# A Profile- and Community-Driven Book Recommender System

I. Petrović, P. Perković and I. Štajduhar
University of Rijeka, Faculty of Engineering, Department of Computer Engineering, Rijeka, Croatia
iva.petrovic5@gmail.com

**Abstract - Book lovers often struggle to find something new to read. The choice of possible books to read can be overwhelming and confusing, making this subjective decision-making process quite difficult. Searching the Internet we often stumble upon opinions and book ratings of people who are strangers to us. We do not know what their favorite categories, authors, publishers and taste in books are, nor whether we should trust them. As a possible solution to the above mentioned problems, we present *Pickbooks* - a Web application that is partly a social network and partly a book database. It integrates the social aspects of today's popular social networks into a book recommender system, and not the other way around. It features book recommendation based on previously read books or personal preferences. It allows users to message and follow people they like, see their updates, reviews and get recommendations from them, and vice versa. The users are encouraged to rate a book they have read before, as those ratings affect the global lists of top rated and currently popular books. There are other features which are in the works and will allow us to further improve the presented application and attract more audience.**

## I. INTRODUCTION

Many book fans often find themselves in a situation where they do not know what to read next. Recommender systems are there to help them in these situations. Consider, for example, the *Internet Movie Database* (IMDb) website (http://www.imdb.com/). It is, most certainly, today's most popular movie database website. What if we could create something similar to that, but for book recommendations, i.e. a website that would incorporate those features and adapt them to books? Motivated and inspired by this idea, *Pickbooks* emerged.

A lot of websites give their users lists of top rated books, but that is no guarantee the users will like them. The main reason for this is the user's possible (and quite probable) disagreement with the interests and tastes of that author's list. One viable solution to this problem is to develop a recommender system embedded with some social network characteristics, enabling us to connect people with mutual interests. Broadening this vision of the recommender system by using these features provides the potential to bring books closer to the user.

In short, *Pickbooks* is a Web application, which fuses a book database, a social network and a book recommender system, into one product, available to users free of charge. Imagine binding a *Twitter*-like application (https://twitter.com/) [1] with a book recommender system, and you would understand the proposed idea.

The paper is structured in the following way. In the following section several popular and comparable related recommender systems are presented and the advantages of the proposed system over its direct competitor, *Goodreads*, are described. In section III, several important approaches to recommender systems are explained, introducing additionally some of the more popular consumer services as their respective representatives. The details concerning the proposed application and its' recommender system are outlined in section IV. In section V, final remarks are given together with an insight into the more probable future development plans for *Pickbooks*.

## II. RELATED WORK

Browsing the Internet, one can find a lot of different types of recommender systems. Some of the most popular recommender systems, comparable to *Pickbooks* in several ways, are:

- *Last.fm* (http://www.last.fm/) is a music website that uses a recommender system called *Audioscrobbler*. Its goal is to create a detailed user profile based on the user's musical tastes. In this case, the profile is based on recording details of the tracks the user listens to.
- *Netflix* (*https://www.netflix.com/*) provides users with the ability to stream online videos on demand. Today, *Netflix* uses one of the best available recommendation algorithms. An extensive personalized video-recommendation system is based on ratings and reviews by its users.
- *Foursquare* (https://foursquare.com/) is a local search, recommendation system that learns what you like and leads you to places you love. Their recommendation system records places visited by the user, the things they like (reported to the application), and other users, whose advice they trust.

The recommendation system introduced in this paper is very similar to the recommender systems mentioned above. All three systems create a detailed profile based on a user's tastes. In the case of *Pickbooks*, this applies to book tastes, e.g. what book categories the user likes to

The work presented in this paper is partially supported by the University of Rijeka research grant 13.09.2.2.16.

read or what authors he or she prefers. Just like *Netflix*, the introduced recommender system is based on user's book ratings.

*Pickbooks* uses the best features of the most famous recommender system algorithms, combining them into one system that strives to give its users best book recommendations. What is new in this recommendation system? The goal is not only to collect data about users. It is also to provide users with the opportunity to give recommendations to their friends.

The main competitor of *Pickbooks* is *Goodreads* (*http://www.goodreads.com/*). It uses a recommendation algorithm to analyze which books people might like, based on books they have enjoyed in the past and books that people with similar tastes have liked. In order to get a book recommendation from this system, the user has to assign a rating to 20 books, at least.

For the authors of this paper, contemplating this from the perspective of *Goodreads* users, it was a big disadvantage. Sometimes rating books can be exhausting, especially if one hasn't read the books recently and cannot remember which 20 books to rate. It would be better to enable users to get a book recommendation immediately after registering. The proposed recommender system gives a book recommendation at the exact same moment the user starts searching for books or authors. Users are also given a list of top rated and trending books.

As was already pointed out, the idea was not only to create a recommender system, but also to interweave it with social network aspects. *Goodreads* implements a similar idea, i.e. it also enables their users to find friends with similar interests and to engage in conversations with them. *Pickbooks* also consists of a friend recommendation algorithm that monitors users' activities, what books they have read and how they have rated them, and based on that information it recommends cliques of users sharing the same interests. Another factor for the algorithm are the users' lists of friends. Using a user's list of friends, friends of his or hers friends are recommended. Most of social networks work on that principle, e.g. *Facebook* (https://www.facebook.com/) [2].

There are some features *Goodreads* is missing, which are implemented in *Pickbooks*:

- Generic representation of the system's data flow;
- Book recommendation when searching for book/author (e.g. based on book information the user is searching for, the system recommends similar books from the same category or books from the same author);
- Friend recommendations based on same interests or on mutual friends;
- Search for friends based on their name, e-mail address, city they come from or mutual friends;
- Easy way to recommend books to friends;
- Posting statuses about the books the user read/likes;
- Recommendation of top rated books from all categories.

## III. RECOMMENDER SYSTEM

According to [3], recommender systems represent a powerful method for enabling users to filter through large information and product spaces, and, thus, pose an important part of the information and e-commerce ecosystem. The goal of a recommender system is to generate meaningful recommendations to a collection of users, for items or products that might interest them. There are several popular approaches to creating a recommendation algorithm: collaborative filtering, content-based filtering and a hybrid approach. They are described next.

### A. *Collaborative filtering*

Collaborative filtering (CF) is a popular recommendation algorithm that bases its predictions and recommendations on the ratings or behavior of other users in the system. This type of approach collects and analyses a large amount of information on users' behavior, activities and personal preferences, and thus predicts what the users will like, based on their similarities to other users using the same service. The key advantage is that it does not rely on machine analyzable content and, as a result, it can generate recommendations without requiring an understanding of the item being recommended [4,5].

CF operates on the assumption that people will like similar things, which they liked in the past, and so can be slow to adapt at sudden changes in user preferences and other related information. The quality of recommendations are solely based on how much information a user has shared with the system, thus more information means better recommendations. Often, the algorithms operate on a user database that counts millions of entries and, as such, needs a large amount of computational power to calculate recommendations [6].

*Last.fm*, *Facebook*, *MySpace*, *LinkedIn* (https://www.linkedin.com/) and other social network services use collaborative filtering to recommend music, new friends, social connections, etc.

### B. *Content-based filtering*

The second popular approach to creating a recommendation system is content-based filtering (CBF). In CBF, all items have a description and all users have a profile of their preferences (user history, ratings, personal information, etc.). The algorithm then matches the keywords from these two sources and generates a list of recommendations. The algorithm sorts and recommends items, based on their weight of importance.

One of the key challenges with this type of approach is concerned with enabling the system to use collected data from one source and apply the filtering to other content sources. For example, if a user likes a book category by the name of "Romance", it would be useful to be able to recommend movies, videos and news articles that contain that exact keyword [6].

CBF filtering techniques suffer from problems like limited content analysis, overspecialization and new users. Limited content analysis refers to the problem when a set of attributes, attached to every item in the collection, is rather small. No suitable suggestions can exist if the analyzed content does not contain enough information to differentiate items the user likes from items the user does not like. Overspecialization occurs when the system judges individual items based on a limited number of indicators. This results in the system recommending items that are either too alike to what the user has seen in the past or items that are content-identical to other recommendations shown at the same time. New users present a problem, because the recommendation algorithm has yet to receive profile information from the user to be able to create a recommendation based on criteria [7].

Examples of services that use this approach are: *Pandora Radio* (http://www.pandora.com/), *IMDB*, *8Tracks* (http://8tracks.com/), *Foursquare*, etc.

### C. *Hybrid recommender system*

Hybrid recommender system (HRS), as the name clearly suggests, is a hybrid of the two aforementioned approaches to recommendation algorithms [8]. In recent history, researchers speculated that this kind of approach, combining collaborative and content-based filtering, may yield better and more accurate results, at least in some cases. Using the hybrid method, problems inherent to pure implementations (CF, CBF), like cold start and sparsity, can be overcome [9,10].

*Netflix* is one of the most popular services, which combines both collaborative and content-based filtering for recommending movies. Table I depicts the links between several popular Web services and their respective recommender system types.

## IV. APPLICATION DESCRIPTION

*Pickbooks* is a Web application that allows people to discover new books, and share their thoughts about the ones they have already read. The application is powered by our own recommendation algorithm, which uses the hybrid approach for generating recommendations. Book recommendations are generated by leveraging user's personal preferences and connections with their friends. Previous searches, book ratings, the same book category and author preference with friends and friends of friends, book popularity - all are taken into account when learning about a user's reading preferences and habits.

TABLE I. RECOMMENDER SYSTEM TYPE USED BY A PARTICULAR WEB SERVICE

| Web service name | CF | CBF | HRS |
|---|---|---|---|
| Last.fm | X | | |
| Netflix | | | X |
| 8Tracks | | X | |
| IMDB | | X | |
| Facebook | X | | |
| Foursquare | | X | |
| Pandora Radio | | X | |
| MySpace | X | | |
| LinkedIn | X | | |

Upon arriving on the landing page (Fig. 1), before starting using our recommender application, it is mandatory to give some information about oneself. The registration process requires the user to choose a unique username, password and enter an email address for further authentication. After a successful registration, the user is immediately able to search through the system's database, connect with his or her friends, meet other book lovers, rate his or her favorite books, keep a list of what to read next, etc. The system's topology is illustrated in Fig. 2. A more thorough analysis of its architecture is described next.

Our system can be roughly divided into five main parts:

- Home page,
- Profile page,
- Library,
- Friends, and
- Chat.

Home page (Fig. 3) is the main page of our Web application, where users are encouraged to search for recommendations for top rated and currently popular

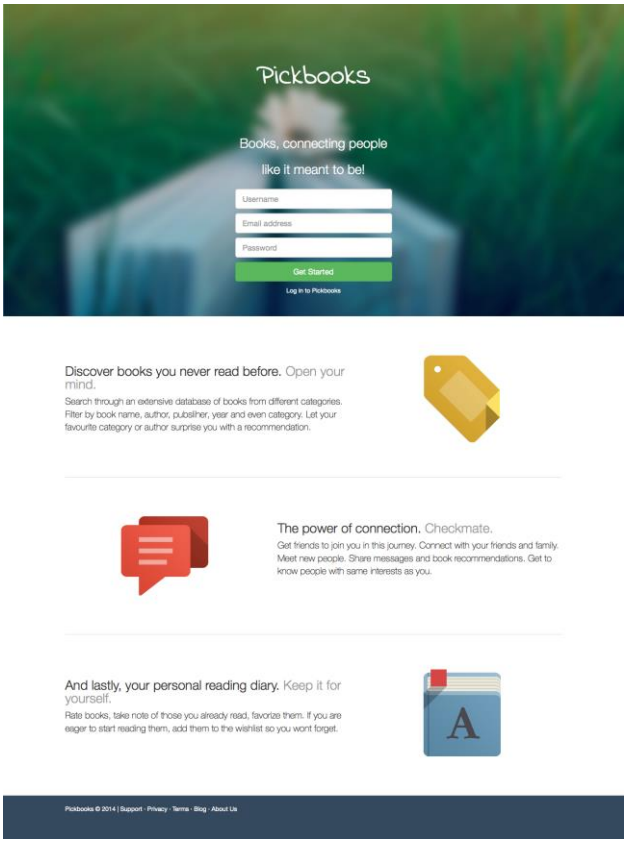


Figure 1. Landing page

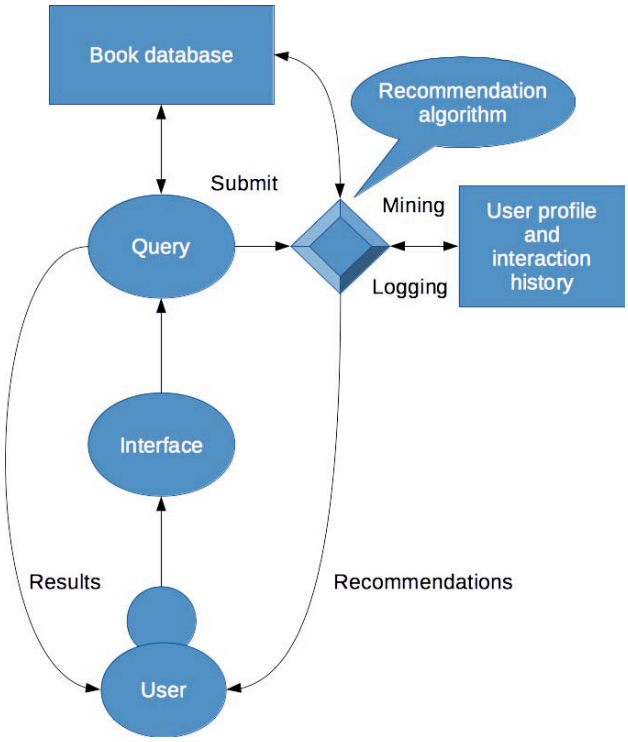


Figure 2. Generic representation of the system's data flow

books, among other things. An important part of this page is the news feed. There, users can find out who added them as a friend, and see what books their friends liked recently. Additionally, users are also able to share their opinions on interesting books they read with their friends through the option of posting statuses. They can also search for other users, sharing the same interests, or friends of friends, add them as their friends or send them messages. Home page combines basic recommendations and social network aspects, and puts them together in one place.

The profile page contains the public profile of the user. A user profile is personally curated by the owner, and contains the user's personal information (profile picture, full name, location, gender, biography and website), books he or she has read, his or her favorite books, books they would like to read next and personal statuses. Additionally, a list of people who are being followed by the user, and a list of people following the same user, are also available. All of the mentioned information can be explicitly hidden from the public, if the user chooses so, and can be adjusted using the security options.

Library page (Fig. 4) basically displays a book database. Searching for new books can sometimes be really exhausting. Our system enables filtering books by their name, author, publisher, publication year or category. To help the user even more in deciding what to read, the system provides a short book description, so he or she can have a quick look at the book content. Whilst reading the short description and basic information about a book, the user is also given a list of books from the same author and/or category.

Additionally, the user can recommend books to his or her friends. They will receive a notification about the recommendation shortly after it has been sent. Most of the book fans have their favorite authors, and want to read all of his or her books. Often they do not know which books they have read, and which they have not read yet. That is why it is useful to have a list containing all the books one has read. Our system allows the creation of lists like that one, and even more. The user is able to make his own wish list, a list of favorite books (once read), and even rate them.

The next part is the friends page. As it was already stated, our idea was to entwine a recommender system with a social network. Therefore, this approach required the introduction of a user's friends. This page allows the user to search for friends, or to find people sharing the same interests. There is an option to search and filter friends by name, email, location or mutual friends. Also, there is a generated list of people the user may know. When users find their friends, they can start following them, gaining the ability to receive information about their activities in the news feed on their home page.

The chat hub is the last main part of the Web application. On this page, the user can exchange messages in real time with his or hers friends, anytime he or she wants. If they are offline in that exact moment, that poses no real problem. They will receive a message when they come back online.

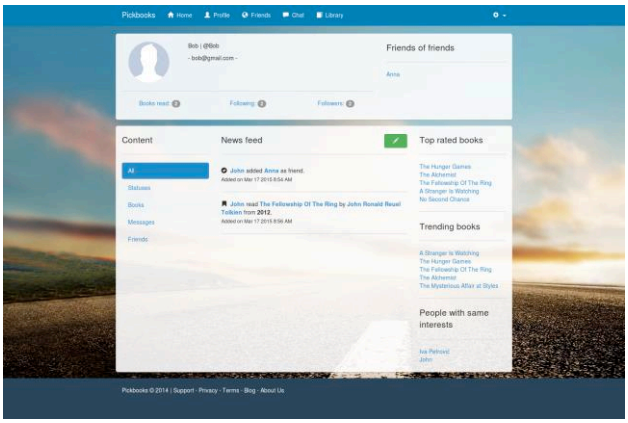


Figure 3. Home page

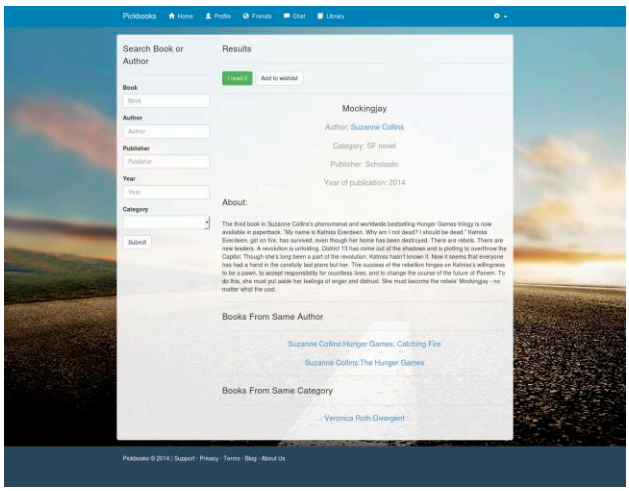


Figure 4. Library page

*Pickbooks* was developed using only free and open source Web technologies. For the frontend design, we used Bootstrap framework [11] and other custom HTML (Hypertext Markup Language), CSS (Cascade Style Sheet) and JS (JavaScript) scripts. The system's backend consists of both PHP (PHP Hypertext Preprocessor) and MySQL scripts, working together to enable users to interact with the application.

## V. CONCLUSION AND FUTURE WORK

In this paper, a Web application for book lovers is presented to discover books and connect with friends. Although the expectations were met and the application has successfully evolved through development, there is still some work that needs to be done.

The idea for this application arose from the disappointment, evoked by the lack of good internet services regarding books. We found that Amazon's *Goodreads* has the biggest market share and that there are no other significant competitors in the market. This is the main reason this application was developed, thus fulfilling our vision regarding the approach to this kind of social experience and making it a reality.

We have a few features in store to improve the overall experience of *Pickbooks*, which will be added at a later date:

- Book lists: Every user will be able to create personal book compilations, like top lists, for every kind of occasion. The lists can then be shared with the public, for others to see, or just with friends. We believe this will allow users to find themselves a focused group of recommendations, created exclusively by other users, ones that will be very hard to reproduce by the algorithms or otherwise.
- Community driven database expansion: We will enable users to submit proposals for adding books to the library. They will be able to provide information about books and their authors. After the submission of data, administrators will confirm or reject books based on information authenticity.
- Administration page: In future we are going to develop an administration page where administrators will have the ability to confirm or reject user's book submissions. Also, another very important role of administrators will be to moderate the activity of the overall community, like deleting inappropriate content.
- Mood centric recommendations: A user's search for a book by categories can result in recommendations that don't necessarily suit his or her mood. We would like to extend our recommendation system, by adding an option of recommending books based on user's mood. This feature will be accessible only on demand. Getting book feedback from other users and analyzing keywords will allow us to get an estimate of a book's psychological effect.
- Online book purchases from retail stores: We would like to offer our users a quick and easy way to find a book and buy it online from different retail stores, available immediately.
- Further security and user privacy improvements: We know that user's personal information are indeed personal, therefore we would like to further improve security and privacy aspects of *Pickbooks*, giving users more control over these settings.
- As always, we strive to improve the accuracy and authenticity of book recommendations. We will take a look at our recommendation algorithms, see where we can improve, and try to fine-tune those areas.

## REFERENCES


[1] A. Java, X. Song, T. Finin and B. Tseng, "Why we twitter: understanding microblogging usage and communities," in Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis, ACM, 2007, pp. 56-65.

[2] N. B. Ellison, C. Steinfield, and C. Lampe, "The benefits of Facebook "friends:" Social capital and college students' use of online social network sites," Journal of Computer-Mediated Communication, vol. 12, no. 4, 2007, pp. 1143-1168.

[3] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative Filtering Recommender Systems," Foundations and Trends in Human-Computer Interaction, vol. 4, no. 2, 2010, pp. 81-173.

[4] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, 1998.

[5] X. Su, and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," Advances in artificial intelligence, vol. 4, 2009.

[6] J. Leskovec, "Recommender Systems: Content-based Systems & Collaborative Filtering", Stanford University.

[7] G. Adomavicius, and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, 2005, pp. 734-749.

[8] J. Basilico, and T. Hofmann, "Unifying collaborative and content-based filtering," in Proceedings of the 21-st international conference on Machine learning, ACM, 2004.

[9] R. Burke, "Hybrid recommender systems: Survey and experiments," User modeling and user-adapted interaction, vol. 12, no. 4, 2002, pp. 331-370.

[10] G. Adomavicius, and A. Tuzhilin, "Context-aware recommender systems," Recommender systems handbook, Springer US, 2011, pp. 217-253.

[11] S. Santo, "Best way to customize Bootstrap CSS style," Bootstrap framework, 2014.