*, Amazon, and LibraryThing are 0.92, 0.74, and 0.80, respectively, which reflect that while on the average users of *PBRecS* are required to browse through the top ($\cong \frac{1}{0.92} = 1.08$) generated recommendations before locating one that is related to a book that (s)he owns or is examining, Amazon's and LibraryThing's users, on the other hand, scan through at least one ($\cong \frac{1}{0.74} = 1.35$ and $\cong \frac{1}{0.8} = 1.25$, respectively) recommended book before identifying one that is appealing to them.

Observations It is worth mentioning that *PBRecS* always presents to users ten recommendations for each given book, as opposed to Amazon and LibraryThing, which occasionally generate less than ten recommendations, the expected number of recommendations. Furthermore, at least one of the top-10 recommendations generated by *PBRecS* for each book in *Test_Books* is treated as *related* to the corresponding book by the appraisers. However, Amazon (LibraryThing, respectively) generated either (i) no recommendations at all or (ii) no related recommendations for 8 (23, respectively) books in *Test_Books*.

As illustrated in Figures 4 and 5, *PBRecS* is more effective in recommending books that satisfy the personal interest of a user than Amazon or LibraryThing does, which supports our claim that considering (i) data extracted from a social media site along with (ii) the personal interactions of a user in a social environment enriches the quality of recommended books.

5 Conclusions

In this paper, we have introduced a book recommendation system, denoted *PBRecS*. Unlike existing book recommenders, such as the recommendation system employed by LibraryThing, which present the *same* recommendations to users that share the same profile information or common interests, *PBRecS* (i) considers the existence of user-defined tags and friendships among users on a social networking site, and (ii) uses word-correlation factors for computing exact, as well as partial, matches among tags representing books to locate the ones that share similar contents, to generate book recommendations, which are *tailored* to the interests of a particular user.

To assess the quality of book recommendations generated by *PBRecS*, we have conducted an empirical study using data extracted from LibraryThing to evaluate *PBRecS* and compare its performance with the recommenders of Amazon and LibraryThing, respectively. The study has verified that *PBRecS* outperforms the recommenders adopted by Amazon and LibraryThing in generating *high-quality*, *personalized* books.

While *PBRecS* is currently applied for recommending books, we intent to extend the proposed recommender so it can suggest items in various domains, such as products and movies, provided that social networking data describing items of interest and friendships among users are available.

References

[1] Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE TKDE **17(6)** (2005) 734–749

[2] Andersen, R., Borgs, C., Chayes, J., Feige, U., Flaxman, A., Kalai, A., Mirrokni, V., Tennenholtz, M.: Trust-based Recommendation Systems: an Axiomatic Approach. Proc. of WWW (2008) 199–208

[3] Carmel, D., Zwerdling, N., Guy, I., Ofek-Koifman, S., Har'el, N., Ronen, I., Uziel, E., Yogev, S., Chernov, S.: Personalized Social Search Based on the User's Social Network. Proc. of CIKM (2009) 1227–1236

[4] Chen, H., Chen, A.: A Music Recommendation System Based on Music Data Grouping and User Interests. Proc. of CIKM (2001) 231–238

[5] Christen, P.: Automatic Record Linkage Using Seeded Nearest Neighbor and Support Vector Machine Classification. Proc. of SIGKDD (2008) 151–159

[6] Croft, W., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in Practice. Addison Wesley (2010)

[7] de Campos, L., Fernandez-Luna, J., Huete, J., Rueda-Morales, M.: Combining Content-Based and Collaborative Recommendations: A Hybrid Approach Based on Bayesian Networks. Approximate Reasoning **51(7)** (2010) 785–799

[8] Givon, S., Lavrenko, V.: Predicting Social-Tags for Cold Start Book Recommendations. Proc. of ACM RecSys (2009) 333–336

[9] Guan, Z., Wang, C., Bu, J., Chen, C., Yang, K., Cai, D., He, X.: Document Recommendation in Social Tagging Services. Proc. of WWW (2010) 391–400

[10] Guy, I., Zwerdling, N., Ronen, I., Carmel, D., Uziel, E.: Social Media Recommendation Based on People and Tags. Proc. of SIGIR (2010) 194–201

[11] Jung, J., Kim, K., Lee, H., Park, S.: Are You Satisfied with Your Recommendation Service?: Discovering Social Networks for Personalized Mobile Services. Proc. of KES-AMSTA (2008) 567–573

[12] Kelley, R.: Blocking Considerations for Record Linkage Under Conditions of Uncertainty. Proc. of Social Statistics Section (1984) 602–605

[13] Koberstein, J., Ng, Y.-K.: Using Word Clusters to Detect Similar Web Documents. Proc. of KSEM (2006) 215–228

[14] Koren, Y.: Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model. Proc. of ACM SIGKDD (2008) 426–434

[15] Li, H., Gu, Y., Koul, S.: Review of Digital Library Book Recommendation Models. Available at SSRN (2009) http://ssrn.com/abstract=1513415

[16] Linden, G., Smith, B., York, J.: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing **7(1)** (2003) 76–80

[17] Liu, C., Jiang, S., Huang, Q.: Personalized Online Video Recommendation by Neighborhood Score Propagation Based Global Ranking. Proc. of ICIMCS (2009) 244–253

[18] Liu, L., Fang, B., Zhang, W.: Speak the Same Language with Your Friends: Augmenting Tag Recommenders with Social Relations. Proc. of ACM HT (2010) 45–50

[19] Luger, G.: Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 5^{th} Edition. Addison Wesley (2005)

[20] Park, Y., Chang, K.: Individual and Group Behavior-based Customer Profile Model for Personalized Product Recommendation. Expert Systems with Applications **36(2)** (2009) 1932–1939

[21] Pera, M., Lund, W., Ng, Y.-K.: A Sophisticated Library Search Strategy Using Folksonomies and Similarity Matches. JASIST **60(7)** (2009) 1392–1406

[22] Smith, F.: LibraryThing. Reference Reviews **21(8)** (2007) 5–6

[23] Wang, J., Li, Q., Chen, Y.: User Comments for News Recommendation in Social Media. Proc. of ACM SIGIR (2010) 881–882

[24] Yang, C., Wei, B., Wu, J., Zhang, Y., Zhang, L.: CARES: a Ranking-oriented CADAL Recommender System. Proc. of ACM/IEEE-CS (2009) 203–212