With a Little Help From My Friends: Generating Personalized Book Recommendations Using Data Extracted from a Social Website

Maria Soledad Pera
Computer Science Department
Brigham Young University
Provo, Utah, U.S.A.
Email: soledadpera@gmail.com

Yiu-Kai Ng
Computer Science Department
Brigham Young University
Provo, Utah, U.S.A.
Email: ng@copmsci.byu.edu

***Abstract*—With the large amount of books available nowadays, users are overwhelmed with choices when they attempt to find books of interest. While existing book recommendation systems, which are based on either collaborative filtering, content-based, or hybrid methods, suggest books (among the millions available) that might be appealing to the users, 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 $PReF$, a Personalized Recommender that relies on Friendships established by users on a social website, such as LibraryThing, to make book recommendations *tailored* to individual users. In selecting books to be recommended to a user U, who is interested in a book B, $PReF$ (i) considers books belonged to U's friends, (ii) applies *word-correlation factors* to disclose books *similar* in contents to B, (iii) depends on the ratings given to books by U's friends to identify highly-regarded books, and (iv) determines how reliable individual friends of U are in providing books from their own catalogs (that are similar in content to B) to be recommended. We have conducted an empirical study and verified that (i) relying on data extracted from social websites improves the effectiveness of book recommenders and (ii) $PReF$ outperforms the recommenders employed by Amazon and LibraryThing.**

***Keywords*-recommendation, personalization, word-similarity**

I. Introduction

Book recommenders have been adopted by online shopping companies, social websites, and digital libraries to further facilitate their users' knowledge acquisition process by offering alternative choices (among the millions available) of books they are likely interested in. While suggestions provided by existing book recommenders can introduce users to books that they are not aware of, these recommenders are not personalized enough to achieve their design goals. To address this problem, we introduce $PReF$, a personalized book recommender that depends on friendships established among users in a social website, which is LibraryThing(.com)[1] in our case, to generate book recommendations *tailored* to individual users' interests. $PReF$ locates, among the books bookmarked by U's friends on a social website, the ones that are *similar* in content to a given book B that U is interested in. Hereafter, $PReF$ ranks the candidate books to be recommended by considering not only the *content similarity* between each candidate book CB and B, but also the *ratings* assigned to CB by U's friends, and the *reliability* of each of U's friends.

$PReF$ is a unique system that relies on (i) *relationships* established between a user U and other members of a social website, since as stated in [2], the quality of recommendations given to U is improved by considering opinions of other users whom U trusts, (ii) *ratings* provided by users of a social site, which aid in identifying highly-regarded books U might be interested in, and (iii) *word-correlation factors* [4], which detect books *similar* in content, even if they are described using analogous, but not the same, tags, to generate personalized book recommendations.

We have conducted an empirical study using data extracted from LibraryThing to validate the effectiveness of personalized book recommendations made by $PReF$. The study has verified $PReF$ is significantly more effective than (the recommenders used at) Amazon and LibraryThing in recommending books that individual users are interested in.

The remaining of this paper is organized as follows. In Section II, we detail the design of $PReF$. In Section III, we present the results of the empirical study conducted for assessing the performance of $PReF$. In Section IV, we give a conclusion and directions for future work.

II. Our Proposed Book Recommender

$PReF$, our book recommender, employs data extracted from LibraryThing to generate personalized book recommendations. Each LibraryThing user U has a *personal catalog* that includes books (s)he owns or is interested in. In addition, U can assign tags to books included in his/her catalog, which serve as personalized identifiers of the contents of the books. To indicate how highly regarded a book B in the catalog is, U assigns a *rating* to B, which is a numerical value between '1' and '5', such that '5' is

[1] LibraryThing.com was founded in 2006 for aiding users in cataloging and referencing books. LibraryThing users can rate and review books, add tags to books to describe their contents, and establish friendships, i.e., bi-directional relationships, with other LibraryThing users.

978-0-7695-4513-4/11 $26.00 © 2011 IEEE
DOI 10.1109/WI-IAT.2011.9

the highest and '1' is the lowest. Moreover, U has a *profile* which includes a list of other LibraryThing users who were explicitly chosen by U to be his/her friends. In LibraryThing, each book B is associated with (i) a *tag cloud*, which is a global visual representation of tags (and their frequencies) assigned to B by LibraryThing users who include B in their catalogs, and (ii) a *global rating*, which *averages* the ratings assigned to B by LibraryThing users. The overall process of $PReF$ is described as follows.

A. Selecting Candidate Books

Given a LibraryThing user, denoted LT_User, and a book, denoted $Source\_Bk$, which has been added by LT_User to his/her personal catalog or browsed by LT_User on LibraryThing, $PReF$ identifies LT_User's friends and determines the set of books, among those included in the personal catalogs of LT_User's friends, that are similar to $Source\_Bk$. As the number of books in the personal catalog of each LT_User's friend can be large, which can be in the thousands, it is not practical to compare each book with $Source\_Bk$ to identify the ones to be recommended to LT_User, since the comparisons significantly prolong the processing time. To minimize the number of comparisons, $PReF$ applies a *blocking strategy* on the books posted under the personal catalogs of LT_User's friends to yield $Candidate\_Set$, the subset of books considered for recommendation. At least one of the tags of each book in $Candidate\_Set$ *matches exactly* or is *highly similar* to one of the tags of $Source\_Bk$ assigned by LT_User. As books in $Candidate\_Set$ and $Source\_Bk$ share the same (or analogous) tags, $PReF$ expects books in $Candidate\_Set$ to be similar in content (to a certain degree) to $Source\_Bk$. To identify highly similar tags, $PReF$ employs a *reduced* version of the word-correlation matrix in [4] 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., 1.0, is used.

B. Ranking LibraryThing Books

$PReF$ ranks each book CB in $Candidate\_Set$ to prioritize them for recommendations using (i) the *degree of resemblance* of CB and $Source\_Bk$ (in Section II-B1), (ii) the *rating score* assigned to CB by each friend of LT_User (in Section II-B2), and (iii) the *degree of reliability* of each of LT_User's friends (in Section II-B3).

1) Similarity Among Books: To determine the (content) similarity between $Source\_Bk$ and CB, $PReF$ computes their *degree of resemblance* by adding the word-correlation factors between each tag in the tag cloud (provided by LibraryThing) of $Source\_Bk$ and CB, respectively using the word-correlation matrix in [4]. The *degree of resemblance*, denoted $Resem$, between $Source\_Bk$ and CB is defined as

$$Resem(Source\_Bk,\ CB) = \frac{\sum_{i=1}^{n} Min\{\sum_{j=1}^{m} wcf(Source\_Bk_i, CB_j), 1\} \times freq_i}{\sum_{i=1}^{n} freq_i} \quad (1)$$

where n (m, respectively) is the number of distinct tags in (the tag cloud of) $Source\_Bk$ (CB, respectively), $Source\_Bk_i$ (CB_j, respectively) is a tag in the tag cloud of $Source\_Bk$ (CB, respectively), $wcf(Source\_Bk_i,\ CB_j)$ is the correlation factor of $Source\_Bk_i$ and CB_j in the word-correlation matrix, and $freq_i$ denotes the number of times $Source\_Bk_i$ is assigned to $Source\_Bk$ as specified in the tag cloud of $Source\_Bk$. We normalize $Resem(Source\_Bk,\ CB)$, so that the computed degree of resemblance is in the [0, 1] range, by dividing the accumulated correlation factors by the sum of the frequencies of occurrence of each tag assigned to $Source\_Bk$.

The Min function in Equation 1 imposes a constraint on adding the correlation factors of tags representing $Source\_Bk$ and CB which ensures that if CB contains a dominant tag T in its tag cloud, i.e., T is highly similar to a few tags in the tag cloud of $Source\_Bk$, T alone cannot significantly impact the resemblance value of $Source\_Bk$ and CB, i.e., "one" does not represent "all".

2) Book Ratings: Among the books in the personal catalog of a LibraryThing user U, U might like some books more than others, which is natural. In recommending books, $PReF$ considers the *rating* assigned to a book CB in $Candidate\_Set$ by a friend of LT_User, denoted LT_Pal[2], that should reflect the degree to which LT_Pal is interested in CB. $PReF$ suggests to LT_User books given high ratings scores by his/her friends, since these books are treated as more appealing to LT_User than books which are given lower ratings.

$$Rate(CB, LT\_Pal) = \frac{Rating\_CB}{5} \quad (2)$$

where $Rating\_CB$ is the rating score given to CB by LT_Pal, and '5' is the normalization factor, i.e., the highest possible rating score that can be assigned to CB.

Note that not every LibraryThing user assigns a rating to each book in his/her personal catalog. Should LT_Pal not provide a rating for CB, $PReF$ considers the collective opinion of LibraryThing users and computes $Rate(CB,\ LT\_Pal)$ using the *average*, i.e., global, *rating* assigned to CB by LibraryThing users as $Rating\_CB$.

3) Reliability of Friends in Book Recommendations: LT_User's friends might include in their catalogs books on various categories, such as politics, fiction, or science, and it is expected that books in certain categories might be more predominant than others in the personal catalogs of

[2] From now on LT_Pal refers to a friend of LT_User who includes a given book (in $Candidate\_Set$) in his/her catalog.

LT_User's friends. Thus, not all (the books included in the catalogs) of LT_User's friends should be given the same "weight" for book recommendation, since recommendations provided by friends who include in their catalog a significant number of books in the same category as, i.e., similar in content to, $Source\_\ Bk$ are more reliable than recommendations provided by friends less familiar with the category of $Source\_\ Bk$. $PReF$ measures the *degree of reliability* of a friend of LT_User, i.e., LT_Pal, in recommending books that are similar (in content) to $Source\_Bk$ as follows:

$$Rel(Source\_Bk,\ LT\_Pal) = \frac{\sum_{i=1}^{m} Min\left\{\sum_{j=1}^{n} wcf(Source\_Bk_i, LT\_Pal_j), 1\right\}}{m} \quad (3)$$

where m (n, respectively) is the number of distinct tags assigned to $Source\_Bk$ by LT_User (LT_Pal to books in his/her personal catalog, respectively), $Source\_Bk_i$ (LT_Pal_j, respectively) is a tag assigned by LT_User to $Source\_Bk$ (LT_Pal in describing books in his/her personal catalog, respectively), and $wcf(Source\_Bk_i,\ LT\_Pal_j)$ is the correlation factor in the word-correlation matrix between $Source\_Bk_i$ and LT_Pal_j. In Equation 3, m is the normalization factor that scales the corresponding *degree of reliability* in a [0, 1] range.

4) Recommendations: Having determined (i) the *degree of resemblance* between $Source\_Bk$ and each book CB in $Candidate\_Set$, (ii) the *rate* score assigned to CB by each of LT_User's friends, and (iii) the *degree of reliability* of each friend of LT_User, $PReF$ computes the *ranking score* of CB, denoted $Rank(CB)$, as follows:

$$\text{Rank(CB)} = ArgMax_{LT\_Pal_i \in Pal_{CB}}\{Rel(Source\_Bk, LT\_Pal_i) \times \frac{Resem(Source\_Bk, CB) + Rate(CB, LT\_Pal_i)}{1 – Min\{Resem(Source\_Bk, CB), Rate(CB, LT\_Pal_i)\}}\} \quad (4)$$

where Pal_{CB} is the group of LT_USer's friends who include CB in their personal catalogs, and LT_Pal_i is the i^{th} LT_Pal in Pal_{CB}. The *ArgMax* function in Equation 4 ensures that the highest ranking score of CB, among the ones computed for each of LT_User's friends, is considered during the recommendation process, which guarantees that no duplicate books are recommended to LT_User. By combining the *resemblance* and *rate* scores using the *Stanford Certainty Factor* (SCF), $PReF$ measures the relative *appealing* value of CB (in $Candidate\_Set$), which is high only when both the resemblance and rate scores are high, since SCF is monotonically increasing (decreasing) function. Furthermore, by employing the Joint Product in Equation 4, $PReF$ *adjusts* the computed appealing value of CB based on the *reliability* of a friend of LT_User in recommending books for $Source\_Bk$.

The Top-10 *ranked* books are recommended to LT_User, which follows the number of recommendations presented by LibraryThing to its users.

III. Experimental Results

In this section, we first introduce the data and metrics which were used for assessing the performance of $PReF$. Thereafter, we detail the results of the empirical study conducted for evaluating $PReF$.

A. Experimental Data

To analyze the performance of $PReF$, 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 and ratings of) books in their personal catalogs, (ii) lists of their friends, and (iii) (tags and ratings of) books posted under their friends' personal catalogs. In addition, the extracted data include the tag cloud and the global rating score of each book listed in (i) and (iii) above. To the best of our knowledge, there is no existing dataset for assessing the performance of personalized book recommenders, and thus we rely on independent appraisers who manually examined the relatedness of each one of the top-10 recommendations generated by $PReF$ with respect to each book in their personal catalogs, yielding a set of 100 books, denoted $Test\_Books$, used in our empirical study.

B. Evaluation Metrics

To evaluate the effectiveness of $PReF$ in generating personalized book recommendations, we apply two well-known information retrieval metrics, the (overall) *Precision at K* (*K*= 1, 5, and 10) and *Mean Reciprocal Rank* [3]. While the $P@10$ measures the overall user's satisfaction of the recommendations created by $PReF$, $P@K$ and MRR evaluate the ranking strategy of $PReF$, since the higher related recommendations are ranked, the higher their corresponding $P@K$ and MRR scores should be.

To further assess the efficiency of our personalized book recommender, we employ the *imp* metric [6], which is a widely-used evaluation method that measures the level of improvement of a personalized approach when compared to a baseline, i.e., non-personalized, approach in ranking relevant recommended resources, i.e., books in our case. The overall *ranking improvement* of a personalization recommender is calculated by averaging the improvement for all the books in $Test\_Books$. The $higher$ the imp score is, the $better$ the ranking strategy adopted by a recommender is, i.e., imp shows the effectiveness of a personalization technique at moving "good" (book) recommendations to the top of the list [6].

C. Performance Evaluation

In this section, we present the experimental results achieved by $PReF$ and compare its performance with

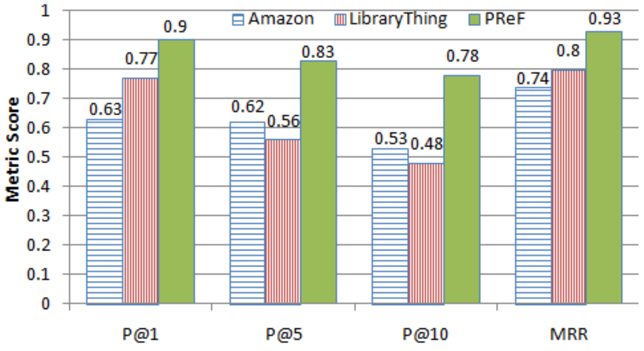

Figure 1. $P@K$ and MRR scores achieved by $PReF$, Amazon, and LibraryThing for the books in $Test\_Books$

the recommendation systems of Amazon and LibraryThing introduced in [5] and [1], respectively.

1) Assessment: To assess the overall performance of $PReF$, Amazon, and LibraryThing we have computed the $P@K$ scores on the top-10 book recommendations generated by $PReF$, Amazon, and LibraryThing, respectively for each book B in $Test\_Books$, based on the books labeled as (not) $related$ to B by each independent appraiser. As shown in Figure 1, the $P@1$ score of $PReF$, which is 0.90, indicates that among the 90 out of 100 books in $Test\_Books$, their first recommended books generated by $PReF$, i.e., the books with the highest ranking score, were treated as $related$. 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 13% lower compared with $PReF$'s $P@1$ score. Figure 1 shows that the $P@5$ score of $PReF$ is at least 21% higher than the $P@5$ scores of Amazon and LibraryThing. Also shown in Figure 1, the $P@10$ scores of $PReF$, Amazon, and LibraryThing, which are 0.78, 0.53, and 0.48, respectively.

Besides the $P@K$ scores, we have also assessed the performance of $PReF$ (Amazon and LibraryThing, respectively) based on the MRR metric. As shown in Figure 1, the MRR scores computed for $PReF$, Amazon, and LibraryThing are 0.93, 0.74, and 0.80, respectively, which reflect that while on the average users of $PReF$ are required to browse through the top ($\cong \frac{1}{0.93}$ = 1.07) 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.

Lastly, we have computed the imp score of $PReF$ over Amazon and LibraryThing. $PReF$ achieves a 19% (13%, respectively) improvement over Amazon (LibraryThing, respectively) in generating books recommendations relevant, i.e., appealing, to users.

IV. Conclusions

It is an unpleasant experience for book enthusiasts to acquire books and later discover that the books do not appeal to their "tastes". In addition, it is difficult for book enthusiasts to keep track of new books published on a regular basis due to their number. Existing book recommenders, such as the one employed by LibraryThing, aid users in identifying books of interests. These recommenders, however, present the same recommendations to users that share the same profile information or common interests and hence are inadequate, since the suggestions do not often meet individual users' preferences. To address this problem, we have developed a personalized book recommender, called $PReF$. $PReF$ relies on (i) online *connections*, i.e., friendships, established among users at a social website, (ii) the existence of user-generated *tags* and *ratings*, and (iii) word-correlation factors, i.e., word-similarity measures, to generate book recommendations $tailored$ to the interests of an individual user. Unlike recommenders that rely on the "wisdom of crowds" to make recommendations, $PReF$ considers only interests shared among a user U and members of U's "inner circle", which yields valuable recommendations for U. In addition, $PReF$ is not limited by an exact match constraint and thus identifies books similar in contents, even if they do not share any common tags, which enriches the set of candidate books to be recommended. We have conducted an empirical study and verified that (i) relying on data extracted from social websites improves the effectiveness of book recommenders and (ii) $PReF$ outperforms the recommenders employed by Amazon and LibraryThing.

While $PReF$ is currently designed for recommending books, we intend to extend $PReF$ so that it can recommend items in various domains, such as songs and movies, provided that data describing items of interest and friendships among users are available on one or more social websites.

References

[1] LibraryThing, http://www.librarything.com/wiki/index.php/Automatic_recommendations.

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

[3] W. Croft, D. Metzler, and T. Strohman, *Search Engines: Information Retrieval in Practice*, Addison Wesley, 2010.

[4] J. Koberstein and Y.-K. Ng, *Using Word Clusters to Detect Similar Web Documents*, KSEM, pp. 215-228, 2006.

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

[6] A. Shepitsen, J. Gemmel, B. Mobasher, and R. Burke, *Personalized Recommendations in Social Tagging Systems Using Hierarchical Clustering*, in RecSys, pp. 259-266, 2008.