ABSTRACT

In this work we present a prototype that calculates similarity between authors and books. The application allows the user to choose the author and the system retrieves books from other authors that are similar to the chosen author’s work.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Digital Libraries; H.3.7 [Digital Libraries]: System issues

General Terms
Algorithms, Recommendation

Keywords
Author recommendation, Book recommendation, Content-book recommendation

1. INTRODUCTION

Literary reading is an important and rewarding activity for people who engage in it. Very often readers choose books by their author. Readers that have a good experience reading a book from an author, most likely will try to read more books from the same author. They like to connect and know the author through her work.

Readers choose books by its author because they like the subject. This is the case, for example, of collections, where several books with same characters, but different stories, are available, e.g., “Harry Potter” by J. K. Rowling or “The Famous Five” by Enid Blyton. Readers may also choose books from the same author, because they like her writing style.

When readers choose books from the same author, because they had a good experience reading books from this author, they develop certain expectations. If these expectations are not met, readers often became disappointed with the choice made.

To choose new books, readers typically go to libraries or bookstores, both real and virtual, and spend considerable amounts of time browsing shelves in search of something to read next. Both libraries and bookstores, even small ones, have enough books to make the selection time-consuming. In this scenario, book recommendation systems [4, 9, 10] are useful to help readers find what they want.

Two main techniques are used to develop recommendation systems [1]. Content-based techniques [5] in which users will be recommended items similar to those the user liked in the past; and collaborative filtering techniques [8, 6] in which users will be recommended items that were preferred together.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 2013 CID 978-2-905450-09-8."
2. THE BOOK RECOMMENDER

The book recommender prototype is a web-based application. Figure 1 shows our prototype. The prototype presents a list of authors to the user. The user chooses an author and the type of similarity: subject, style, or subject and style. When the user clicks on the “Search button”, the recommender searches for books that match the criteria and presents a top of the 20 most similar books to the user.

3. OUR ALGORITHM

The algorithm implemented in this work is based on the kNN item-based collaborative filtering algorithm proposed by [6]. Our prototype first calculates a book \times book similarity matrix, where \text{cell}_{i,j} contains the similarity between book\textsubscript{i} and book\textsubscript{j}. Similarity is calculated using the Pearson correlation in Equation 1, where \text{B}_1 and \text{B}_2 are the vectors representing books.

\[
\text{sim}(\text{B}_1, \text{B}_2) = \frac{\sum_{i=1}^{n} (\text{B}_{1i} - \overline{\text{B}}_1)(\text{B}_{2i} - \overline{\text{B}}_2)}{\sqrt{\sum_{i=1}^{n} (\text{B}_{1i} - \overline{\text{B}}_1)^2} \sqrt{\sum_{i=1}^{n} (\text{B}_{2i} - \overline{\text{B}}_2)^2}}
\]  

Authors are represented as classes of books that they have written. For each book that was not written by the author, the algorithm ranks the book calculating how close the book is to the author using Equation 2, where \text{a} is the author for whom we are calculating the top-20; \text{b}_i is the book not written by author \text{a} that we are ranking; \text{A} is author \text{a} book set; and \text{b}_{a,j} is a book written by author \text{a}.

\[
\text{rank}_{a,b_i} = 20 \times \frac{\sum_{b_{a,j} \in A} \text{sim}(b_i, b_{a,j})}{||A||} \land b_i \notin A
\]  

After calculating the rank for all books not written by the author, the algorithm returns the top-20 ranked books for which the rank is greater than 1.

Books are represented as feature vectors. The similarity in subject is calculated using vectors of content-word frequencies and the similarity in style is calculated using vectors of stylistic features [7]. For this work we used vectors of stop-word frequencies. The final rank is calculated using the proper similarity matrix.

To find books both similar in subject and style, first, ranks are calculated separately and, then, the algorithm averages both ranks, as shown in Equation 3.

\[
\text{rank}_{b_i} = \frac{\text{rankInSubject}(b_i) + \text{rankInStyle}(b_i)}{2}
\]  

4. CONCLUSION & FUTURE WORK

In this work, we have presented a content-based book recommendation prototype that searches for books similar to books written by a given author. The prototype represents books as feature vectors and uses content-word frequencies to retrieve books with similar subjects. To retrieve books with similar writing styles, the prototype uses stylistic features. Stylistic features present a promising ground in the book recommendation field, because they are subject-independent and allow the retrieval of books that are more similar to the type of books the user likes to read. This is an important feature in literary book recommendation. This work is part of our ongoing study of the use of these features in book recommendation.

5. REFERENCES