Natural Language Processing for Book Recommender Systems

by

Haifa Alharthi

Thesis submitted in partial fulfillment of the requirements for the

PhD degree in Computer Science

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Abstract

The act of reading has benefits for individuals and societies, yet studies show that reading declines, especially among the young. Recommender systems (RSs) can help stop such decline. There is a lot of research regarding literary books using natural language processing (NLP) methods, but the analysis of textual book content to improve recommendations is relatively rare. We propose content-based recommender systems that extract elements learned from book texts to predict readers' future interests. One factor that influences reading preferences is writing style; we propose a system that recommends books after learning their authors' writing style. To our knowledge, this is the first work that transfers the information learned by an author-identification model to a book RS. Another approach that we propose uses over a hundred lexical, syntactic, stylometric, and fiction-based features that might play a role in generating high-quality book recommendations. Previous book RSs include very few stylometric features; hence, our study is the first to include and analyze a wide variety of textual elements for book recommendations. We evaluated both approaches according to a top-k recommendation scenario. They give better accuracy when compared with state-of-the-art content and collaborative filtering methods. We highlight the significant factors that contributed to the accuracy of the recommendations using a forest of randomized regression trees. We also conducted a qualitative analysis by checking if similar books/authors were annotated similarly by experts.

Our content-based systems suffer from the new user problem, well-known in the field of RSs, that hinders their ability to make accurate recommendations. Therefore, we propose a Topic Model-Based book recommendation component (TMB) that addresses the issue by using the topics learned from a user's shared text on social media, to recognize their interests and map them to related books. To our knowledge, there is no literature regarding book RSs that exploits public social networks other than book-cataloging websites. Using topic modeling techniques, extracting user interests can be automatic and dynamic, without the need to search for predefined concepts. Though TMB is designed to complement other systems, we evaluated it against a traditional book CB. We assessed the top k recommendations made by TMB and CB and found that both retrieved a comparable number of books, even though CB relied on users' rating history, while TMB only required their social profiles.

Acknowledgements

I would like to express my deepest gratitude and thanks to Professor Diana Inkpen, my supervisor, mentor, and role model. Her genuine care, help, and patience brought out the best in me and other students. I am grateful for all of the resources that Professor Inkpen has provided from Tamale seminars and group discussions to the productive and energetic lab working environment. It is an honour working with and learning from such an icon.

Thank you to the Saudi Electronic University (SEU) and the Natural Sciences and Engineering Research Council of Canada (NSERC) for funding my research. Also, I am thankful to NVIDIA Corporation for their generous donation of a Titan X GPU which helped reduce the experimentation time.

Thank you to all of the professors at University of Ottawa who sparked my passion in the field and provided me with the knowledge I needed. I am grateful for all my Ph.D. committee members, Professor Anna Feldman, Professor Chris Tanasescu, Professor Stan Szpakowicz, and Professor Tony White for their insightful comments and feedback. Special thanks to Professor Szpakowicz for his constructive criticism and challenging questions and to Professor Tanasescu for his valuable suggestions.

Thank you to Dr. Kenton White and Dr. Jelber Shirabad for their time and feedback. Thank you to all of the Tamale seminar speakers who motivated us to explore new methods. Thank you to the researchers, especially Dr. Paula Cristina Vaz and Dr. Julian Brooke, who shared their collected datasets and developed tools that facilitated my work. Thank you to my colleagues, especially Zunaira Jamil, Ruba Skaik, Prasadith Buddhitha, SHY, Ehsan Amjadian, and Doaa Ibrahim, for being there when I needed help. Thank you to the city's baristas who brought colour and joy to my day-to-day research life.

I thank my mom and all my family members who have endlessly supported and believed in me. Special thanks to Khalid who has always been there through the ups and downs and to

whom I owe this achievement.

None of this could have been achieved without the help of God to whom I am most grateful for inspiring me, putting amazing people along my way, and helping me through hardships.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

With the current state of information overload, Internet users can often find it difficult to choose from the multitude of available products and services, and this has generated a demand for recommender systems (RSs) that can provide personalized suggestions. The idea behind RSs is not new; just as it is common to ask acquaintances for recommendations when choosing restaurants, movies, books and such, recommender systems predict how likely the target user will be interested in a particular item, even if they are unfamiliar with it.

To make recommendations, RSs typically need items (i.e., recommended objects), users and user feedback about the items. Users who receive recommendations interact with the system, and their interactions are stored in a database to be used for future recommendations. User opinion or feedback is logged and stored whether it is explicit or implicit (Shani and Gunawardana, 2011; Ricci et al., 2011); users' provide explicit feedback in the form of ratings (*e.g.*, 1 to 5), and implicit feedback is determined from user behaviour (*e.g.*, reading an article). For simplicity, in this thesis, we refer to user feedback as ratings. To make suggestions, RSs exploit users' rating history, social-media content, relationships, personality and emotions, as well as product features.

There are two main types of recommender systems: collaborative filtering (CF) and content-based (CB). CF requires a rating matrix of all other users in the system in order to predict a target user's reading preferences, while CB relies solely on the target user ratings to find patterns in their previous preferences. CB avoids many issues that impact CF, as detailed in chapter 2.2. Our proposed approach is a CB that makes personalized book recommendations after exploiting the textual content of relevant books. Personalized RSs make suggestions specific to each user, rather than recommending items for groups of users (*e.g.*, those clustered according to demographics). To deal with new users, we developed a module that complements the main system, learns users' interests from social media, and makes appropriate personalized book recommendations.

1.2 Motivation

1.2.1 The importance of reading and book recommender systems

The deployment of RSs in e-commerce has advantages for both sellers and consumers. Sellers' goals are to make their products accessible to interested clients, achieve consumer satisfaction, and gain loyalty. These objectives can be met if users continually receive products that meet their needs. Sellers can also make profit using automatic RSs, since there is no need for additional overhead (*e.g.*, employees). In addition, consumers receive a list of products most likely be useful to them, therefore saving the time, effort, and resources required to find items they truly appreciate.

The benefits of recommendations can surpass e-commerce scenarios. In this work, we focus on book recommendations that could be useful for libraries, schools, and e-learning portals. The proliferation of e-books gives readers access to vast and inexpensive resources with little effort. Because of this, one might assume that book-reading would become more widespread. However, the numbers actually prove the opposite; the practice of reading for pleasure is declining, particularly among the young people[1].

A variety of studies have found that reading is beneficial. A comparison of the well-being of 7,500 adult Canadian readers and non-readers determined that the former are significantly more likely to report better physical and mental health, to do volunteer work, and to feel satisfied with life(Hill, 2013). Such statistics support the assertion by the National Institute of Child Health and Human Development (NICHD) that ‘Reading is the single most important skill necessary for a happy, productive and successful life’[2]. Moreover, reading fiction has been found to stimulate profound social communication (Mar and Oatley, 2008), and is correlated with greater ability for empathy and social support (Mar et al., 2009). It also has lingering biological effects, particularly with respect to the connectivity of the brain (Berns et al., 2013). Thus, employing artificial intelligence techniques to spark interest in reading by recommending appropriate types of books is a worthwhile endeavor.

In 2010, Google estimated that there are 129,864,880 books in the world[3] and the count goes up. Approximately one million new and revised titles were published in the UK, China, and the USA in 2013 alone[4]. It is overwhelming for library staff to go through this massive numbers of books, comprehend them, then suggest them to individuals, and it has become evident that automation is required to manage the functions of personalized book recommendations. Moreover, people tend to trust RSs; one study Chen (2008) found that consumers were

[1] https://tinyurl.com/jgunwfx

[2] https://www.ksl.com/?sid=15431484

[3] https://tinyurl.com/ydz4w8jt

[4] https://tinyurl.com/yd9a6f9t

more interested in books labeled ‘customers who bought this book also bought’, than books marked ‘recommended by the bookstore staff’.

1.2.2 The use of book textual content

In this thesis, we propose a book RS that analyzes book text and determines the user reading preferences. The following reasons drive this approach.

The exploitation of book texts in RSs is limited

In chapter 2, we summarized approximately thirty research papers dedicated to book RSs and found that the majority applied book metadata (*e.g.*, author, genre) to build content-based RSs; only a few took the actual text of the books into account. Vaz et al. (2012a) represented books using topics learned by Latent Dirichlet Allocation (LDA), and the style features of vocabulary richness, document length, part-of-speech bigrams, and the most frequent words in a book. Zhang and Chow (2015) represented authors as a four-layer tree containing author information and their book texts divided into pages and paragraphs. Garrido et al. (2014) relied on a book’s text to predict its social tags, which were then deployed to generate recommendations.

Though using book texts can raise copyrights issues, there are some initiatives that encourage RS researchers to follow this direction. The well-known retrieval system Google Books, which searches the full text of books for terms that appear in a given query, receives books from their authors and publishers through Google Books Partner Program[5]. Another project is the HathiTrust Digital Library[6] which allows searching the texts of 16 million book volumes either in the public domain (six million) or copyrighted works (ten million). Commercial recommender systems that perform natural language processing on the text of books are now

[5] https://www.google.com/googlebooks/partners/

[6] `https://www.hathitrust.org`

appearing. For example, BookLamp, which was acquired by Apple in 2014,[7] is described by its previous CEO as a *Book Genome Project* that, upon receipt of a digitized book text from its publisher, "measures the *DNA* of each scene, looking for 132 different thematic ingredients, and another 2,000 variables"[8]. In 2014, the number of books indexed by BookLamp each week was 40000-100000[9].

The proliferation of e-book readers such as Kindle and Kobo has contributed to the most severe decline in regular book sales since the printing press was developed. Amazon declared that sales of e-books surpassed regular books in 2012 (Zhang and Chow, 2015). Since the vast majority of books are now available in electronic versions, RS researchers can make use of these resources without converting hardcopies to digital texts. E-book promotions are already provided through Wifi to Kindle, Kobo, and other e-readers (Zhang and Chow, 2015). Our proposed system could also be deployed by e-book providers, and help current online stores boost their deployed systems.

Advances in natural language processing encourage the exploitation of book texts

The use of NLP to analyze literary books is already an active area and has been applied to profiling characters (Kokkinakis and Malm, 2011) and their personalities (Flekova and Gurevych, 2015), automatic genre identification (Ardanuy and Sporleder, 2016) and extraction of social networks of characters (Elson et al., 2010). Recent conferences and workshops have been dedicated to the application of computational linguistics for literature, including ICCLL[10] and LaTeCH-CLfL[11]. Though a great deal of work has devoted to literary books, analysis of the

[7] http://www.businessinsider.com/apple-buys-booklamp-2014-7

[8] https://tinyurl.com/3qvq8zj

[9] https://tinyurl.com/ycpwnvze

[10] http://www.iccll.org/

[11] https://tinyurl.com/yaquubks

textual content to improve book recommendations is still relatively rare. In fields such as video and music recommendations (Shao et al., 2009; Deldjoo et al., 2016), it is common to rely on the full content of items, such as visual features (*e.g.*, lighting, color) and acoustics, instead of the metadata. This has motivated us to analyse the actual text of the books in order to model user interests.

Books have features that distinguish them from other textual items

Current RSs have applied NLP techniques to analyze textual items, including books (Vaz et al., 2012a), news articles (Kompan and Bieliková, 2010), microblogs (Gurini et al., 2013), movie plots (Bergamaschi and Po, 2015), and scientific papers (Wang and Blei, 2011). Each of these domains has unique characteristics that distinguish it from the others. For example, news have a short lifetime that can become irrelevant within days or even hours, plus there are always new articles which have not yet been rated by users. The content is dynamic, and changes are continuous. When dealing with scientific papers the timeframe is critical in a different way; recent papers are more relevant but older works are fundamental to the learning of the basics of the research field. In addition, papers must meet the relatively narrow interests of the targeted researcher. Microblogs are short texts, typically written in informal language, that can track interactions among users (*e.g.*, replies), and could contain hashtags that summarize their main topics. Movie plots are conversational, and complement visual and sound effects.

The nature of items must be taken into consideration when making recommendations. Unlike other text-based content, many books are hundreds of years old and are still widely read and recommended. Books include aspects that can be unrelated to present times and actions, and specific elements of books can influence a user's reading preference. Recommendations by reader advisory services that recommend literary books in libraries, rely on the following seven appeal factors:

- *Pacing* represents how fast or slow reading a book feels (*e.g.*, intense, leisurely);
- *Characterization* describes the nature and number of fictional characters (*e.g.*, well-defined characters, emphasis on one vs. several characters);
- *Storyline* reflects the plot, theme and genre (*e.g.*, complex, character-driven);
- *Style* represents the language in a book, such as dry, poetic or conversational, and how readers perceive it,. (Readers who tend to appreciate writing style consider it the most appealing factor);
- *Setting* identifies where and when the story takes place (*e.g.*, descriptive, a historical period);
- *Tone/Mood* is the feelings a book stimulates in the reader (*e.g.*, upbeat, creepy); and,
- *Frame* is reader impressions of a book, which comprises settings, atmosphere and tone (Smith et al., 2016).

Researchers are encouraged to create book recommendations that stem from features specific to books. To gather information about books' appeal factors, librarians can subscribe to fee-based reader-advisory databases such as NoveList[12], that are established by professionals (Pera and Ng, 2014b). It is helpful to exploit book texts directly, rather than rely on labels (*e.g.*, genre) tagged by experts. The list of available labeled books is far from complete, due to the difficulty of manually processing the vast number of existing books. For example, in some tests, we did not find information about specific authors/books in NoveList. Also, shallow categorizing of books by just labeling literary works as fiction is common, and when books

[12] https://www.ebscohost.com/novelist/our-products/novelist-plus

are classified under multiple sub-genres, there is no indication of the degree to which a book belongs to a particular genre.

In this thesis, two CB approaches are proposed. One represents books as features that touch on many appeal factors, while the other recommends books after learning their authors' style. It is natural to think that an author's writing style is a factor when it comes to book recommendations. Style is defined by the Oxford English Dictionary as "The manner of expression, characteristic of a particular writer (hence of an orator) or of a literary group or period; a writer's mode of expression considered in regard to clearness, effectiveness, beauty, and the like" (Jeremy, 2000). Stylometry, the computational analysis of writing styles, emerged in the late nineties (Stamatatos, 2009) and it is usually applied in authorship identification tasks that recognize the author of a given text. We are motivated to present books according to their authors' writing styles, because the use of stylometry features for literary book recommendations is promising, as Vaz et al. (2012a) suggests. In addition, the study of authorship attribution is an active and well-known research area. We use deep neural networks (DNN) (see section 2.6) that have been shown to enhance the performance of many NLP applications, including authorship attribution (Solorio et al., 2017; Qian et al., 2017). By applying transfer learning, modeling books according to style becomes straightforward.

1.2.3 User cold start

One challenge facing RSs is user cold start, which occurs when new users with no rating history are introduced to the system, making it difficult to generate personalized recommendations for them. A commercial portal that provides inaccurate first recommendations to new users will risk losing them. This issue has been widely investigated in RSs. One method is to ask new users to rate items at registration, until sufficient ratings are received (Rashid et al., 2002; Kohrs and Mérialdo, 2001). As the signup process can be lengthy and consume much of users' time and effort, many methods were proposed to reduce users' effort, including (Rashid et al., 2008).

Another method is to make recommendations based on personal information about users, such as demographics (Safoury and Salah, 2013) and personality traits (Fernández-Tobías et al., 2016). Social media is a helpful resource to 'warm up' a user's cold start, as users voluntarily share textual and visual content on these platforms; they also create social networks of friends and followers. User connections on social network were used in (Castillejo et al., 2012), (Sedhain et al., 2014), (Guy et al., 2010), (Ben-Shimon et al., 2007) and (Mican et al., 2012). In addition to using social media friends' lists, (Sedhain et al., 2014) analyzed user demographics and the pages a user liked.

To our knowledge, there are no book RSs that exploit social networks other than book-cataloging websites. The topics learned from user shared text on social media can help recognize users' interests and connect them to relevant books. Thanks to topic modeling techniques, the process of extracting user interests can be automatic and dynamic, without the need to search for occurrences of predefined concepts.

Privacy issues are raised whenever users' personal information (*e.g.*, interests) is involved. However, the data we used has already been shared publicly by users whose consent is required prior to processing their profiles. Moreover, once deployed, our system can give users the ability to remove any topics they consider sensitive.

1.3 Problem Statement

This thesis addresses two recommendation problems. Subsection 1.3.1 describes standalone systems which use textual book content to make recommendations according to the target user's explicit ratings. In case of user-cold start situations, the module in 1.3.2 complements the former methods by suggesting books based on user interests inferred from social media, without the need for the user's rating history.

1.3.1 Book-recommendations based on textual content

Traditional RSs do not take advantage of literary books' texts, the use of which could lead to better understanding of user reading preferences and to high-quality recommendations. We propose two recommendation approaches, both of which are content-based recommender systems (CB) with a classifier or regressor that analyzes the text of the books read by the target user and predicts future interests. We aim to improve the recommendation accuracy of English literary books, which are typically formal and well-written (mostly free of misspellings and grammatical errors). Literary books can be fiction or non-fiction (i.e., memoirs with story elements such as characters and plot (Brown and Krog, 2011)).

Here, we assume the availability of the target user's explicit ratings that might exist, depending on the deployed system. Our system works in a ranking recommendation scenario, which is widely applied in e-commerce platforms and multimedia and content recommendations (*e.g.*, news). Here, the system presents a list of the top items in a sidebar, sorted in descending order according to their relevance to the user. In addition to ranking, other RSs apply prediction and classification. The former is useful in situations such as movie rental, to help a customer decide about a product by predicting the level of user interest (*e.g.*, four stars), while the latter predicts if the user will like or dislike a particular product (Schröder et al., 2011).

The first CB transfers information learned by an author-identification DNN model inspired by (Solorio et al., 2017), to a book recommendations module. Using the book text as input a convolutional neural network (CNN) predicts the author, and once it achieves good accuracy the penultimate hidden layer, which is the book representation, is extracted. Then, a regressor is trained over book representations associated with the target user ratings to generate a list of ranked books. The CNN learns the aspects that are most important to distinguish the authors from one another. Though facets are latent, we believe they could reflect distinct diction (word choices), topics, themes and more. However, such a network is not expected to capture all the elements of style (*e.g.*, sentence structure).

The second proposed system characterizes books as vectors of calculated lexical, syntactical, character-based, fiction-specific, and style-based measurements. We include a total of 120 features as explained and justified in 4.2.1. Although our goal is not to model books according to the above appeal factors explicitly, many of the considered dimensions superficially relate to them. For example, we calculated the number of fictional characters in a book which may reflect aspects of characterization. Book vectors are input to multiple recommendation modules, including a 'forest' of randomized regression trees whose variable importance values can provide insight into factors that contribute to recommendation accuracy. Identification of significant elements in reading preferences helps writers, publishers, and RS owners understand elements that appeal to individuals and communities.

1.3.2 Topic modeling in book RSs for new users

Book recommender systems can provide new users with quality suggestions, without requiring them to fill lengthy forms regarding their demographic information, personality traits, or previous reading preferences. The availability of user-generated texts on social media provides a chance for RSs to learn information voluntarily shared by users. We propose an automatic personalization module that uses text shared by the target user on Twitter and matches it to book topics. A profile is created for each user to summarize the subjects discussed on their social media account using topic modeling techniques. The topics found in the user and book profiles are represented as word embeddings. The user profiles are then matched with book descriptions and the most similar ones are suggested.

In this module, we use book descriptions available online; however, topics can also be learned from books' full texts. When assessing the system, we found that few users in the dataset are both active on social media and have read books with out-of-copyright text. Furthermore, book descriptions could serve as a sufficient resource that can capture the main topics of interest to users.

Our research investigated using Twitter data to make book recommendations, since it is the most popular microblogging service with more than 313 million active users writing in 40 languages[13]. As Twitter is not exclusive to book lovers, it can help address the issue of new users without reading profiles. Since its establishment, Twitter has been used to survey opinions, report news (more than 85% of Twitter activities are related to news events), raise awareness, create social and political movements, and more. Topics discussed on Twitter are up-to-date and diverse (De Francisci Morales et al., 2012). Hence, it offers a chance to understand the reactions of active users to their surroundings, *e.g.*, the social and political scene.

1.4 Contributions

- We summarize the current literature on book recommender systems and categorize them based on features used to characterize books (published in (Alharthi et al., 2018a)).
- We propose a system that recommends books after learning their authors' style, and to our knowledge this is the first work that applies information learned by an author-identification model to book recommendation. Given its textual content, a book representation is learned by the author-identification classifier, and then fed to a recommendation module. The system has higher recommendation accuracy than many competitive

[13] `https://about.twitter.com/company`

content-based and collaborative filtering systems. When analyzing the effect of text length, we observed a trend: the more text fed to the author-identification model, the more accurate the recommendation. We also conducted a qualitative analysis by checking if similar books/authors were annotated similarly by experts from NoveList.

- We employ a book RS that applies up to 120 linguistic aspects learned from book text, including lexical, syntactic and fiction-specific features (*e.g.*, the number of fictional characters). In the literature of book RSs, we found one RS (Vaz et al., 2012a) that applies multiple stylometric features, two of which are included in our study (document length and vocabulary richness). Also, (Pera and Ng, 2014a,b, 2015) used readability score to filter out books recommended for emergent readers. To our knowledge, our study is the first to include and analyze all these types of features for book recommendation. We propose two recommendation methods and show that they outperform several content-based and collaborative filtering recommenders in the top-k recommendation scenario. We studied the effect of multiple linguistic elements on reading preferences; in particular, we highlight the variables considered important using an ensemble of randomized trees and produce book suggestions based on them. We conducted a further qualitative analysis by studying book descriptions from NoveLis and authors of similar books.

- We propose an automatic personalization module called the Topic Model-Based book recommendation component (TMB), which exploits text shared by a target user on Twitter and makes book recommendations accordingly. TMB automatically represents the dominant topics discussed by a user without searching for predefined concepts, recognizing named entities or developing ontologies; it also improves any deployed system's capability to address the user cold-start issue. We believe this is the first book

RS that uses social media rather than book-cataloguing websites, as well as the first to extract user-discussed subjects from social media and map them to books. As we did not know of any datasets with such information, we collected one that contains users' social media accounts (from Twitter), their reading preferences and book information (from Goodreads[14]). TMB retrieved a comparable number of books as CB in a top-k recommendation scenario, even though CB relied on user rating history, while TMB only needed their social profiles.

1.5 Organization of the Thesis

The rest of this thesis is organized as follows:

- Chapter 2 explains the types of recommender systems, and describes the related work of each.
- Chapter 3 explains how book representations are created using author identification CNN model and how they are used for recommendations. The chapter also presents the evaluation and the results of this approach.
- Chapter 4 describes the linguistic feature selection, the recommendation algorithms, as well as the evaluation process and results.
- Chapter 5 describes the methodology, evaluation, and results of TMB, our proposed approach that addresses the user cold-start problem. Each of the methodology chapters ends with a summary that highlights its contributions and future work.
- Chapter 6 summarizes the thesis and suggests overall future work.

[14] `https://www.goodreads.com/`

1.6 Published Papers

- H. Alharthi, D. Inkpen, and S. Szpakowicz. A survey of book recommender systems. In *Journal of Intelligent Information Systems*, volume 51, pages 139–160. Springer, Aug 2018a. doi: 10.1007/s10844-017-0489-9. URL https://doi.org/10.1007/s10844-017-0489-9

- H. Alharthi, D. Inkpen, and S. Szpakowicz. Unsupervised topic modelling in a book recommender system for new users. In *SIGIR 2017 Workshop on eCommerce (ECOM17)*, 2017. ISBN 978-1-4503-5022-8. doi: 10.1145/3077136.3084367. URL http://doi.acm.org/10.1145/3077136.3084367

- H. Alharthi, D. Inkpen, and S. Szpakowicz. Authorship identification for literary book recommendations. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 390–400, 2018b. URL https://aclanthology.info/papers/C18-1033/c18-1033

- H. Alharthi and D. Inkpen. Study of Linguistic Features Incorporated in a Literary Book Recommender System. In *Proceedings of the 34th ACM/SIGAPP Symposium On Applied Computing (SAC '19)*, New York, NY, USA, April 2019. ACM. doi: https://doi.org/10.1145/3297280.3297382

Chapter 2

Background

This chapter[1] explains the different types of recommender systems and surveys book RSs under each category. It also highlights the datasets and resources dedicated to book RSs. Comparisons against our proposed approaches can be found in sections 2.2.2 and 2.3, which discuss RSs that exploit whole book texts and covers RSs related to TMB, respectively. In addition, sections 2.1 and 2.2.3 describe algorithms considered as baselines.

2.1 Collaborative Filtering

Collaborative Filtering (CF) assumes that if two users have similar rating history, their future ratings will be similar as well. This method uses the available ratings of active users to predict the preferences of other users. CF comes in two forms. User-based CF finds the similarity between users. Item-based CF computes the similarity between two co-rated items (rated by common users) (Sarwar et al., 2001). To make recommendations, CF only requires an item-user rating matrix as depicted in Table 2.1.

[1] This chapter in based on (Alharthi et al., 2018a)

Table 2.1: An example of a ratings matrix of 4 user-book preferences

	The alchemist	Life of Pi	1984	Alice in Wonderland
Anita	like	like		dislike
Bob		like	dislike	like
Maria		dislike	like	
Elsa	like		dislike	?

Techniques adopted in CF are categorized into memory-based and model-based. Neighborhood based CF, which falls under the former category, calculates the similarity between two users or two items and predicts ratings by computing the weighted aggregate of nearest neighbors' ratings. There are many similarity measures including Pearson correlation and cosine similarity. From the set of nearest neighbors, the CF recommends the most relevant predicted items as a ranked list (Su and Khoshgoftaar, 2009). One common model-based algorithm is matrix factorization (MF) that represents users and items in a space where each item/user is modeled as a vector of f latent factors. A user-item interaction is characterized as an inner product in the space. The predicted rating is the dot product between the user and item vectors (Koren et al., 2009). Matrix factorization, which we consider as a baseline, provided more accurate top-k recommendations when compared with neighborhood algorithm in (Hu et al., 2008)

However, the rating matrix can be large and sparse, particularly in the case of new items or new users; this is called the *cold start* problem, and it could result in inaccurate recommendations. Sparsity issues are common in libraries where many books have never been checked out. For example, 75% of the books in the library of the Changsha University of Science and Technology were not checked out (Yang et al., 2009). CF also suffers from two other problems. The *gray sheep* problem occurs when the system predicts the preferences of a user who has an entirely different taste from other users. The *shilling attacks* issue occurs when an item receives fake ratings as a form of promotion (Su and Khoshgoftaar, 2009).

Collaborative filtering is applied in many book RSs. An item-based CF implemented in Vaz et al. (2012b) calculates the cosine and Euclidean distance in $users \times books$ and $users \times authors$ rating matrices. The author RS and the book RS were evaluated using the LitRec dataset (section 3.3.1). The prediction performance was the best when 10% of author RS and 90% of book RS were merged. Vaz et al. (2013) assessed the temporal relevance of ratings in item-based CF. Experiments on a closed version of the LitRec dataset found that high prediction errors resulted from using only recent ratings but neglecting early ones[2]. It was also found that for recommendations of good quality one needs all ratings of the community but only recent ratings of the target user.

2.2 Content-based Recommender Systems

A content-based recommender system (CB) creates item representations and user profiles and matches them to make recommendations. The CB system consists of the following components:

- **A content analyzer** that transforms unstructured data of items to features. The new item representations are fed to the next components: the profile learner and filtering component. Items could be represented in many ways including a bag of words, a vector space model, and ontologies (*i.e.*, a class of domain knowledge connected with relations).
- **A profile learner** that develops a user profile by discovering patterns among representations of items consumed by the target user to make generalizations. The literature lists

[2] In a closed dataset, the rating matrix has no missing values: every user rates every book.

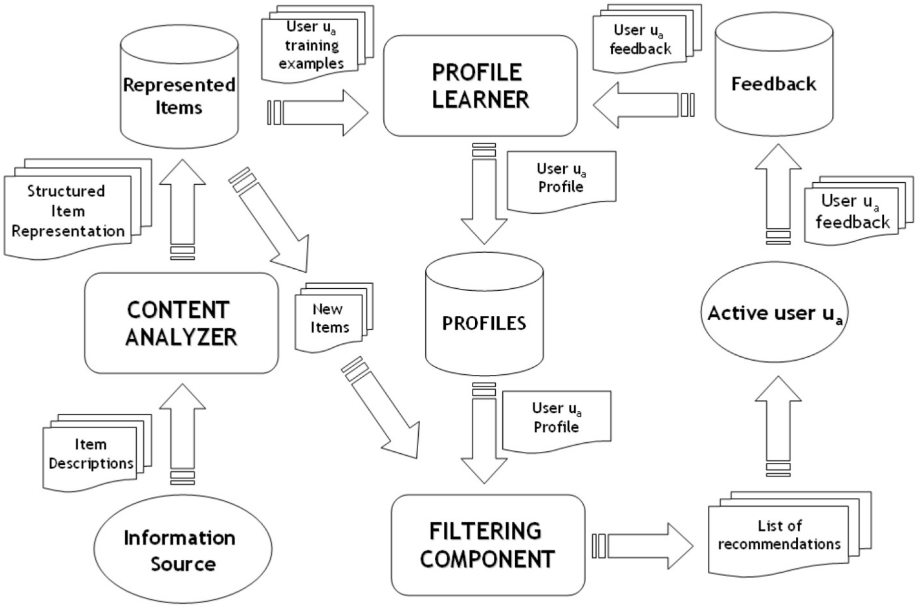


Figure 2.1: The process of recommendation in content-based RSs (Lops et al., 2011)

several machine learning algorithms used in CB system to build user profiles, including Naive Bayes (NB), support vector machines (SVM), decision trees (DT) and k-nearest neighbors (KNN).

- **A filtering component** that makes recommendations of items that correspond to the user profile (Lops et al., 2011).

In other words, a CB recommender is a classifier or regressor that learns the patterns and similarities in a user's purchase history to predict her future interests. In the field of books, the content might refer to title, summary, outline, whole text, or metadata, including author, year of publication, publisher, genre, size, and so on.

Figure 2.1 illustrates the process of recommendation in content-based RSs. The content of items is gathered from information sources and transformed into structured data using a content analyzer which passes the already rated items to the profile learner and the unrated items to the filtering component. After the profile learner constructs users' profiles, they are sent to the filtering component to match each profile with relevant items. The system provides a user with a list of predicted relevant items and receives the user feedback on them to send it later to the profile learner (Lops et al., 2011).

CB avoids CF's many issues by not relying on the ratings of the community, including the new item problem. Once a new item is added to the system, the CB can directly match it with user profiles. Also, it gives justified recommendations based on attributes of items. For instance, CB system would indicate that book A is recommended because the target user tends to prefer genre K and author M. Still, CB experiences difficulties when items have inadequate descriptions. *Overspecialization* is another problem with CB when the recommendations are not diverse enough. Like CF, CB suffers user cold start because CB expects enough data from one user to build her profile (Pazzani and Billsus, 2007; Lops et al., 2011). CB is widely used in book recommendations as shown in the following subsections.

2.2.1 Book recommendations based on their metadata

The use of book contents and reviews for recommendation purposes dates back to 1999. Mooney and Roy (2000) extracted book metadata from Amazon, specifically the title, writers, summary, reviews, comments, and similar authors, titles, and terms. When one removes the last three features, based on collaborative filtering, the performance of the CB decreases significantly. This system represented books as bags of words and adopted a binary classifier based on Naive Bayes.

Book metadata is deployed by Kapusuzoglu and Öguducu (2011) who use an extended ontology. If an ontology has the field *author*, for example, its extended ontology would include the author's awards and other publications. The system characterizes books as a relational database (multiple tables connected by foreign keys). Extended forms of cosine and Euclidean similarity measurements are proposed to handle the new ontologies. Experiments are run on 2791 server logs of an online bookstore. They show that stretched ontologies increase the accuracy of recommendations when compared with standard ontologies, especially with the proposed Euclidean distance.

2.2.2 Recommendations based on books' textual content

Out of more than thirty surveyed RSs, only a few take the actual text of books into account. Vaz et al. (2012a) proposed a Stylometric CB. A book is represented in two ways: as a vector of words extracted by LDA, or as a normalized vector of the style features, specifically vocabulary richness, document length, part-of-speech bigrams and the most frequent words in a book. The Rocchio algorithm (Manning et al., 2008) was adopted to discover stylometry features that matter the most to the reader. The experiments on the LitRec dataset showed that the merging of Stylometric CB with CF performs better than an individual CF or CB system. When the different representations of books were compared, LDA topics showed the best results.

The widespread of e-book readers introduces another level of book recommendations. E-readers allow the exploitation of the digitized book text. Given the whole text of an e-book, Zhang and Chow (2015) recommend authors and e-books after receiving the name of the user's favorite author. The system builds a four-layer tree for each author, with her background information (*e.g.*, education and political views) at the highest level, followed by the author's books, pages of every book, and paragraphs on each page in the next three layers. This hierarchical structure was proposed to overcome the problem of spatial distribution. The problem arises when sequences of words are treated without taking into consideration their context. For example, if one works with bags of words, "computer", "science" and "school" are treated as distinct terms, even if they occur as a trigram.

To deal with tree structures, Zhang and Chow (2015) adopted a multilayer self-organizing map (MLSOM) (Rahman et al., 2007). The book RSs use the last three layers to measure the similarity between two books. The author RSs use all layers to find similar authors. The dataset extracted from Project Gutenberg contained the text of 10,500 books and 3868 items of author information. This dataset contains neither users nor their ratings. The goal of the experiment is to "assess the relevance between two books" (Zhang and Chow, 2015). If the queried book shares similar genres with the retrieved book, it is considered a relevant recommendation. The system is reported to surpass the performance of other CB systems using LSI, LDA and PLSA; however, LDA gives a similar performance.

Givon and Lavrenko (2009) address the problem of *cold start* of items; the system incorporates social tags. Each book is characterized by *tf-idf* (term frequency – inverse document frequency) vectors of social tags (extracted from the book cataloging website, LibraryThing[3]) and book tags (extracted from the whole text of a book). For new books with no available social tags, a relevance model (RM) is adopted to learn from a book's tags to predict social tags. A comparison of item-based CF, user-based CF and a combination of CF and RM is presented. The pure RM gives results similar to those of CF systems. Also, book recommendations are no different whether they are on actual or predicted social tags.

We are not aware of any work exploiting stylometry features other than (Vaz et al., 2012a), which includes only two of our considered linguistic features namely document length and vocabulary richness. Our linguistic elements contain a variety of syntactic, lexical, stylistic and fiction-based features. Moreover, we illustrate the factors that play a role in the generation of accurate recommendations. Our proposed system based on authorship identification does not require demographic information about authors like (Zhang and Chow, 2015), which is not always obtainable. Also, unlike (Zhang and Chow, 2015), our CB systems are designed and assessed to make personalized recommendations for each user. Pera and Ng (2014a,b, 2015)

[3] `https://www.librarything.com/`

incorporate writing style in combination with other factors in a book RSs (explained in details in 2.3.3). The writing style, however, was learned from the book reviewers' point of view and not automatically from books textual content. Book reviews, which are not always abundant (if any), are influenced by users' biases and level of judgment.

2.2.3 Recommendations of text-based items

The basic approach in book retrieval systems is to adopt the bag-of-words method, which creates book representations based on term frequencies. In this case, cosine similarity is often used to retrieve the most similar book representations. Vector space model (VSM) was used to find similar books based on their descriptions in (Tsuji et al., 2014; Pera and Ng, 2014a) and not their actual text.

Other popular information retrieval approaches using topic-modeling techniques are also used in textual item recommendations. Topic modeling methods learn latent topics from a corpus where a mixture of topics characterizes each document. One technique is latent semantic indexing (LSI) (Deerwester et al., 1990), which is also called Latent Semantic Analysis (LSA). LSI performs singular value decomposition (SVD), which is a dimensionality reduction method, on a set of documents to learn words' contextual meanings. Afterward, LSI represents documents (either seen before or new) in a "semantic space" where relevant documents are considered similar (Landauer et al., 1998). Latent Dirichlet allocation (LDA) (Blei et al., 2003) represents a document as a mixture of topics and measures to what degree a word is associated with each of the topics. It assumes that a document is related to a limited number of topics, and a topic has limited reoccurring terms (Girolami and Kabán, 2003). Topic models have helped estimate preferences in many RSs. To name a few, recommendations were based on the topics extracted from movie plots (Bergamaschi and Po, 2015), articles (Nikolenko, 2015; Wang and Blei, 2011) and online courses syllabi (Apaza et al., 2014).

A more recent state-of-the-art approach is paragraph2vec, also called Doc2vec, which was proposed by Le and Mikolov (2014) to provide a representation of fixed size for documents regardless of their lengths (see 2.6 for more details). In recommender systems, Doc2vec was used by Gupta and Varma (2017) and Wang et al. (2016) to recommend scientific articles and answers in question-answering systems respectively. All techniques in this subsection are considered as baselines.

2.3 Social Recommender Systems and User-generated Content

Social recommender systems exploit posts, relationships, tags and other content found on social media to make suggestions (Tang et al., 2013). Recommendations by friends are found to be more trusted than by other users (Ricci et al., 2011). This type of RSs usually complements other RSs to solve the new user problem. The following three subsections present RSs exploiting book-cataloging networks, social media and users shared book-reviews.

2.3.1 Recommendations on book-cataloguing platforms

One system by Pera et al. (2011) takes advantage of LibraryThing, a social book cataloging Web site. It allows users to form friendships, and to catalog and tag books. The proposed system measures the similarity between books cataloged/liked by a target user and books cataloged/liked by her friends. The similarity is calculated in two ways. First, since each book in LibraryThing has a tag cloud attached to it, the system finds the similarity of tag-represented books, using a word-correlation matrix. Second, it computes the strength of friendship relations, in that it measures the resemblance between the tags assigned by a user and her friends.

The ranking of books is based on a single ranking score calculated using the joint product of the two methods' similarity values. For assessing the system, data of appraisers is extracted from LibraryThing, and the results are compared with Amazon's and LibraryThing's lists of recommendations; they significantly outperform both.

Tags are also integrated into the system in (Pera and Ng, 2011), which finds similar books in a user's friend list. Books are considered similar when they share one or more tags with friends, or when are highly rated by friends. The system goes further by measuring the reliability of a user's friends. The most reliable friend is the one with the highest number of mutual tags. Similar to the evaluation in (Pera et al., 2011), this system's recommendations surpass Amazon's and LibraryThing's.

Users in (Zhou, 2010) are represented as nodes in a network, and trust among them is calculated using inference and propagation. When users consider buying books, they request recommendations of their neighbors, who in turn pass the request to their neighbors. Alternatively, a user may receive suggestions from a CB system. Books in the system are also embodied as nodes connected because of the similarity in their content. However, no experiments or results are reported.

To our knowledge, no book RSs exploit social networks other than book-cataloging websites. Also, previous social book RSs relied on the use of tags (Givon and Lavrenko, 2009), (Pera and Ng, 2011) and(Pera et al., 2011) as well as friends' lists. Thus, TMB is the first social book RS that automatically extracts user interests from general social media and map them to books subjects.

2.3.2 Recommendations based on social media

Social media has been a great resource to ”warm up” the user cold start. A user's connections on social network were exploited in (Castillejo et al., 2012), (Sedhain et al., 2014), (Guy et al., 2010), (Ben-Shimon et al., 2007) and (Mican et al., 2012). In addition to using Facebook friends lists, Sedhain et al. (2014) analyzed users' demographics, and pages liked by a user. Nair et al. (2016) solved the new user issue by analyzing a target user 's tweets and identifying which movie genres she likes. The cosine similarity between a tweet and a movie storyline is calculated. If the similarity is higher than 0.5, the movie's genre is added to the user's favorite genres. Later, movies from the most frequent genres are recommended. Based on topics learned from users' Twitter accounts, RSs could suggest hashtags (Godin et al., 2013) and friends (Pennacchiotti and Gurumurthy, 2011). TMB, on the other hand, addresses the new user issue by exploiting tweets to recommend items that are not Twitter-relevant (*e.g.*, not hashtags).

To make news recommendations, Abel et al. (2011) treat a user profile as a query; the k most similar candidate news articles are recommended. User profiles are constructed from three elements: hashtags, entities, and topics. A concept is weighted by counting the times a user mentions it (*e.g.*, #technology = 5). A framework, OpenCalais, is used to spot the names of people, places and other entities in addition to topics; there is a limitation to 18 different topics (*e.g.*, politics or sports). All articles published by 1619 Twitter users in the last week of observation time are considered as candidates for recommendation. Entity-based user profiles scored the highest accuracy.

Chen et al. (2010) propose a Twitter-based URL recommender. Cosine similarity is computed between user profiles and URL topics, and the system recommends URL items with the highest scores. For each user, self-profile and followee-profile are constructed out of bag-of-words. For a URL, a bag-of-words is also created out of terms occurring in tweets which embed the URL. In a field experiment, 44 participants rated the recommended URLs. The best performance was 72.1% accuracy when the RS used self-profiles and candidate URLs from FoF (followee-of-followees).

Unlike work in (De Francisci Morales et al., 2012; Abel et al., 2011; Jonnalagedda et al., 2016) which looks for news-related and narrow lists of entities and categories, TMB is dynamic and represents the dominant topics discussed by a user without searching for predefined concepts. Our proposed system does not require entity recognition or ontology development. Moreover, our system focuses on a limited number of topics frequently discussed by the user, and this makes it easier to enrich the topics, *e.g.*, with word embeddings. Furthermore, as mentioned in section 1.2.2, news and books have different characteristics. News recommendation using Twitter may require the analysis of hashtags and entities such as names and places that may correspond with the rapidly changing news. However, literary books may include broad aspects that are mostly unrelated to present names and actions.

2.3.3 Recommendations based on user-generated texts

Many readers share their opinions about books on the Web. Such views offer a chance to learn user preferences in detail. Research discussed in this section ranked books based on the similarity of their reviews, and on the correspondence in keywords (established in advance).

In (Garrido et al., 2014), for each book preferred by a user, a topic map is created based on the book description and user reviews. A topic map is a form of ontology built using TM-Gen (Garrido et al., 2013), which represents the extracted text as a map. Before the extraction, many NLP techniques are applied, including morphological analysis and Named Entity Recognition. All the topic maps of the favorite books of a user are aggregated and compared to candidate books with the same representation. The system is evaluated using the BookCrossing dataset. It produces fewer errors than some implemented state-of-the-art systems.

In a series of RSs for emergent readers (Pera and Ng, 2014a,b, 2015), book recommendations are made based on four factors: readability and similarity in content, topics, and appealing terms. The readability level of a user is measured and matched with a set of candidate books.

To measure readability, TRoL (Pera, 2014) and ReLAT (Pera and Ng, 2013) are used to decide the level of a book without the need for an excerpt. For content similarity, publicly available summaries of books are represented as bags of words, and the similarity is calculated using word-correlation factors (WCF) (Koberstein and Ng, 2006). The topical similarity depends on the Library Congress Subject Headings (LCSH). To find the resemblance of the topic distribution of books, a VSM is created, and the similarity is calculated between vectors. In (Pera and Ng, 2015), a considerable number of LCSH associated with a book is penalized, because they entail content complexity.

Finally, to analyze the appealing terms of books in (Pera and Ng, 2014a,b, 2015), the literature of reader's advisory is consulted, and six facets of books are determined: *characterization, frame, language and writing style, pacing, special topics, storyline*, and *tone*. Usually, appealing terms associated with each book can be obtained by readers' advisory from databases, such as NoveList Plus which requires paid access. Therefore, in this work, the appealing terms are extracted from readers' reviews, automatically collected from websites such as Amazon.com, Bertrams.com, Bookfinder4u.com, Bookmooch.com, Dogobooks.com, and Fishpond.com. 124 predefined terms classified under each of the six facets are extracted from reviews, as Pera and Ng (2014b) explain in detail. The resemblance between vectors of appealing terms of the favored and candidate books is calculated.

The RS in (Pera and Ng, 2015) considers the illustrations on book covers. It uses the Open Source Computer Vision (OpenCV) library to check the resemblance of covers which are freely available via the APIs of Google Books and LibraryThing. Pera and Ng (2013) extend the RS — it comprises the readability level and content similarity — by including readership similarity, which is simply an item-based CF. The above similarity scores are combined using multiple linear regression in (Pera and Ng, 2014b), CombMNZ in (Pera and Ng, 2014a, 2015), and Borda counts in (Pera and Ng, 2013). The systems are compared to other popular RSs, *e.g.*, Amazon and Goodreads, or to previous versions of the system. For the comparison and additional qualitative analysis, appraisers rate the system using Amazon Mechanical Turk[4].

The reviews of a specific user can be extracted to recognize her fine-grained interests. To make personalized book recommendations to an individual, Priyanka et al. (2015) analyzes each user's reviews. The sentiment of each review is assessed by only counting the occurrences of positive and negative words. For each user, a matrix is created. Its rows and columns represent books and features, extracted from the user reviews. For example, one column can be "understand" which can have a positive, negative or neutral value in each row (book). The sentiment of the total value of a feature is computed. The approach is not evaluated, and no performance was reported. Sohail et al. (2013) adopts a non-automatic book recommendation method based on opinion mining. To discover the top-rated computer science books, the reviews of the books are exploited. Seven categories of features are analyzed, and each is assigned a weight. The weights are aggregated, and books are re-ranked accordingly. No evaluation is reported.

2.4 Book Recommendations based on Association Rules

Book recommendations can also use unsupervised learning by employing association rules, as in (Rajpurkar et al., 2015; Tsuji et al., 2014; Maneewongvatana and Maneewongvatana, 2010; Zhu and yan Wang, 2007). It works by finding patterns in an extensive database of library transactions. If two co-preferred books are associated, the occurrence of one of them in a transaction implies the occurrence of the other. To evaluate the reliability of a rule, the confidence and support are computed. The confidence of a rule $(X, Y) \rightarrow Z$ is the number of transactions that contain X and Y in addition to Z, while support is the number of transactions

[4] `https://www.mturk.com/mturk/welcome`

that have X and Y (Hahsler et al., 2005).

In libraries, data about members' demographics and reading activities are stored and can be retrieved. To increase the efficiency of public and school libraries, one must consider a personalized system that understands each patron's needs. Such a system allows members to take advantage of a library's abundant resources, some of which may have never been checked out (Yang et al., 2009).

Circulation analysis is an extensively researched field. We only highlight some of that research. To deal with library loan records, association rules, and clustering techniques are usually applied to the recognition of patterns in the circulation. Also, the categories of the Library of Congress Classification (LCC) and Dewey Decimal Classification (DDC) — classification systems already used in libraries — are useful in making recommendations. In LCC, all books are categorized into twenty-one classes; each is denoted by a letter, and extended into two or three letters for further subclasses. For example, class H refers to social sciences, and its subclass HM denotes sociology.[5] DDC consists of ten broad classes, each involving ten divisions, which in turn contain ten sections each[6]. For evaluation, most researchers had access to student records in a university library. They conducted surveys which included asking participants to rate some items in order to calculate the accuracy of their systems.

Tsuji et al. (2014) analyze over two million loan records at a university library. A user of the system gives a query on which to base the recommendations. Many sources for the making of recommendations have been compared. First, a loan is considered as a transaction, and association rules are applied. A book is recommended if it co-occurs with the queried book with high confidence and support. Second, nouns in titles and outlines of books are represented as vectors of tf-idf weights to retrieve books similar to the query; cosine similarity was applied.

[5] http://tinyurl.com/je3662l

[6] http://tinyurl.com/hgc4tn2

Third, a book is also recommended if its categories, divisions, and sections of Nippon Decimal Classification (NDC) — a Japanese library system based on DDC — agreed with the queried book. When comparing the different sources, the use of association rules in addition to titles gave the best results. They were, however, compared with the results of the Amazon RS, and found to be less accurate.

Maneewongvatana and Maneewongvatana (2010) also study the circulation of a university library. Members' check-outs, reservations, and renewals were collected. Each member was represented by a vector of LCC categories. After members had been clustered using K-means, Apriori (Agrawal and Srikant, 1994) and Tertius (Flach and Lachiche, 2001) were used to discover association rules in every cluster. Patrons are given recommendations of books associated with books which they already read. Based on the rating of 14 members, an accuracy of 42% was reported. Another system (Yang et al., 2009) creates user profiles based on members' demographics, and on the attributes of checked-out books, *i.e.*, LCC, and the Chinese classification system. Similar user profiles are clustered, and recommendations are made correspondingly.

2.5 Other Approaches to Book Recommendations

One method used for book recommendations is context-aware recommender system (CARS), which includes the contextual data about a target user into the recommendation procedure (*e.g.*, times and location of movies in theaters) (Shani and Gunawardana, 2011). Pathak et al. (2013) combine sequentially three RSs: CB, CF, and context-aware RS. In the beginning, CB filters books based on a user's predefined topics, from which the highest-ranked recent books are recommended. This hybrid system outperforms each of the three individual systems.

Another approach is demographic RSs that associate items with users' demographic classes

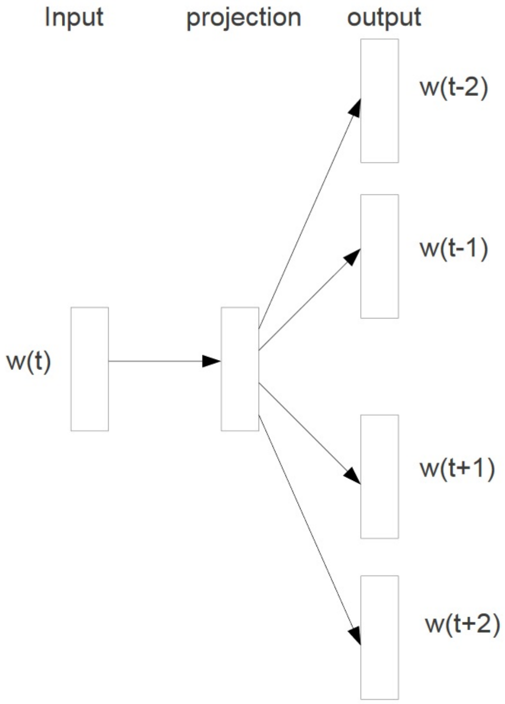


Figure 2.2: The architecture of Skip-gram model (Mikolov et al., 2013)

based on age, gender, country, etc. They overcome the new user problem as they do not expect a rating history to make recommendations. Moreover, they are not dependent on a specific domain, *e.g.*, movies vs. books (Shani and Gunawardana, 2011). However, the collection of users' information may disrupt their privacy. Mikawa et al. (2011) employ SVM to classify the gender and age of people walking into a library. The SVM is fed sequences of images captured by a camera at the library entrance. The books are filtered by whether a patron is male or female, young or middle-aged.

2.6 Deep Learning and Recommender Systems

2.6.1 Basics of deep neural networks

In recent years, word embeddings have gained much attention. They are dense representations of words built with the use of neural networks. There are many unsupervised learning approaches to building word embeddings, including word2vec, GloVe, FastText and ELMo (Embeddings from Language Model). Word2vec (Mikolov et al., 2013) is shallow NN that takes large text corpus and generates word vectors in a space where two words occurring in similar contexts are neighbors. Word2vec comes in two forms: Continuous Bag-of-Words model (CBOW) and the Skip-Gram model. Some parameters are set before training the model such as the word vector's length and the window size (context words). CBOW takes the vectors of words, which occur in a specified window, and predict one word. In Skip-Gram (as in figure 2.2), the model takes one input word vector and predicts the words that occur in the same window. The Skip-gram model is found to be more accurate when dealing with rare words (Mikolov et al., 2013; Lau and Baldwin, 2016). Doc2vec is proposed later by Le and Mikolov (2014) to represent texts of various length whether it is a paragraph or large document. It extends CBOW model by introducing an additional vector for document id. The model predicts the next word given an input of the document vector and context word vectors (concatenated (joined) or averaged). After training, a document can be represented by its id vector (Le and Mikolov, 2014).

In (Mikolov et al., 2013), a high cosine similarity is found between semantically similar words. Cosine similarity does not rely on vector magnitude, which in case of word embeddings depends on word frequency (except for words that occur in conflicting contexts such as May (month)) (Schakel and Wilson, 2015). This makes it suitable to measure the similarity between two semantically similar words regardless of their frequency.

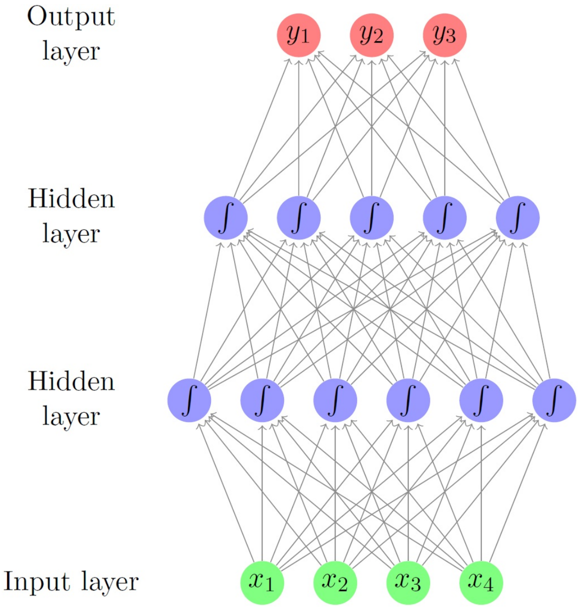


Figure 2.3: Feed-forward neural network (Goldberg, 2016)

The simplest neural network is a feed-forward NN which tries to learn a function that maps an input x to an output y in that $y = f(x; \Theta)$ by searching for the Θ that gives the best estimation. The feedforward NN consists of input, output and one or more hidden layers as depicted in Figure 2.3. The circles represent neurons, and the arrows show the flow of information in the network which as the name suggests is always moving forward (from input to output layer). In a hidden layer (also called fully-connected or dense layer), every neuron is connected to all neurons in the next layer. The basic unit in NN is the artificial neuron (figure 2.4[7]), which receives the inputs passed from the previous layer multiplied by different weights, calculates their weighted sum (plus bias) and feeds them into an activation function that transmits the output to the following layer (Goldberg, 2016).

An activation function (non-linearity) decides which neurons fire (i.e., are activated). One

[7] Illustration by `https://tinyurl.com/yb72e8tk`

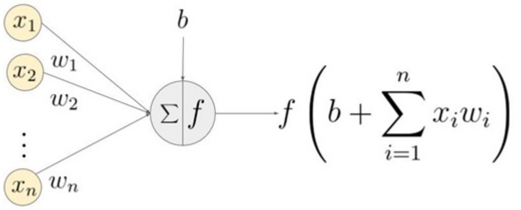


Figure 2.4: Illustration of the calculation inside one artificial neuron

popular non-linearity function is rectified linear unit (ReLU) (Nair and Hinton, 2010), which as in equation 2.1 only activates the neurons with positive outputs. ReLU can deal with the gradient vanishing issue, which happens when the gradient becomes very small and the weights are not updated efficiently, better than other functions, *e.g.*, sigmoid (Xu et al., 2015).

$$relu(x) = max(0, x) \tag{2.1}$$

Training the network requires a loss function such as categorical cross-entropy loss (equation 2.2) that is used in multi-class classification. The loss computes the difference between the probabilistic distribution of the actual labels y and predicted labels $\hat{y}$ produced by a softmax layer. The goal is to minimize the loss over the training samples by iteratively updating the parameters of the network to the opposite direction of the gradient of the loss function. The parameters are the weights, biases, and sometimes embeddings (if not fixed) (Goldberg, 2016).

$$L_{cross-entropy}(\hat{y}, y) = \sum_{i} y_i log(\hat{y}_i) \tag{2.2}$$

Unlike feed-forward NNs, Recurrent Neural Networks (RNN) have feedback connections which allow signals to loop. In a simple RNN, the input of the network is combined with the previously computed hidden node activations. This allows RNN to work with sequential data. Nevertheless, RNN suffers from the vanishing gradient issue which results in a decay of information in a long sequence. Long short-term memory (LSTM) and Gated recurrent units (GRU) are two variations of RNN that overcome this issue (Chen, 2016; Goldberg, 2016).

Another deep learning architecture, which was invented by researchers in computer vision, is Convolutional Neural Network (CNN). It showed state-of-the-art performance in multiple NLP tasks including text classification (Johnson and Zhang, 2015) and author identification in (Solorio et al., 2017). In text-related tasks, a one-dimensional CNN is applied. A CNN layer has one or more associated filters where a filter, which is basically a matrix of randomly-initialized parameters, moves over all the instantiations of windows of tokens (i.e., regions). The filter should eventually identify the patterns distinguishing a class from another regardless of their place in the text. Each filter produces a variable-length activation map (feature map), which is a vector with each dimension representing the result of the filter multiplication with a specific region followed by a non-linearity (see section 3.2 for formal description) (Goldberg, 2016). One hyperparameter of CNN is stride length that decides the steps a filter takes while moving forward. It is common in the literature to move the filter one step at a time to capture all the variation of the input. Also, it is common to apply padding, which means adding values (*e.g.*, zero) to the beginning and end of short text sequences to make them equal to the maximum text length(Tixier, 2018).

A CNN layer is usually followed by a pooling layer that decreases the dimensionality of the generated feature maps, which helps in reducing the number of network parameters and accelerating the training. Two main pooling approaches are usually used: max and average pooling. The former returns the maximum value per feature map whereas the latter computes the mean value of each feature map. The idea behind max pooling is to obtain the most salient information that helps in the prediction task regardless of their position in the text (Goldberg, 2016; Kim, 2014).

2.6.2 Deep learning recommender systems

Neural collaborative filtering (NCF) (He et al., 2017) deploys multilayer perceptron (MLP) by developing an embedding of a user and an item (concatenated). With the help of hidden layers, the non-linear interactions between users and items are learned. NCF is extended in (Wang et al., 2017) to tackle a new issue called cross-domain social recommendations. The system (NSCR) takes advantage of the user-item interactions in an information domain (*e.g.*, Tripcase) to make recommendations in the social domain (*e.g.*, to Facebook friends). The user embeddings learned in the information domain are propagated to the social domain to help in creating other users embeddings.

Musto et al. (2016) propose a system called Ask Me Any Rating (AMAR) which incorporates RNN to learn the sequences of words describing items. The RNN output is fed to a mean pooling layer which generates an embedding for each item. An item embedding is concatenated with a unique user embedding and fed to logistic regression layer which predicts a user interest in an item. We have implemented a similar approach to (He et al., 2017) and (Musto et al., 2016) and customized them to work with book features. The preliminary experiments did not show high performance in top-k ranking settings. As NN algorithms require a large dataset to perform well, it is possible that our dataset has no enough user-item interactions to be able to learn a meaningful function. We also considered using embeddings for fictional characters similar to (Grayson et al., 2016) in book RSs, but they did not achieve high recommendation accuracy as shown in appendix C.2.

Furthermore, unsupervised NNs were used to generate recommendations. A CF based on Restricted Boltzmann Machines (RBM) was first proposed by (Salakhutdinov et al., 2007). For each user, an RBM model is created with her seen movies represented as nodes in the visible layer connected to all hidden nodes. The weights and biases corresponding to specific movie are shared among all RBM models. Moreover, generative adversarial networks (GANs) were applied recently to RSs, including (He et al., 2018).

2.7 Datasets for Evaluating Book RSs

There are existing datasets with distinguished attributes as shown in table 2.2; they can help evaluating different book RSs. Due to the lack of a dataset that contains information on books and user social media accounts, we collected it from Twitter and Goodreads as described in section 5.3.1. We also used LitRec dataset (Vaz et al., 2012c) which has users' ratings from Goodreads and book texts from Project Gutenberg (further details on LitRec dataset in section 3.3.1).

Goodreads have about 55 million users, 1.5 billion books and 50 million reviews. It delivers information about users' demographics, tags, reviews, friend lists, reading groups, and favorite quotes. It also provides full access to book metadata including the number of ratings and reviews received, and the average rating. The API user can also extract information about authors[8].

Project Gutenberg[9] has more than 50,000 out-of-copyright digital books made available; they were published at least 50 years ago. Even though it mainly consists of novels, short stories and other literary works, it has many nonfictional works. The collection includes approximately all publications of English canon literature that precede 1923. Unlike texts in HathiTrust and Google Books which were scanned by OCR, items in Gutenberg have gone through proofreading or even were hand-typed (Brooke et al., 2015).

[8] http://www.goodreads.com/about/us

[9] https://www.gutenberg.org/

[10] http://www.macle.nl/tud/LT/

[11] http://inex.mmci.uni-saarland.de/data/documentcollection.html#books

	Book-Crossing **(Ziegler et al., 2005)**	**LitRec** **(Vaz et al., 2012c)**	**LibraryThing**[10]	**INEX**[11]	**Amazon reviews** **(McAuley and Leskovec, 2013)**
Rating form	1-10	1-5	1-10	1-10	1-5
Demographics	locations, ages	locations			Amazon user id
Book metadata	title, authors, year, publisher, cover image	title, authors		title, authors, publisher, year	Amazon book id, title, price
Book summary				yes	
Complete text of a book		yes			
Reading start and end dates		yes			
User-generated tags			yes	yes	
Semantics (mapped to DBpedia)			yes		
Textual reviews				yes	yes
Users' requests for recommendations				yes	

Table 2.2: Features of book-recommendation datasets

Chapter 3

Authorship Identification for Literary Book Recommendations

3.1 Overview

One factor that influences reading preferences is writing style (Smith et al., 2016). In this chapter[1], we propose an authorship-based RS that recommends books after learning their authors' style. It is a content-based RS that analyzes the texts of books to learn users' reading interests. Our book RS transfers information learned by an authorship identification (AuthId) classifier to a book recommendation module. It is common in neural network literature, especially in image processing, to train a model (source model) on a dataset for a specific task and then transfer the learned features to another model (target model) working with a different dataset and task. In particular, CNN models are used as feature extractors (Athiwaratkun and Kang, 2015). The features are considered *general* if they are learned from first layers in neural networks. *General* features tend to hold basic information (*e.g.,* , color blobs in image processing), and are

[1] This chapter in based on (Alharthi et al., 2018b)

suitable to work on a different dataset/task. *Specific* features, on the other hand, generated by the last layers, are very dependent on the dataset/task (Yosinski et al., 2014). Transferability in NLP applications is explored in (Mou et al., 2016) which concludes its usefulness for tasks that are mutually semantically similar.

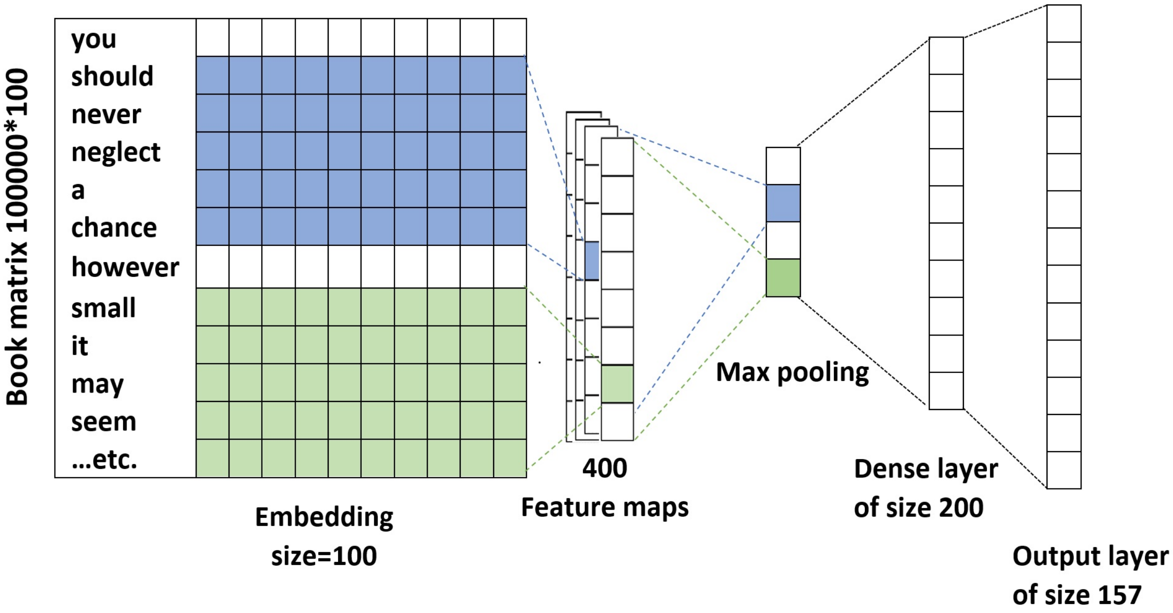
One paper (Razavian et al., 2014) trained an SVM over image representations extracted from a fully connected layer in a pre-trained CNN model and achieved superior results in image retrieval, object image classification and other tasks. Another work (Athiwaratkun and Kang, 2015) yields an increase in performance when feeding CNN extracted features into Random Forests and SVM to perform the original CNN prediction task. The study also shows that features extracted from an overfitted or underfitted (stopped at early epochs) model can also result in accurate classification.

There are many NN based authorship identification approaches that achieved high accuracy. Recent research also shows how the use of NN models has led to accurate author identification. The work in (Qian et al., 2017) achieved 89% accuracy using Gated Recurrent Unit (GRU), an RNN algorithm, on a dataset from the Gutenberg Project. Their model represents a sequence of words (initialized as GloVe pre-trained embeddings) in a sequence followed by average pooling, and then another GRU that represents the sequence of sentences in an article. Our preliminary experiments show using RNN-based author identification models with books results in poor accuracy; therefore, we did not adopt RNN models. A simple system using CNN is proposed in (Solorio et al., 2017). It takes a sequence of tokens of a tweet as input and predicts its author, a Twitter user. The system performance is evaluated with different inputs (including character bigrams, character unigrams and words). Although it has been proposed for short texts, we noticed that CNN gives acceptable accuracy when predicting authors of books; that is why we have adopted it.

The authorship-based recommender system has two components: authorship identification and book recommendation. For the former, we adopt an approach inspired by (Solorio et al., 2017); it uses CNN over a sequence of words. Given the text of a book as input, the AuthId classifier predicts its author, and once it has achieved good accuracy, we extract features from the last hidden layer (before the output layer) and use them for another task (book recommendation). These AuthId book features, then, are *specific*. In fact, the first task, author identification, is just a way of representing books as vectors that encode information about their authors' writing styles. The recommendation module, on the other hand, is a content-based RS which makes recommendations after finding patterns in the representations of books read by the target user. To achieve this, a regressor is trained over book AuthId features associated with the target user ratings to generate a list of ranked books.

As presented earlier, learning users' reading preferences from books' texts is rarely practiced. Recommendations based on a limited number of stylometric features are explored in (Vaz et al., 2012a); however, the features are not associate with authors. Moreover, the author's writing style was also considered in (Pera and Ng, 2014b,a, 2015), but it is learned from the online reviewers' point of view and not automatically from the textual content of books. The reliance on users' reviews might introduce bias and would require books in the systems to be widely read and reviewed. Furthermore, an RS is proposed in (Zhang and Chow, 2015), which exploits the whole text of an e-book to suggest authors and e-book. This approach does not focus on authorship identification, and it requires writers' demographic information. Also, one can exploit experts' description of authors writing styles in book RSs, but not all books are manually tagged (*e.g.*, NoveList lacks the information of many authors).

Figure 3.1: The use of a convolutional neural network for authorship identification



3.2 Methodology

This section describes the two components of the system. It first illustrates and explains the author identification system. Next, the recommendation procedure is described.

3.2.1 Author identification

As shown in Figure 3.1, the neural network consists of the layers described bellow. Our work differs from (Solorio et al., 2017) in that we tried different hyperparameters setting, did not use n-grams, and added a fully connected layer to work as the AuthId features.

Embedding layer (also called lookup table). It takes a sequence of words as input and maps each word to a non-static (trainable) pretrained embedding of size *k*. The embedding of the *i*th token is a dense *k*-dimensional vector $x_i \in R^k$. Each book is represented as a matrix with each word embedding in one row. A book has sequence length *n* (*n* is the number of tokens) where a sequence $x_{1:n} = x_1 \oplus x_2 ... \oplus x_n$ is a concatenation of all tokens from x_1 to x_n.

Convolution layer (CNN). This is a one-dimensional CNN layer that deals with temporal sequences (i.e., not spatial). A CNN layer consists of one or more filters that are applied to windows of tokens to generate feature maps. Let a filter $w \in R^{hk}$ slide over a window of *h* words $x_{i:i+h-1}$ to generate feature $c_i = f(w \cdot x_{i:i+h-1} + b)$. *f* is the activation function, and *b* is a bias. A feature map **c** $= [c_1, c_2, ...c_{n-h+1}]$ is created by a filter *w* that slides over all the possible windows of words in a book. This layer generates *m* feature maps equal to the number of filters (specified parameter) with variant lengths depending on the book sizes.

Pooling. This layer decreases the dimensionality of the feature map. Given a feature map **c**, global max pooling returns *max{**c**}*; as a result, only the features with the highest importance (maximum value) are kept (Kim, 2014). It generates a vector, which we call AuthId_MP, of the same size as the number of feature maps.

Fully connected layer (FC). Each neuron in this layer is connected to all neurons in the previous layer. The weights associated with every node are initialized and updated independently from other nodes which allows it to represent a function that would ideally specialize in a specific aspect of the input. We added this layer to increase the classifier performance and discover interactions and dependencies between the features extracted by the pooling layer. This layer produces a fixed-size vector with dimensions equal to its number of neurons. We call this vector the AuthId book representation or AuthId_FC.

Output layer. It is a fully connected softmax layer that outputs a vector with dimensions equal to the number of authors (labels). Each dimension represents the probability of the book to a specific author — values of all dimensions in one vector sum to one.

The network training is performed by minimizing loss function over batches of the training set. After each epoch, the errors are back-propagated and the parameters of CNN filters, word embeddings, weights of the FC layer and bias are updated to reduce the loss. This iterative process will eventually help CNN filters focus on the aspects that are most important to distinguish authors from one another (Goldberg, 2016; Kim, 2014). After the model has been trained and achieved accurate predictions, it is used to extract the features of max pooling or FC layer. Because ReLU is used as the activation function, each dimension in the generated vectors has a value of zero or higher.

3.2.2 Book recommendations

Given a user's reading history associated with book representations, a regressor predicts her future ratings. Book recommendations are ranked according to continuous predicted rating values; hence, the ranking list is unique. We applied Support Vector Regression (SVR)[2] which is an extension of Support Vector Machine (SVM) (Vapnik et al., 1996). In a non-linear SVR, the training samples X are mapped to a high-dimensional feature space which allows it to learn a linear model in that space. The mapping is achieved by a kernel function such as Gaussian kernel which also called Radial Basis Function (RBF)—see Equation 3.1. For every AuthId book representation x_i that has the rating y_i, the SVR algorithm aims to learn a function f(x) as in Equation 3.2 with α_i^*, where α_i are Lagrange multipliers and N is the number of data points (Gunn, 1998; Basak et al., 2007).

[2] Experiments showed that SVR gives the best performance compared to other regressors

$$k(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{3.1}$$

$$f(x) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)k(x_i, x) + b \tag{3.2}$$

3.3 Evaluation

3.3.1 Dataset and preprocessing

We use the Litrec dataset (Vaz et al., 2012c) that contains data on 1,927 users who rated 3,710 literary works. The dataset incorporates data from Goodreads and Project Gutenberg. It also has the complete texts of books, labeled with part-of-speech tags (Vaz et al., 2012b). In Goodreads, books are rated on a scale of 1-5 where 1-2 indicate a dislike and 3-5 a like; we adopted this range when converting the ratings to binary. In the dataset, a book can have a rating of 0 to indicate that the user has read the book but not rated it. We filtered out all zero-rated books (17,976 ratings) because it is not clear whether the user likes them or not. Also, in order to train the author identification network, we need multiple books per author; thus, we only kept authors with a minimum of two books. Data for users with fewer than ten ratings are also deleted. The lowest number of ratings needed to develop CB with quality recommendations is ten, a threshold adopted by many researchers, including (Wang et al., 2009). The remaining 351 users rated 1010 unique items authored by 157 distinct authors.

The Gutenberg texts have copyright information at the beginning and the end, which we removed using heuristics or manually. The part-of-speech tags are also removed. Some books are very long; the maximum is 565,570 words, while the average is 99,601 words per book. Because of the high memory requirement, the processing of large books resulted in a system crash. That is why we considered only the first 100,000 words of each book—slightly higher than the average length[3].

3.3.2 The experimental setting

We measure the predictive power of the system using off-line evaluation, which is appropriate for obtaining the accuracy of an RS. The online appraisal would provide more performance insights, but it is an expensive option that requires the deployment of a real-time system. A user study is another option; it was avoided because it usually includes a limited number of users. In the off-line assessment, systems are evaluated in three ways: rating prediction, ranking and classification. A recent survey (Zhang et al., 2017) states that the percentage of the surveyed RS publications that report ranking metrics is 66%, rating prediction metrics is 28%, and usage metrics 6%. We adopt ranking metrics to evaluate our system using a top-k recommendation scenario where systems provide a user with a ranked list of books. Recommending top-k items is a common practice in real-world RSs particularly in content and multimedia recommendations (Schröder et al., 2011).

In content-based RSs, a user's books are divided into training and testing sets, and the CB learns the user preference only from the training set. Though the same split of training and testing sets per user is applied in the MF system, MF is trained on the whole rating-matrix excluding the target user's ratings on the items in the test set. For a user, each system generates a ranked list from all the books except those in the user's training set. The test set has only preferred books which are considered more relevant than the unread books on the ranked-list; so, an ideal system would rank items from the test set in the top-k list of books. Similar to many related projects (Matuszyk et al., 2015; Braunhofer et al., 2015; Wibowo et al., 2018),

[3] To download: https://tinyurl.com/y998428p

we set k to 10. Three-fold cross-validation is adopted per user[4], and the results are averaged. Macro averaging is applied by first calculating the recommendation accuracy per test case and then averaging them. We measure if the differences between two versions of our models are statistically significant using t-tests at a p-value less then 0.05 or 0.01.

Metrics. We compute the recommendation accuracy by precision at k (P@k) and recall at k (R@k)—Equations 3.3-3.4—where a *relevant* book means a preferred book, and *recommended* means ranked in the top k list.

$$P@k = \frac{\#\, relevant\ books\ in\ top\ k\ recommended}{k} \tag{3.3}$$

$$R@k = \frac{\#\, relevant\ books\ in\ top\ k\ recommended}{\#\ of\ relevant\ books} \tag{3.4}$$

Baselines. We implemented multiple content-based baselines namely LDA, LSI, VSM and Doc2vec using gensim[5], a Python library. The texts of books, which have the same lengths considered in the AuthId classifier, were tokenized and down-cased, and NLTK stopwords and least frequent words were filtered out. In LDA, LSI, and VSM, relevant books from a target user training set are aggregated to create a query. In VSM, recommended books are the top-k books similar to the query according to cosine similarity. In LDA and LSI, we conducted many trials to find the best number of topics from 10, 50, 100 and 200 topics and reported the best results. A Doc2vec model was created based on the book texts, with book IDs as the labels. We experimented with multiple Doc2vec parameters, including 100 or 300 dimensions and 5 or 10 window sizes, and reported the best results. Using cosine similarity, we recommend

[4] Some baselines take a long time to complete 3-fold cross-validation; more folds would require extensive time.

[5] `https://radimrehurek.com/gensim/`

the k most similar books to the user query, which is the average of the training set's books vectors. We also compare with a plain author-based RS (a content-based system) which uses SVR similarly to our proposed system, with one difference: instead of book AuthId representations, author ids are used. For collaborative filtering, we implemented matrix factorization (MF) ranking method (Hu et al., 2008) in Graphlab[6]. MF represents users and items as vectors of latent factors which we chose empirically from 8, 16, 32 and 64 factors. Given the user binary feedback, the algorithm is fitted using logistic loss.

Parameter Settings. We stop training the author identification model at the point when the validation accuracy stops increasing for five epochs, or if validation accuracy keeps increasing while the training accuracy is becoming close to 100%. We save the model that provides the best validation accuracy and used it to generate the AuthId book representations. We used the dataset in section 3.3.1 as a training set for AuthId classifier. The validation set contains 119 books, which were written by 67 unique authors, that were randomly sampled from a collection of books that do not exist in the training set but written by the same authors where a maximum of three books per author are selected. To create the pretrained word embeddings, we trained word2vec over lowercased Gutenberg texts after eliminating NLTK stopwords and punctuations. The embeddings size and context window are set to 100 and five respectively.

The author identification network is trained using RMSprop optimizer over 32 or 16 batches. For the non-linearity, rectified linear units (ReLU) is adopted. We also experimented with various combinations of parameters: number of filters={400,500,600}, kernel size={3,5,7}, number of neurons in the first fully connected layer={100,200}. We followed Keras[7] default settings for the remaining parameters such as stride length. The step (stride) in the CNN layer

[6] https://turi.com/

[7] https://keras.io/

is set by default to one; hence, the algorithm goes over all the windows of words in the book. The best validation accuracy (31%) was achieved at the 16th epoch over mini batches of 32 samples when using 400 filters, kernel size of 7 and a dense layer of 200 dimensions. These experiments were implemented using Keras[8], a Python library, on a NVIDIA GeForce Titan X Pascal GPU with a memory of 12184 MiB.

The SVR was developed using scikit-learn with RBF kernel. The performance of SVR when predicting raw ratings (1-5 stars) is better than binary; hence, the former is reported here (see Appendix C.1 for more details). For some users who do not have negative ratings, the regressor ended up not distinguishing between relevant and irrelevant items. To solve this issue, we include in the training stage some randomly selected books not read by the target user to work as irrelevant books (excluded from baselines as well)[9]. The final number of irrelevant books in the training set is the same as the relevant ones.

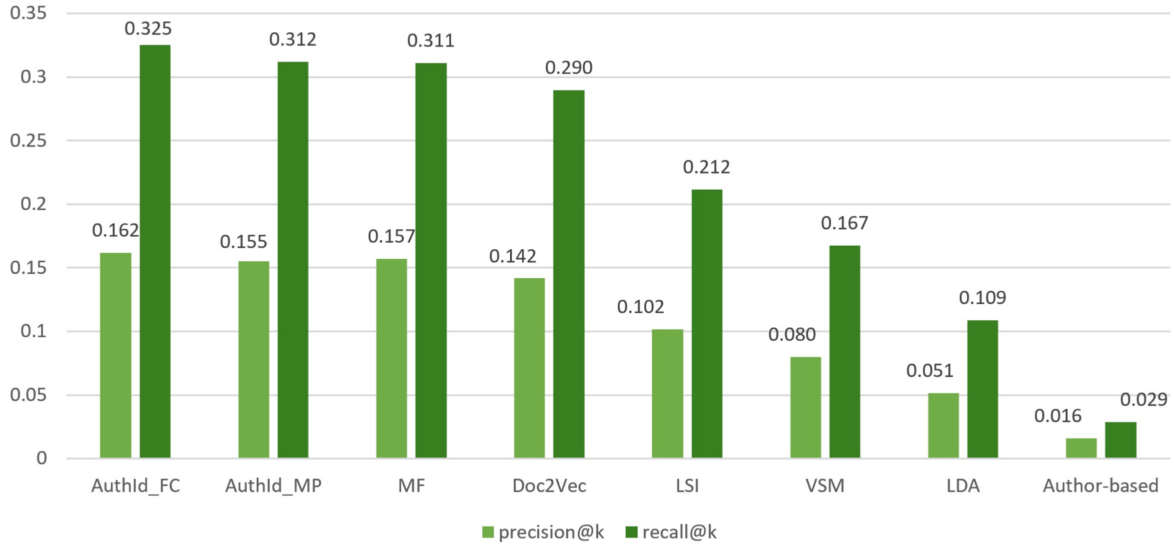
3.4 Results and Analysis

Figure 3.2 illustrates how the proposed system, whether using the fully connected layer or max pooling layer, retrieves relevant books more than the baselines, with the former achieving statistically significantly higher accuracy at a p-value of 0.01 when compared to CB baselines. Many users have fewer than ten books in their test set. This means that P@10 would never become 1, and also explains why R@10 is greater than P@10. On the other hand, some users have more than ten books in their test set, making the R@10 low even for an ideal system. The best baseline is MF, followed by Doc2vec. LDA's low accuracy is surprising, yet it is possible that more preprocessing is required for LDA to work correctly. We expected the plain

[8] `https://keras.io/`

[9] This is similar to real-world situations where some users have no negative feedback. It is expected that the performance increases if negative ratings are specified.

Figure 3.2: Precision@10 and recall@10 generated by our system (AuthId_FC and AuthId_MP), and the baselines.

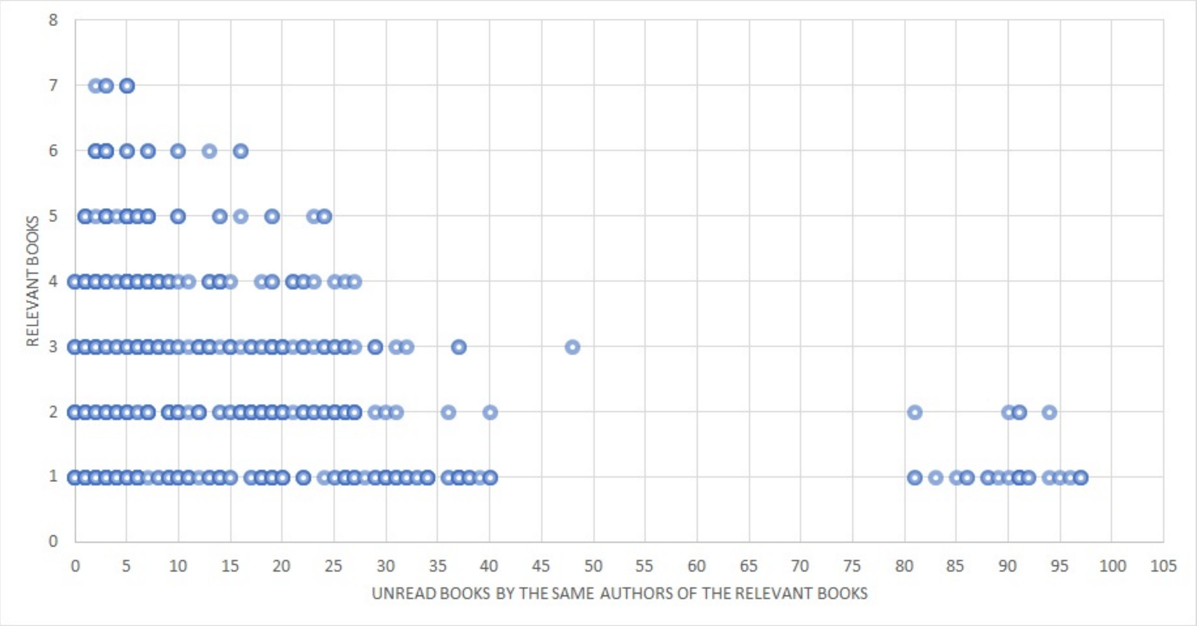


author-based system to have high R@10 by just assigning high predictions to the target user's favorite authors, but the performance is the poorest. A closer look shows that a preferred author might have many books not read by the user, and when the author-based system recommends a random sample of these books, many of them are considered irrelevant (not rated by the user).

This observation has led us to ask how many unread (unrated) books there are for the authors of books retrieved by AuthId_FC[10]. To investigate, for each user we obtain the authors of her relevant recommended books and count the unread books they wrote on the list of books to rank. This analysis is shown in Figure 3.3 where one circle refers to one test case (one fold for one user) and darker circles refer to multiple test cases. The y-axis represents the number of relevant books in the top 10 recommended list. The x-axis refers to the number of irrelevant books by the same authors who wrote the relevant books. For example, the mark at (37,3) means that our system could recommend three books relevant to a target user from a list

[10] As AuthId_FC provides the best performance, we adopted it in all the following analysis

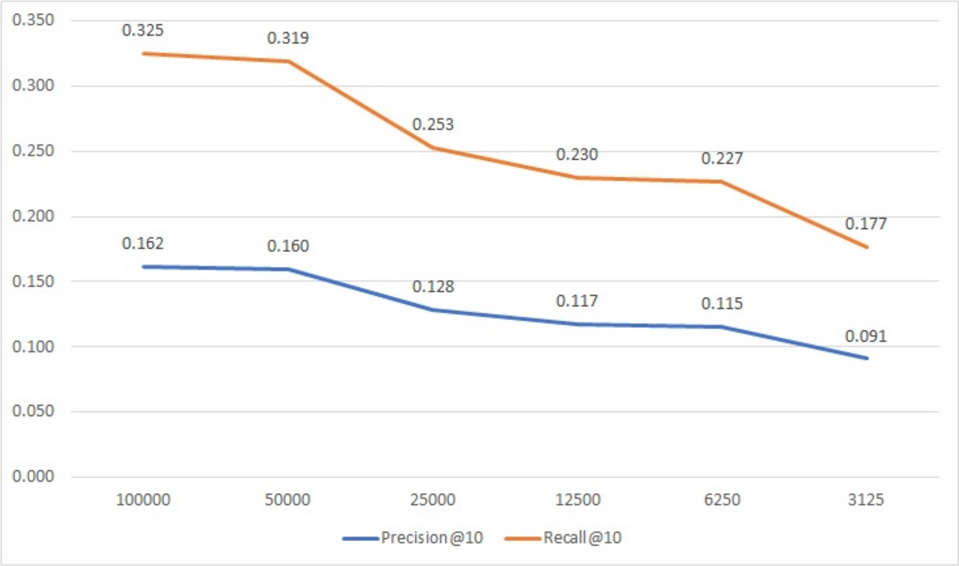
Figure 3.3: Users' relevant recommended books versus unread books by the same authors



of items containing 37 books (irrelevant to the user) written by the same authors as the three retrieved books. In 294 cases, the system could retrieve one or more relevant books from a list with more than ten irrelevant books by the same authors. In 29 test cases, the number of unread books exceeded 80.

Furthermore, in 17 test cases, the system could rank in the top ten list at least one relevant book that was written by an author who did not appear in the training set. For example, one user was recommended a relevant book "Barchester Towers" by the author "Anthony Trollope“ who did not appear in the training set, which has books authored by Elizabeth Cleghorn Gaskell, William Shakespeare, George Meredith, Honoré de Balzac, Joseph Conrad, Robert Louis Stevenson, E. Nesbit, John Buchan, Frederick Marryat, Anthony Hope, G. K. Chesterton, Alexandre Dumas, Frances Hodgson Burnett and Rafael Sabatini. This would indicate that the system could learn latent characteristics from the target user's previous preferences that allows it to discover books by authors that the user did not read.

Figure 3.4: Effect of text length on recommendation accuracy



To assess the effect of text length on the accuracy of recommendations, we developed book AuthId representations using fewer texts. We iteratively divided the length by half and fed it to the author identification model. The model was chosen after searching for the most accurate combination of a number of filters and kernel size as in section 3.3.2. Figure 3.4 shows that the shorter the text length, the less accurate the recommendations. A statistically significant difference starts to occur when using 25,000 or fewer words. It is assumed that feeding texts longer than 100,000 words would result in better performance. It is possible that longer texts would reveal more discriminative word choices, topics, etc., which distinguishes an author from another.

We went further to analyze the quality of AuthId book representations by measuring if similar representations have overlapping descriptions in NoveList Plus. Using cosine similarity, we studied the ten books most similar to “ The Fitz-Boodle Papers“ by William Makepeace Thackeray. We selected this book because NoveList has information on all its related authors.

Table 3.1 shows Gutenberg books in descending order according to their similarity values, as well as author information. Thackeray himself authored the first four books. Most of the similar books share genre or topics with the queried book. One book has a similar writing style description.

Table 3.1: Author information of books similar to “ The Fitz-Boodle Papers“

Book name (similarity)	Description on NoveList
The Fitz-Boodle Papers by William Makepeace Thackeray #2823 (1)	Author Characteristics: Male; India, England, Great Britain; British, English Genre: **Classics**; **Satirical fiction**; Literary fiction; **Historical fiction** Character: Flawed; Unlikeable; Complex Storyline: Character-driven; Intricately plotted Pace: Leisurely paced Writing Style: Richly detailed; **Descriptive**; **Witty**; Engaging Time Period: 19th century; 18th century Subject headings: Young women – England – History – 19th century; Upward mobility; **Men/women relations**; Inheritance and succession; Manipulation by women; Social status Location: England – Social life and customs – 19th century
The Book of Snobs by William Makepeace Thackeray #2686 (0.963)	
Ballads by William Makepeace Thackeray #2732 (0.957)	
Barry Lyndon by William Makepeace Thackeray #4558 (0.948)	
Waverley; Or ’Tis Sixty Years Since — by Walter Scott #4966 (0.937)	Author Characteristics: Male; Scotland, Great Britain; British, Scottish Genre: **Classics**; **Historical fiction** Time Period: Medieval period (476-1492); 17th century; 12th century; Jacobite Rebellions (1689-1746); Scottish Stewart period (1371-1603); Stuart period (1603-1714); Plantagenet period (1154-1485); 1700s (Decade) Subject headings: Richard I, King of England, 1157-1199; John, King of England, 1167-1216; Charles Edward, Prince, grandson of James II, King of England, 1720-1788; Nobility – Scotland; Romantic love; Knights and knighthood; Religion; Inheritance and succession; Jacobites; Kidnapping; Love triangles; Crusades – Third, 1189-1192 Location: Scotland – History – 18th century; Great Britain – History – Richard I, 1189-1199
Stray Pearls: Memoirs of Margaret De Ribaumont, Viscountess of Bellaise by Yonge #5708 (0.935)	Genre: Love stories; **Classics**

The Adventures of Peregrine Pickle by T. Smollett #4084 (0.934)	Author Characteristics: Male; Scotland, Great Britain; British, Scottish Genre: **Satirical fiction**; Picaresque fiction Tone: Amusing; Strong sense of place; Offbeat Writing Style: **Witty**; **Descriptive** Time Period: 1760s Subject headings: Scots; Misadventures; Stable hands; Voyages and travels; Nobility; Character; Fathers and sons; **Men/women relations**; Rescues; Human nature; Travelers Location: England – History – 18th century
Captain Blood by Rafael Sabatini #1965 (0.9328)	Author Characteristics: Male; Italy, England, Great Britain; British, English, Italian Genre: **Historical fiction**; Swashbuckling tales; Adventure stories; Sea stories Tone: Dramatic; Atmospheric Writing Style: Engaging Time Period: Revolutionary France (1789-1799); 1780s; 17th century; 16th century Subject headings: Actors and actresses; Pirates; Buccaneers; Physicians; Swordfighters; Traveling theater; Disguises; French Revolution, 1789-1799; Swordplay; Nobility; Injustice; Revenge; Class conflict; British in the Caribbean Area; Pirates – Mediterranean Region; Brothers Location: France – History – Revolution, 1789-1799; Caribbean Area
Diana of the Crossways — Complete by George Meredith #4470 (0.9325)	Genre: **Satirical fiction**; Fantasy fiction; Middle Eastern-influenced fantasy Subject headings: Egoism in men; Courtship – England; Barbers; Magical swords Location: Arab countries
Mont-Saint-Michel and Chartres by Henry Adams #4584 (0.92)	Genre: Political fiction

It is expected that the AuthId classifier learns words, topics, and other discriminative signals of the author style. However, we recognize that such a network may not capture all style elements such as the structure of sentences and paragraphs or identify figures of speech and rhythm. Other deep learning approaches such as RNN and hierarchical CNN/RNN may learn more complex style aspects. It is also worth mentioning that we tried global average pooling instead of max pooling and received acceptable accuracy in both author identification and book recommendations. Average pooling returns the mean of each feature map and does not extract an author's most distinguishing features as in max pooling. However, it was observed when conducting a qualitative analysis that max pooling helped to create AuthId book representations that have more common annotations by experts.

3.4.1 Limitations

- Authors are not always consistent in their stylistic aims and genres. One author might write essays, poems and mystery, and a user is not expected to like them all. Our AuthId classifier does not consider such differences. However, the book recommendation component tries to find patterns in the user previous reading preferences and learns what the kind of books a user likes.

- Any CB is a classifier/regressor that needs to be trained once the target user has accumulated fresh responses. Moreover, the AuthId classifier must be re-trained to generate representations for new books/authors. The process of updating both components in this system will require extended time and effort.

- In some cases, even though not very common in literary works, a book is authored by multiple people. To adapt to this situation, the authorship identification model could learn an AuthId book vector for each author and then the vectors are merged (averaged or summed). Also, when multiple authors have regular collaboration, they can be considered as one entity (one label). In our implementation, we filtered out authors with fewer than two books. If multiple authors have two or more collaborations, they would be considered as one label. More general limitations of content-based systems are discussed in section 6.2.

3.5 Conclusion and Future Work

In this chapter, we:

- Represented books as vectors learned in relation to authors and used such representations to make recommendations. To our knowledge, this is the first work that transfers the information extracted from an author-identification model to book recommendations.

- Evaluated the system according to a top-k recommendation scenario and found that the use of AuthId-book representations gives statistically significantly higher accuracy when compared with many competitive CB recommendation methodologies. Our system, as any CB, avoids several shortcomings found in CF by relying solely on the target user ratings rather than using all the community rating; yet, it results in higher accuracy.

- Showed that the longer the texts used in creating AuthId-book representations, the more accurate the recommendations. This encourages the exploitation of books' whole texts for recommendations.

- Conducted a small qualitative analysis that concluded that similar books share similar experts' annotations.

This work can be extended in the following ways:

- Author writing style may change with time or when writing in different genres. Here, we trained the model to predict the authors regardless of the genre, topics or time of their writing. Taking into consideration these factors may help develop a more accurate author-identification model, which is expected to result in better book representations. One proposed method is to use a multi-task classifier that predicts genre and author.

- In general, there is no ideal neural network architecture. There is always room for improvements. Our architecture is by no means perfect. Enhancements could be conducted by searching for the best hyperparameters, increasing the number of layers to provide a higher level representation of the text, or using different architectures. Other approaches to authorship identification should be investigated in the context of book recommendations. Using more powerful machines would be helpful for advanced implementation.

- One possible recommendation approach is to adopt multi-task learning where a neural network has two outputs, specifically author name and user preferences. We have tried to use NN models to predict user reading preferences from the text of books, but we could not achieve high accuracy in the top-k scenario. We think that performance would improve if we used a larger dataset with more user-item interactions.

- Even though we only experimented with English corpus, it would be interesting to apply AuthId method to other languages.

Chapter 4

Study of Linguistic Features Incorporated in a Literary Book Recommender System

4.1 Overview

This chapter[1] proposes a content-based recommender system that considers books' textual features. We are motivated to investigate the textual elements that may play a role in generating high-quality book recommendations. Two recommendation algorithms were trained on 120 linguistic aspects learned from book text including lexical, syntactic, stylometric and fiction-specific features. One of the two adopted recommendation methodologies ranks books using a forest of randomized regression trees, and its variables importance can provide insight into the factors that contribute to the accuracy of the recommendations. The identification of elements that contribute the most to users' book preferences may help the system owners and publishers to understand factors that appeal to individuals and communities.

It is a common practice to use linguistic features to analyze the content and style of litera-

[1] This chapter in based on (Alharthi and Inkpen, 2019)

ture. One book RS (Vaz et al., 2012a) employs limited stylometry features of document length, vocabulary richness, part-of-speech bigrams, and most frequent words. Only two out of our considered elements are similar to (Vaz et al., 2012a) namely document length and vocabulary richness. Moreover, Pera and Ng (2014a,b, 2015) used readability score to filter out books recommended for emergent readers. We cover a wide set of lexical, syntactical, character-based, fiction-specific and style-based features that, to our knowledge, were not included in book RSs before.

4.2 Methodology

4.2.1 Feature selection

We considered multiple feature sets learned from book texts, except the author's name, converted into a numerical value using one-hot encoding. Here, we exploit book texts directly instead of relying on labels tagged by experts, *e.g.*, genre. Table 4.1 depicts categories of our linguistic features classified into lexical, syntactical, character-based, fiction-specific and style-based. Appendix A.2 shows the full list of all features.

One considered feature set is Linguistic Inquiry and Word Count (LIWC) (Pennebaker et al., 2015a) which is a widespread resource that was originally developed for psychological analysis, and thus focuses on psychological in addition to grammatical and content word categories. Psychological studies have consistently shown that style (i.e., function words such as articles) is more useful than content, i.e., emotions. For example, preposition and conjunction are found to be reliable indicators of cognitive complexity. Function words may give a way to learn how (not what) people think (Pennebaker et al., 2015b). Other analysis tasks using LIWC includes authorship profiling in (Argamon et al., 2009) where an unknown writer's gender, age

Table 4.1: Categories of features included in our study

Category	**Feature Names**
Lexical	*Token-based:* average length of paragraph, sentence and words; average number of commas per sentence; average variance in paragraph length and in sentence length; length of book; words>6-letters; dictionary words; average syllables per word Lexical density (type-token ratio) *Word frequencies:* percent of latinate words; function words; affect words; social words; cognitive processes; perpetual processes; biological processes; core drives and needs; time orientation; relativity; personal concerns[2]
Character-based	Numbers; all punctuation
Syntactic	Percentage of adjectives, adverbs, nouns, and verbs; comparatives, interrogatives; quantifiers
Fiction-based	Percentage of dialog text; the number of dialogs; the average length of dialog texts; the number of unique fictional characters; the number of times fictional characters are mentioned; the number of unique places; the number of times places are mentioned; Percentage of dialogue by female characters; Percentage of characters which are female
Six-styles	Literary; abstract; objective; colloquial; concrete; subjective

and personality are identified from a given text. Another study explored differences in the way men and women's writings (Newman et al., 2008).

Moreover, a quantitative analysis of literary genre profiling is conducted in (Nicholsa et al., 2014) which tests seven directional hypotheses about the differences in content and style of science fiction, mystery and fantasy. Measurements of content and style of carefully selected literary books are conducted using seven LIWC categories. It is found that compared to mystery, science fiction has more cognition and religious terms and less social words, pronouns and auxiliary verbs. Compared to fantasy, science fiction has a higher rate of cognition word and auxiliary verbs*[3], and a lower rate of social, biological* and religious terms*. Also, fantasy has significantly more biological* and religious* term than mystery. One of the statistically confirmed hypotheses is that fantasy has fewer auxiliary verbs, which indicate mood, voice and modality, due to its focus on the story settings rather than actions. Another confirmed hypothesis is that fantasy, which describes multiple social aspects including creatures' beliefs, has more religious terms than science fiction, which is more into science and secular aspects. Also, an established assumption is that fantasy has a high rate of biological terms because imaginary creatures are described physically (*e.g.*, "She has red hair"). The previous studies encourage us to apply content and style word classes to make book recommendations. Many LIWC dimensions can be categorized under the reader's advisory appeal factors. Dimensions that would relate to a book's *mood/tone* are emotional tone, affect words, sentiment and subjectivity. Also, time orientation and relativity may capture some aspects of the book *settings*.

We computed all categories using the LIWC 2015 dictionary, which contains more than 6,400 terms including word stems and emoticons. Each of the 92 categories has associated sub-dictionary. Whereas many LIWC classes simply reflect their names (*e.g.*, articles dictionary has only three words "a", "an" and "the"), other classes are constructed by experts (*e.g.*, the

[3] * indicates statistically significant difference

emotion category). An agreement of two out of three judges is needed to add or remove a word from a dictionary. A word may belong to multiple dictionaries such as sadness and emotions. The value of a category is the percentage of its words in the book (i.e., 4.54% of words in a book are auxiliary verbs) (Tausczik and Pennebaker, 2010). Table A.1 shows the full list of LIWC dimensions supported by examples.

In authorship analysis, it is assumed that writers have a limited vocabulary, certain diction and syntax complexity. However, an author's writing style may differ according to the target readers and genre. Researchers have considered some standard text measurements to analyze author styles. One measure is word length, which, for example, helped Brinegar (1963) successfully demonstrates that Mark Twain did not author a certain book (i.e., The Quintus Curtius Snodgrass Letters). Yet, it is noticed by Smith (1983) that the texts by two contemporaneous authors have almost identical word-length distributions and that it is difficult to identify one author's writings across genres or eras (Holmes, 1985). Another feature, the mean length of sentences, which is debatable stylometric measurement adopted by Smith (1888) and Sherman (1893) who argue its utility to detect style variations over time (Holmes, 1992). Others consider it to be not very valuable in authorship detection (Yule, 1939). Moreover, the average number of syllables per word as well as other measurements helped Fucks (1952) identify unique characteristics of authors (Holmes, 1985). All text measurements are calculated by GutenTag (Brooke et al., 2015), which uses NLTK for sentence and word tokenization, and the average number of syllables per word is computed by pyphen[4].

Other measurements of writing style are also computed by GutenTag (Brooke et al., 2015) which, similarly to our dataset, was built on the Project Gutenberg texts. GutenTag's built-in tagger uses a stylistic lexicon (Brooke and Hirst, 2013, 2014) to calculate stylistic aspects usually considered when analyzing English literature. Text styles are not exclusively meant to

[4] `http://pyphen.org/`

reflect aesthetic characteristics, but to represent generic differences of texts such as genres. The styles are colloquial (informal) vs. literary (traditional), concrete (related to physical objects) vs. abstract (related to complex concepts), subjective (influenced by feelings) vs. objective (not affected by feelings) and polarity (Brooke et al., 2017; Brooke and Hirst, 2013).

The stylistic lexicon is automatically created by first using 829 seed words related to each defined style, then discovering and refining the terms that co-occur with them in a big corpus (Gutenberg dataset). The adopted six styles model considers not only n-grams but also phrases. It only analyzes the style at the lexical level instead of syntactic. A word may belong to multiple stylistic categories. Evaluation of this approach resulted in more than 90% accuracy for all categories except subjectivity (85%). In the model, each word is represented as a vector of six values associated with the styles. To create a 'stylistic profile' of a given text, the vectors of its words (only types of words not tokens) are averaged (Brooke et al., 2017). The six styles model helped in some literary analysis; it is found in (Brooke et al., 2017) that free indirect discourse is quantitively distinguished from and centered between direct discourse and narration. It is also found, in close study of one of Virginia Woolf's book, that the feminist author indeed defied the traditional image of women by expressing their indirect discourse in a language which is more objective and abstract and less subjective and colloquial than men's language.

Moreover, similar to other book RSs (Pera and Ng, 2014a,b, 2015, 2013) which were directed at young readers, we incorporate a text readability measurement. We calculated the Flesch reading ease score using a formula that measures the length of words and sentences using textstat[5]. Its score comes in a range of 100 (very easy) to zero (very hard).

In addition, some elements specific to fictional works are also considered. As Bakhtin suggests (Bakhtin, 1981), the number of fictional characters can provide information about the

[5] http://neon.niederlandistik.fu-berlin.de/en/textstat/

type of book. A small number of characters could indicate that the story takes place in a rural setting where characters typically know one another (*e.g.*, family), while many characters suggest urban settings where interaction with unknown people (*e.g.*, in a theatre) occur (Elson et al., 2010). To identify characters, we used the fiction-aware named entity recognizer (LitNER)(Brooke et al., 2016), which outperforms other popular POS taggers. We counted the number of characters in a book, and how many times they were mentioned. Since LitNER also recognizes locations, we counted the number of places in a novel as well. To identify characters' gender, Gutentag considers how a character is referred to (e.g., her and miss) and compares the character's first name to a list of female and male names. We calculated the *number of fictional characters*, *the number of their occurrences*, *number of dialogues*, *average dialogue length*, *percentage of female characters and percentage of their dialogues*. Such features may reflect the appeal factor, *characterization*. Other measurements such as *number of places in a book* and *number of times places are mentioned* may describe the *setting* factor.

In this study, we encompassed numerical characteristics that may play a role in reading preferences. When selecting the linguistic features, we rely on previous studies about factors related to reading preferences such as book length (Hussain and Munshi, 2011) and readability (Pera and Ng, 2014a), on the features' usability in literary analysis of content and style such as LIWC (Nicholsa et al., 2014) and six-styles (Brooke et al., 2017) and on the availability of text measurements within a utilized package such as *percentage of female characters* which is calculated via GutenTag (Brooke et al., 2015). Even though we consider some elements to be related to reader's advisory appeal factors, it is not our aim to model books according to the appeal factors, which would require knowledge and expertise of professionals.

4.2.2 Recommendation procedure

In content-based recommendations, for each user, a classifier or regressor is created to learn from previous preferences to predict future interests. In this section, we describe two approaches which empirically provided the best results.

K-nearest neighbors (KNN) is commonly used in RSs which, in our case, takes the input of books represented as vectors of features and finds the most similar ones. A predicted value for book *x*, which is unrated by user *u*, is computed as the average of similarity values between *x* and *K* books preferred by *u*. The similarity between two items is first calculated at the feature level, then averaged (equation 4.1, where *m* is the number of features). Books with highest similarity values are recommended in the top *k* list (Lops et al., 2011). We experimented with multiple similarity measures, and a combination of Gaussian kernel distance for numerical features (equation 4.2, where σ^2 is a constant (Phillips and Venkatasubramanian, 2010)) with cosine distance for categorical features (equation 4.3, where x_i and y_i are binary vectors) gives the best results. The similarity ranges from zero to one with one being the highest.

$$Overall\_similarity(x, y) = 1 - \frac{\sum_{i=1}^{m} sim(x_i, y_i)}{m} \tag{4.1}$$

$$Gaussian\_kernel\_distance(x_i, y_i) = 1 - exp\left(-\frac{\|x_i - y_i\|^2}{2\sigma^2}\right) \tag{4.2}$$

$$Cosine\_distance(x_i, y_i) = 1 - \frac{x_i \cdot y_i}{\|x_i\| * \|y_i\|} \tag{4.3}$$

The other approach constructs for each user a forest of randomized regression trees. A single tree-based model (Breiman et al., 1984) uses the training sample of size N, where each input X is a vector $(X_1, X_2, \ldots .X_P)$ labeled with output Y, to build a tree T with internal and external nodes (leaves) representing features and labels respectively. The feature space χ

is divided at the tree root into sub-regions which is recursively divided at each node t. At an internal node *t*, a test $s_t = (X_m < c)$ is performed to further split its sub-region into t_L and t_R[6]. In regression trees, an external node t is labeled with $\bar{y}(t)$, which is the average output in t subset (Louppe et al., 2013).

To ensure sub-regions with as homogeneous output as possible, the algorithm finds the best split which maximizes the impurity reduction $i(t)$ measured when splitting Nt (samples at node t) into two partitions t_L and t_R where $p_L = N_{tL}/N_t$ and $p_R = N_{tR}/N_t$. In equation 4.4, $i(t)$ refers to any impurity reduction such as variance in regression trees.

$$\triangle\, i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R) \tag{4.4}$$

The growth of a tree is stopped once a stopping criterion is met. After the model is constructed, a new sample can be propagated through the tree and eventually labeled with $\hat{y}$, the value on the leaf it ends up at (Louppe et al., 2013).

Ensemble approaches, such as randomization-based algorithms, are adopted to enhance the performance of single trees that may have low accuracy due to high variance. Random Forests (RF) (Breiman, 2001), a popular extension of tree-based models, constructs trees based on multiple bootstrapped copies, and considers a random subset of some features at each node when deciding the best split. Another ensemble method is Extremely Randomized Trees, also called Extra Trees (ET), (Geurts et al., 2006), which provides competitive results in a learning-to-rank task in (Geurts and Louppe, 2011). ET generates multiple trees from the whole training set, with no bootstrapping required. When performing a split, ET also considers a random subset of features. The smaller the number of features to consider at each node, the more the randomization and the less the dependence on the target value (i.e., when it equals one, the trees are completely randomized). ET takes the randomness a step further by choosing the split

[6] In case of binary test.

point at each internal node at random without considering the output. Using random splitting decreases the chance of overfitting (high variance), and the use of the whole training sample can reduce underfitting (bias) (Louppe et al., 2013; Geurts and Louppe, 2011; Geurts et al., 2006).

In addition to their high accuracy, forests of randomized trees can provide insight regarding the reasons for a good performance by identifying key variables for predicting the output. One way to calculate the importance of a feature X_m is through Mean Decrease Impurity importance (MDI), as in equation 4.5. In the context of RF and ET, the average of MDI for all N_T trees is used (Louppe et al., 2013).

$$imp(X_m) = \frac{1}{N_T} \sum_{T} \sum_{t \in T : v(s_t) = X_m} p(t) \bigtriangleup i(s_t, t) \tag{4.5}$$

$p(t) \bigtriangleup i(s_t, t)$ refers to the weighted impurity reduction for all internal nodes t where a feature X_m appears, $p(t)$ is $\frac{N_t}{N}$ and $v(s_t)$ is the relevant feature.

4.3 Evaluation

4.3.1 Dataset and preprocessing

Similar to the previous chapter, experiments were conducted on a the Litrec dataset (Vaz et al., 2012c); however, more users, books and authors are included because there is no need to filter out authors with no enough items to train AuthId classifier. Also, to calculate the fiction-based features, any book without characters was excluded (only 71 titles). We used Gutentag tool (Brooke et al., 2015) to generate clean book texts as well as texts tagged with characters, places and dialogs. Similar to many CB related works, such as (Wang et al., 2009), we filtered out any user with less than ten ratings to provide adequate data for the CB. The remaining number of users was 367, and they rated 1,050 unique books authored by 410 writers. We utilized books' whole texts with an average length of 91580.07[7].

Baselines. The same algorithms as those in section 3.3.2 are used. However, to report their best results, we ran them again to choose the best parameters from: number of factors in MF, number of topics in LDA and LSI and context and vector sizes in doc2vec.

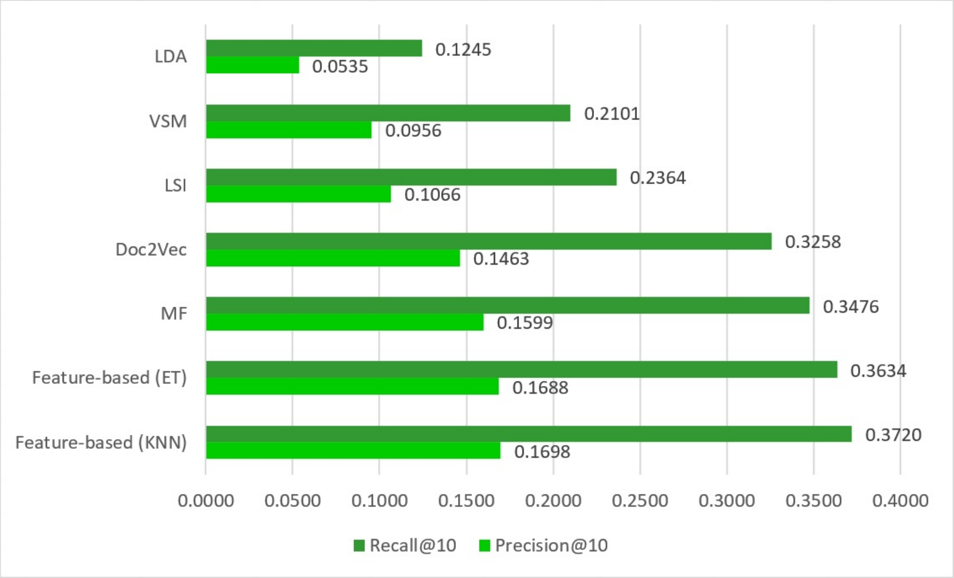
Parameter Settings. To search for the best parameters, we built multiple feature-based systems using a validation set of randomly selected thirty users. We experimented with 1, 5, 15, 30 and 64 neighbors in KNN, and 15 led to the best accuracy. For ET and RF, we tried 500, 1000, 1500, 2000 and 3000 trees, and 120, 60, 10, 7 and 3 features to consider at a split. It is observed that a smaller number of features leads to better accuracy and found that the best performance was recorded at 1000 and 3 for ET, and 1500 and 10 for RF. As it has non-competitive performance on the validation and test sets, we dropped RF from all the next results, and since many users lacked negative feedback, we randomly included in the ET training set irrelevant books for the target user equal to the relevant ones (excluded from other algorithms as well). Prior to applying KNN, Z-score normalization is performed in order to transform each feature independently to a scale of zero average and one standard deviation. The systems were implemented using scikit-learn[8] and Graphlab[9]. In the current experiments, the average number of books in the training set is 9.2; as a result, the average number of books to rank is about 1,040.8 (1,050 unique titles minus 9.2).

[7] To download the calculated features `https://tinyurl.com/y9svvgwg`

[8] `http://scikit-learn.org/stable/index.html`

[9] `https://turi.com/`

Figure 4.1: Recommendation accuracy of our approach against the baselines



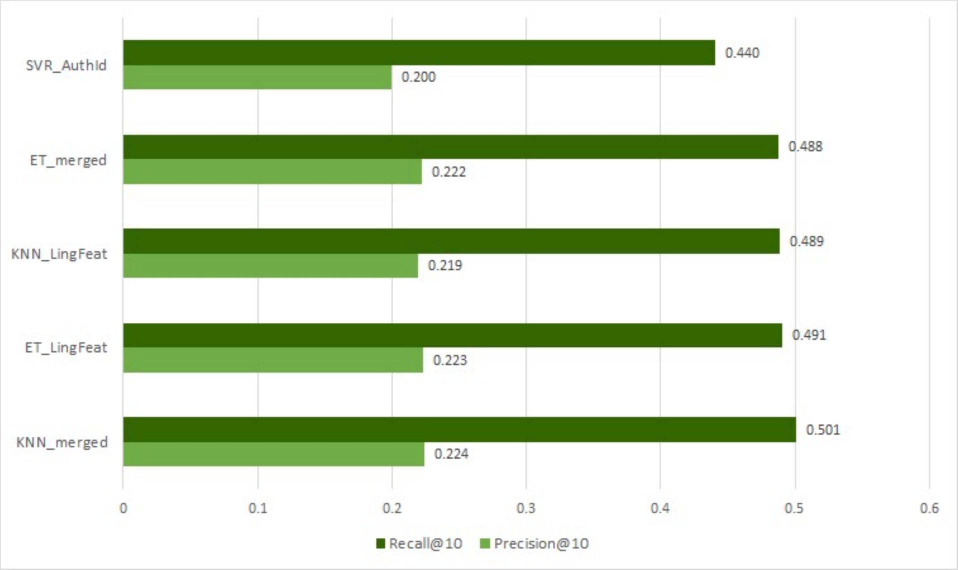
4.4 Results and Analysis

More than a third of the relevant books in the test sets are ranked in the top ten recommended list generated by each feature-based system (Figure4.1). Both KNN and ET algorithms generated statistically significantly more accurate top ten recommendation lists compared to the CB baselines, for which KNN was the most accurate. Also, there is a statistically significant difference when comparing KNN-based CB to MF. All content-based baselines are statistically significantly lower than MF, with LDA being the lowest recommendation algorithm.

The nature of KNN makes it an appropriate choice to learn short-term interests such as recently-read articles which is studied in (Billsus and Pazzani, 1999). If the user likes a new item of a different type, KNN can easily find similar items. On the other hand, other machine learning algorithms may need more than a few training instances to recognize a new pattern. The number of training examples in our dataset has an average of 9.2 which may explain the superiority of KNN over ET.

It is worth mentioning that recall@10 cannot equal one in the case of users with more than

Figure 4.2: Comparison of linguistic features vs. AuthId book representations



ten books in their test set, and precision@10 can only be ideal if the target user has ten or more books in their test set, which was is not the case for many users in our dataset. This also explains why precision@10 is lower than recall@10.

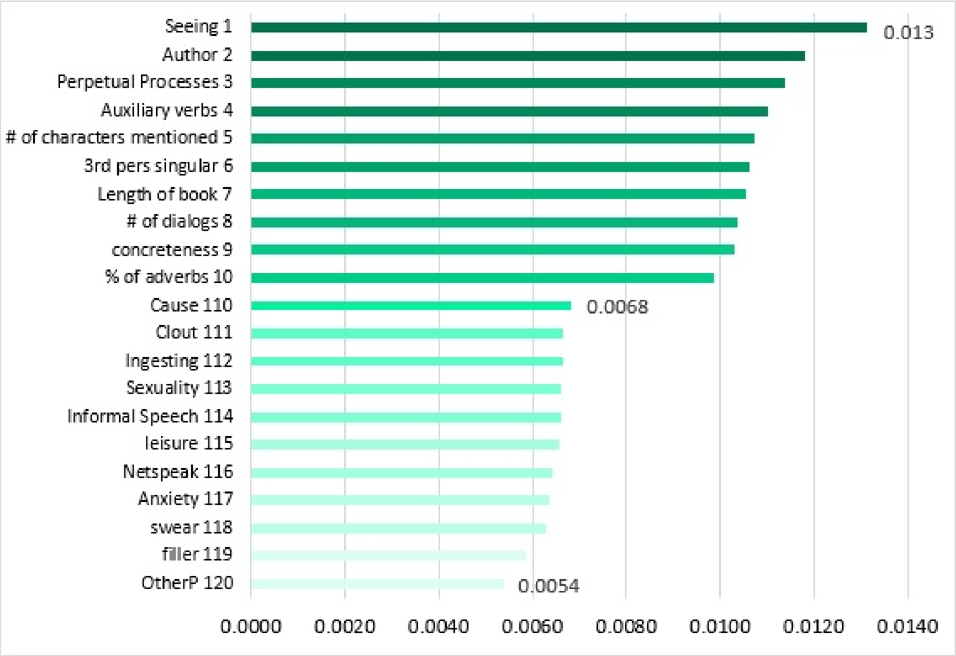
We also compared with AuthId approach proposed in chapter 3. We removed any book with no AuthId representation and any user with less than ten items. The number of remaining users is 264, and the number of unique items is 639. We also tried concatenating our 120 linguistic features (lingFeat) with AuthId book representations. As plotted in figure 4.2, both recall and precision of lingFeat approaches are statistically significantly higher than SVR_AuthId. The use of the merged book representations has increased KNN and slightly decreased ET[10]. However, AuthId book representations are based on 100,000 words and it was noticed that the more texts included in the creation of AuthId representations, the higher the system's accuracy. Since our linguistic features are based on word counting, it is not meaningful to measure frequencies of content and style words based on book excerpts (i.e., 100,000 words).

[10] SVR gives low accuracy when dealing with linguistic features, and ET and KNN give a competitive performance (yet less than SVR) when working with AuthId representations

One advantage of using ET is learning what key features supported the prediction of the model. In the following analysis, we consider only 730 ET models that resulted in accurate recommendations of at least one relevant retrieved book; hence, they could learn what the target user likes/dislike in term of linguistic aspects. The variable importance values of these ET models were then averaged which should smooth any bias (*e.g.*, a user who dramatically likes books with a high *percentage of female characters*). The randomization in ET reduces the chance that choosing one predictor will significantly suppress the importance values of its correlated features. The most and least influential variables are presented in Figure 4.3 and the full list is presented in A.2. The top two variables are *Seeing*, which indicates an act of observation such as view, and *author name*. While the latter is expected to be ranked highly, the former is not. *Seeing* is strongly correlated (using Spearman's rank correlation coefficient) with other top-ranked features, including *concreteness* (0.59) and *perpetual processes* (0.82), which is composed of *Seeing* words plus *Hearing* and *Feeling*. Many studies, including (Holcomb et al., 1999) show that concrete language, which is ranked higher than and negatively correlated with abstract (-0.79), is easier and more precise to process than abstract language. This is explained by the dual-coding theory indicating that while abstract words are only processed using a human's verbal system, concrete words use the same system in addition to the non-verbal one (Holcomb et al., 1999). The full list of highly correlated variables is in appendix A.3.

Similar to the findings by (Hussain and Munshi, 2011), the *Length of book* has a strong impact on users' reading preferences. Two characterization measurements are also important: *the number of characters mentioned* and *the number of dialogs*, both of which are positively correlated with the *length of book*. In general, all characterization and six-style features are placed above the 60^{th} rank (i.e., the first half of the feature list). One study (C. Beyard-Tyler and J. Sullivan, 1980) showed that female and male students preferred story summaries with a

Figure 4.3: Most and least influential variables in accurate ET models



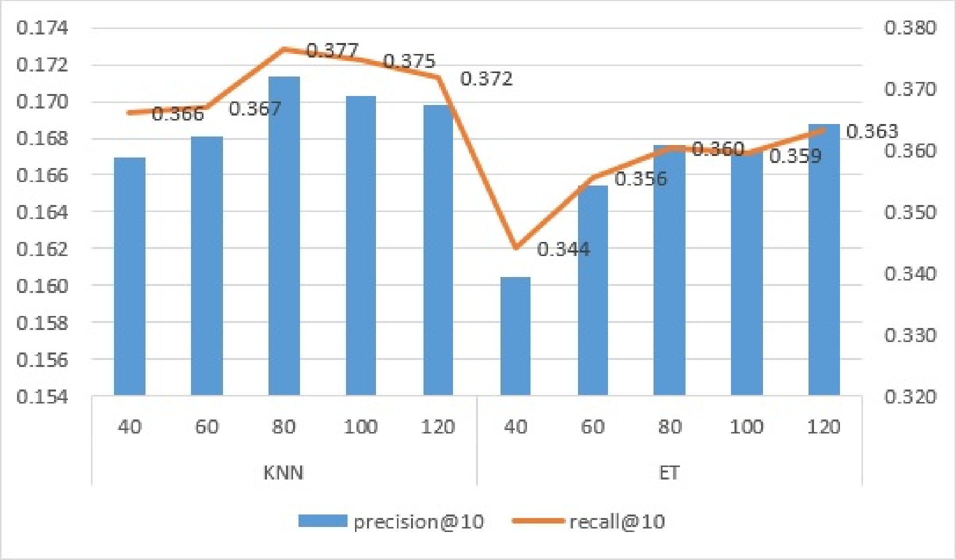
main character similar to their gender; this preference diminished when female readers' grade increased.

While *the percent of adverbs* is one of the top predictors, *the percent of nouns* and *adjectives* are ranked low, at 98 and 93, respectively. From the punctuation list *period* is best at the 20^{th} rank, while the lowest is *other-punctuations* which counts only uncommon punctuation and ASCII characters. All informal language indicators, such as *Filler, Swear, Netspeak* and *Informal speech*, are ranked at the bottom. This is expected, as classic Gutenberg literary books do not typically include modern informal words; thus, such findings could differ in other datasets. It is also worth mentioning that *colloquial*, which measures the casual language of literary texts (*e.g.*, dude), is the 46th most important feature.

It would be interesting to use the previous list of top variables to generate recommendations using KNN and ET. Figure 4.4 shows that the more features ET is fed, the higher its accuracy[11].

[11] We experimented with many feature selection methods such as chi^2, but they did not result in competitive

Figure 4.4: The performance of ET and KNN when only top features are considered



This indicates that the least important features are not entirely irrelevant to ET predictions. This is not the case in terms of KNN with increasing performance after the removal of twenty and forty least significant features.

Further analysis was conducted to determine the best and worst features for the majority of users. We counted the number of test cases (ET models) where a feature is considered the most or least influential and presented the most frequent in Table 4.2, which overlaps the previous figure list but at different ranking. Some variables, such as *length of book*, did not appear in Table 4.2; though they seemed to have high significance for many users, they were not the highest. However, the top feature, *drives and needs*, is not in Figure 4.3 but is ranked 18th. It counts words that describe *affiliation* (*e.g.*, ally or friend), *achievement* (*e.g.*, winning), *power* (*e.g.*, bully), *rewards* (*e.g.*, prize) and *risks* (*e.g.*, danger).

The average, maximum and minimum *drives and needs* and *seeing* values are 5.8, 12.9 and 3.42 and 1.51, 4.7 and 0.53, respectively. Of 301 relevant books for the twenty users with *drives and needs* as the best feature, only 26 have a value higher than the average, suggesting

results

Table 4.2: Most and least important features according to ET model

Top Feature	**Number of test cases**	**Last Feature**	**Number of test cases**
Core drives and needs, Percent of text adverbs, objective, author	25-31	Fillers	151
Seeing, Feeling	20-25	Other punctuation	45
Number of characters mentioned, Dictionary words, Period, Auxiliary verbs	15-20	Swear, Netspeak	15-25
Number of dialogs, Female referents, Discrepancies, Future focus, Abstract, Negations, Perpetual Processes, Semicolons, 3rd pers singular, Subjective, Power, Reward focus, motion, Parentheses	10-15	Friends, Sexuality, Tentativeness, Percent of characters which are female, 2nd person, Achievement, cause, home	10-15

that these users tend to prefer books with less focus on *drives and needs*. Twenty users with *seeing* as the top feature have 103 relevant books out of 253, and 18 of 22 irrelevant books with *seeing* values higher than the average. It seems that these users prefer to read books with smaller *seeing* values.

A qualitative analysis was conducted by assessing if books considered similar by the KNN system share similar authors or descriptions on NoveList. We represented books as vectors of all features except for *author name*. All the top ten books similar to a randomly selected book by Anthony Trollope were written by him as well, and eight of the top ten books by Honore de Balzac are similar to one of his books. Table 4.3 shows the books similar to *Madame de Treymes* by Edith Wharton. Only the first two books were written by her, though all the retrieved books have at least one shared description (highlighted in bold) with the queried book. Most have a topical resemblance; however, only two share the same storyline and character type. To determine whether fiction-specific features affect the retrieval of two books by Henry James, a reevaluation was done after removing them. One of his books (*Sir Dominick Ferrand*) was not retrieved, and one book by Emile Gaboriau, which has nothing in common with the queried book, was added.

4.5 Conclusion and Future Work

In this chapter, we:

- Take advantage of book texts to learn their style and content to predict future reading preferences. In the literature of book RSs, we are aware of (Pera and Ng, 2014a,b, 2015) that include a readability score and (Vaz et al., 2012a) that studies a few stylometric features, out of which two are included in our study (document length and vocabulary richness). Our investigation covered a wide variety of style and content features that were not studied before.

- Designed two recommendation algorithms that outperformed several CB systems and a collaborative filtering system in the top k recommendation scenario.
- Compared the use of linguistic features to AuthId approach and tested their joint book representations.
- Studied the effect of multiple linguistic elements on reading preferences. Particularly, we highlighted the variables considered important using an ensemble of randomized trees and produced book suggestions based on them.
- Conducted further qualitative analysis which showed that similar books generated by our system share the same author or similar descriptions on NoveList.

This work can be extended in the following ways:

- Word disambiguation could be conducted before word counting to precisely calculate word categories which may lead to better overall accuracy.
- Other text measurements that require further preprocessing, such as the number of characters in each scene, could also be applied. Moreover, semantic aspects of books content could be considered.
- Due to copyrights issues, texts of books other than Gutenberg texts were not accessible for this study. It would be interesting to conduct similar research on books from various times and different genres.
- Sequential patterns of words and POS tags could be discovered at the author/user level.

Table 4.3: Top ten similar books to Madame de Treymes by Edith Wharton

Book name (similarity)	**Description on NoveList**
1- Madame de Treymes by Edith Wharton (1) 2- The Descent of Man and Other Stories by Edith Wharton (0.72) 3- The Glimpses of the Moon by Edith Wharton (0.71)	*Genre:* **Love stories**, Modern classics; *Character*: **Complex**; *Storyline*: **Character-driven** ; *Writing Style*: Descriptive, Lush, Richly detailed; *Time Period:* 1920s; *Location*: Europe – Social life and customs – 20th century; *Subject headings*: Social status, Freeloaders, Americans in Europe, Honeymoons, **Married people**, Misunderstanding, **Men/women relations**
4- Our Friend the Charlatan by George Gissing (0.71) 5- Denzil Quarrier by George Gissing (0.698)	*Genre*: Psychological fiction, Domestic fiction, **Love stories**, Historical fiction, Satirical fiction; *Time Period*: 19th century; *Subject headings*: Women – Employment - England, Middle class women - England, Women England, Single women - England, Sisters - England, Married women - England, Self-discovery in women, Authors, Fiction writing, **Married people**; *Location*: London, England
6- A Daughter of To-Day by Sara Jeannette Duncan (0.697)	*Genre*: Canadian fiction; *Time Period*: 20th century; *Subject headings*: Brothers and sisters, Politicians, Romantic love, Clergy, Political science, **Men/women relations** ; *Location*: Ontario
7- In the Year of Jubilee by Gissing (0.69)	See above
8- The Tragic Muse by Henry James (0.687) 9- Sir Dominick Ferrand by Henry James (0.682)	*Genre*: Classics, Psychological fiction; *Character*: **Complex**, Flawed; *Storyline*: **Character-driven**; *Pace*: Leisurely paced; *Tone*: Melancholy, Reflective, Thought-provoking; *Writing Style*: Stylistically complex
10- The Tragic Bride by Francis Brett Young (0.678)	Genre: Domestic fiction *Subject headings*: Young women, Quests, Love, **Men/women relations**, Marriage, Remarriage, Self-discovery in women, Happiness in women; *Location*: Midlands, England

Chapter 5

Topic Model-Based Book Recommendation Component

5.1 Overview

Both CB approaches in chapters 3 and 4, like any content-based or collaborative filtering system, suffer from the *new user* issue. It is difficult for RSs to generate accurate recommendation lists without having the rating history of the target user. Real-world RSs adopt the ask-to-rate technique by presenting a list of items to new users and asking them to rate what they already purchased/consumed, until sufficiently many ratings have been acquired. As the signup process can be lengthy and consume much of users' time and effort, many ways were proposed to reduce users' effort including (Rashid et al., 2008). One solution to alleviate such a burden is to exploit social media, where users willingly share their opinions and interests.

This chapter[1] proposes a recommendation component that learns the users' interests from social media data and recommends books accordingly. This automatic personalization module,

[1] This chapter in based on (Alharthi et al., 2017)

which we call Topic Model-Based book recommendation component (TMB), can help existing RSs deal with new users with no rating history. We believe this is the first book RS that uses social media rather than book-cataloguing Web sites, as well as the first to extract user-discussed subjects from social media and match them with books.

For each user, a topic profile is created that summarizes subjects discussed on her social media account. Our method for modelling users' interests acquires a user's distinctive topics using *tf-idf* and represents them as word embeddings. User profiles are matched with descriptions of books, and the most similar ones are suggested. To evaluate TMB, a dataset was collected that encompasses user profiles on Twitter and Goodreads[2], a social book cataloging Web site. Even though the system is designed to complement other systems, we evaluated it against a traditional content-based RS that relies on books' metadata. We compared the top k recommendations made by TMB and CB. Both retrieved a comparable number of books, even though CB relied on users' rating history while TMB only needed their social profiles. So, TMB's new user would receive recommendations as accurate as current users in CB.

5.2 Topic Model-Based Book Recommender System

This section explains the TMB components, book and user profiles, and formally defines the recommendation process.

5.2.1 Book and user profiles

A book profile (BP) is represented as a vector of terms comprising its description. We used short descriptions of books available online. On the other hand, a user profile (UP) is a vector that consists of terms extracted from the target user's Twitter timeline. Terms are elicited from

[2] https://www.goodreads.com/

the textual content of tweets and their embedded links. Retweets and replies are included with tweets so as to avoid sparsity, while hashtags are counted in if they are spelled correctly. User profiles are built automatically using topic modelling techniques without being mapped to an external ontology or to predefined categories. For topic modelling, we considered two techniques: Term Frequency - Inverse Document Frequency (*tf-idf*) and Non-Negative Matrix Factorization (NMF). Preliminary tests showed that LDA resulted in poor performance generating more inconsistent topics (*e.g.*, cricket, test, essay, history, memory) whereas NMF produced more coherent topics (*e.g.*, cricket, test, India, literature, player); therefore, we dropped LDA from the following comparisons.

Term Frequency - Inverse Document Frequency

The *tf-idf* weighting approach is widely used in information retrieval. Term frequency ($tf_{t,d}$) of a term t is the number of times it occurs in document d. A document in this context is all tweets and/or links in one user's timeline. Inverse document frequency (Equation 5.1) helps distinguish the terms that are specific to a user/document.

$$idf_t = log\frac{N}{df_t} \tag{5.1}$$

N is the number of users and df_t is the number of documents where term t occurs. Equation 5.2 defines the *tf-idf* weight of term t in document d.

$$tf\text{-}idf_{t,d} = tf_{t,d} * idf_t \tag{5.2}$$

The terms with highest weights are considered the *tf-idf* topic model (Manning et al., 2008).

Non-Negative Matrix Factorization

This dimensionality reduction and topic modelling technique has been found to work well with short text (Cheng et al., 2013; Yan et al., 2012; Godfrey et al., 2014); hence, we adopt it as a baseline. For a user, a term-document matrix is created; a document here is one tweet or link. NMF factorizes the $m \times n$ term-document matrix A into two non-negative matrices W and H. The former represents the term-topic matrix $m \times k$, whereas the latter is the topic-document matrix $k \times n$. The number of NMF topics k should be defined ahead of decomposition. The matrix *WH* approximates the original matrix A. Every document in *WH* represents a linear combination of k topic vectors in W with coefficients given by H (Godfrey et al., 2014).

Topic embeddings

Pre-trained word embeddings increased the performance of many applications and are commonly exploited for augmentation. In TMB, we map terms in book and user profiles to pre-trained word embeddings developed using word2vec (see section 2.6) on the Google News dataset of around 100 billion words. It comprises vectors of 300 dimensions for 3 million words and phrases[3]. Other available pre-trained models (*e.g.*, Global Vectors for Word Representation[4]) have been built using text from Twitter (1.2M vocab), Common Crawl (1.9M vocab) and Wikipedia 2014 (400K vocab). The Google news model covers larger vocabulary (3M) and its embeddings are more relevant to both formal books descriptions and casual tweets.

[3] `https://code.google.com/archive/p/word2vec/`

[4] `http://nlp.stanford.edu/projects/glove/`

5.2.2 The recommendation procedure

Let $U = u_1, u_2, \ldots u_n$ be a set of Twitter users. For user u_i, a time threshold T_{u_i} is established to avoid the overlap in learning and prediction times. The learning timeframe LT_{u_i} involves all tweets and links created by u_i before T_{u_i}, whereas the recommendations timeframe RT_{u_i} contains books read by u_i after T_{u_i}. For user $u_i \in U$, a user profile UP_{u_i} is a vector comprising terms $w_1, w_2, \ldots w_m$ extracted from tweets or links shared by u_i during LT_{u_i}. Let $B_{u_i} = b_1, b_2, \ldots b_l$ be the set of books read by user u_i during RT_{u_i}. For book $b_j \in B_{u_i}$, the book profile BP_j is a vector of words $w_1, w_2, \ldots w_h$ found in b_j's description. To recommend books to u_i, TMB calculates the cosine similarity (Equation 5.3) between UP_{u_i} and BP for every book in B_{u_i}, and suggests the books with k most similar BP.

$$similarity = \frac{UP_{u_i} \cdot BP_j}{\|UP_{u_i}\| * \|BP_j\|} \tag{5.3}$$

If terms are replaced by their word embeddings, an average vector is created for word vectors in UP_{u_i} and another for BP_j. Then, cosine similarity is performed between the resulting average vectors.

5.3 Data Preparation and System Implementation

This section describes how the dataset was collected and preprocessed. It also presents the implementation of the system, unfolding technical details of the creation of book and user profiles.

5.3.1 Data collection

We collected user data from Goodreads and Twitter because there are no datasets with both users social profiles, their reading lists and book information [5]. In Goodreads, once a user finishes reading, she can write a review and share it with her followers on other social media Web sites including Twitter. A Goodreads review shared on Twitter has the default format "(1-5) of 5 stars to book name by author name http://link–to–review", so finding users with Twitter and Goodreads accounts becomes an easy task.

The Twitter API was queried to retrieve any review shared by Goodreads users, and more than 1,000 tweets were found, from which we accessed their authors and IDs. Twitter API allows the collection of a maximum of 3,500 tweets per user. We gathered text, ID, and date of creations of tweets for users with Goodreads reviews. Links were extracted from user timelines and their textual contents (if any) were collected. This was achieved by applying an efficient Python library called Newspaper, which obtains a clean tag-free text from a given Web page. Once the Twitter user profiles were complete, we collected data from Goodreads for the book profiles.

User Goodreads IDs were obtained from the tweets of default reviews. Next, a scraper was developed to retrieve all review IDs and dates from users' "read books" lists, which contain only completed books. The Goodreads API was consulted to extract information about all books read by a user, including book metadata, text reviews, ratings, read date and added date. The book metadata, which can be used to build content-based recommender systems, include title, authors, language, the average rating of all reader, the number of pages, publisher, publication date, text review count and book description. Figure 5.1 shows the statistics of the dataset after preprocessing.

[5] To download: https://tinyurl.com/ycc9qfhw

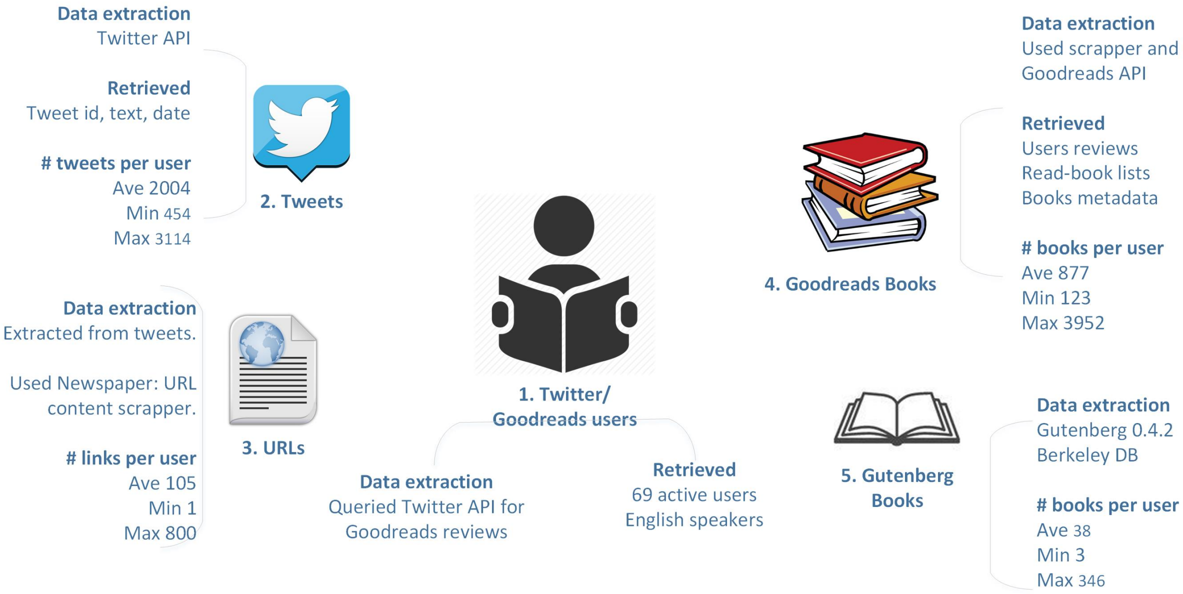


Figure 5.1: Dataset collection and statistics.

When users insert new books into their lists, they may discuss them on their social media. Therefore, in TMB, the recommendation timeframe RT considers added dates instead of read dates. The read date indicates the time of completion of a read book, while the added date is the time when a book was catalogued.

The rating scale, according to Goodreads, treats 1-2 stars as “dislike”, and 3-5 stars as “like”; books rated 3-5 will be called *relevant* in the remainder of the chapter. The number of users shrunk to 69 after the deletion of non-English users, inactive users and those with private Goodreads accounts. Even though many datasets with large number of users exist, some recommendation methodologies such as TMB require personal information about users. This makes it hard to experiment on large datasets. Examples of such work include (Tkalčič et al., 2013) which used a dataset of 52 users to test affective-based RS, and (Odić et al., 2011) which tested a context-aware RS on an 89-user dataset.

5.3.2 Data preprocessing

Before topic extraction, we cleaned text of tweets, links, and book descriptions as follow:

- *Tokenization and POS tagging*:

The tweets were tokenized using Tweet Tokenizer from NLTK (Bird et al., 2009), which is Twitter-conscious, and tagged using the GATE Twitter part of speech tagger (Derczynski et al., 2013). For links and book descriptions, regular NLTK Word tokenizer (Bird et al., 2009) and Stanford part-of-speech tagger (Toutanova et al., 2003) were applied after the deletion of HTML tags using BeautifulSoup, a python library.

- *Noun-based user and book profiles:*

Only nouns (singular or plural) were kept, then lemmatized by NLTK WordNet Lemmatizer (Bird et al., 2009). A noun, according to Merriam-Webster, represents an "entity, quality, state, action, or concept". Nouns, then, can capture the interests of users more than any other part of speech. Other researchers also considered only nouns to model user interests based on their social media accounts (Choi et al., 2014) and to model books (Tsuji et al., 2014). Also, Jockers and Mimno (2013) found that the best way to identify literary themes was extracting noun-based LDA topics from book's full texts.

- *Removal of useless information from user profiles:*

- Repeated content of links.
- Tweets with Goodreads links as they included information about users' reading history. Any Goodreads link was also discarded.
- Web-related terms, *e.g.*, *website* and *Facebook*.
- 100 most common English nouns in Oxford English Corpus because such words mostly do

not reflect interests *e.g.*, *man* and *world* (see Appendix B.1).

- Words of fewer than four letters as to eliminate noise (i.e., unfiltered abbreviations such as RT).
- Misspelled hashtags because the goal is to match them with book descriptions, which are spelling-error-free. They were checked using aspell.[6].
- Words with lowest *idf* weights, which indicates that the topics are not specific to the target user (see Appendix B.2).

User profiles were built automatically in that we did not select the topics for each individual. However, we evaluated the topic models based on our observations. For example, after building topic models from tweets and links, we noticed that unimportant (generic) terms such as *website* are dominant which led to the deletion of web-based and 100 common nouns. Also, when creating NMF-based profiles, we selected the number of topics that resulted in coherent and non-redundant topics.

5.3.3 System implementation

A user Twitter timeline was divided in half, and the date of the middle tweet was considered a time threshold that differentiates learning and recommendations periods. This division keeps enough tweets to build user profiles and enough books to predict. To ensure that tweets do not address the predicted books, a one-month difference was set between the timeframes. The average numbers of tweets and books included in the learning period are 758 and 802, while the minimum numbers are 121 and 13 respectively. Many researchers, including (Wang et al., 2009) adopted 10 to be the lowest number of ratings needed to develop CB.

[6] http://aspell.net/

We developed twelve variations of user profiles. They differ in the topic modelling technique (NMF or *tf-idf*), in the source of data (tweets alone, links alone, or tweets and links) and in the word representations (embeddings [emb] or none). The NMF algorithm was implemented using the scikit-learn Python package (Buitinck et al., 2013). After conducting many trials, the number of NMF topics was set to five, with six words in each, because topics became redundant afterward. The maximum number of *tf-idf* topics was set to 100 to avoid including unimportant words. To calculate cosine similarity between words vectors, we used genism, a Python library. Not all topics have corresponding word vectors, and a reduction in the number of topics is expected.

5.4 Experiments and Results

We measure the predictive power of the system using off-line evaluation, which is appropriate for obtaining the accuracy of an RS. Our approach was compared to a traditional metadata-based CB and to a random system in a top-K recommendations scenario. CB was implemented using the default settings of Graphlab, a well-established framework for RSs. Books in the CB training and test sets were represented with book metadata (see section 5.3.1). Although we considered a comparison with collaborative filtering, the rating matrix is highly sparse, which means that the results would not reflect a typical CF.

Since each user data is divided according to a time threshold, we cannot apply k-fold cross-validation; therefore, we tested TMB in a similar fashion to the leave-one-out evaluation applied in (Koren, 2008; Lu et al., 2012). We created one set of 1000 random books that are unique and not rated by any user. For each user, we randomly selected one relevant book from the recommendation timeframe, added it to the 1000 books and asked our systems to perform ranking. If the rank of the relevant book is f, the RS should have the lowest f value (preferably

1). If $f \leq k$, it is a hit, otherwise it is a miss. Similar to many related projects, we set k to 10. Metrics adopted are hit-rate (Equation 5.4), sometimes called recall@k, and the average reciprocal hit-rank (Equation 5.5) (Deshpande and Karypis, 2004). For each test case, recall@10 is either equal to one (hit) or zero (miss). To avoid bias, five trials were conducted, and the reported results averaged. We measured the statistically significant difference in results using the t-test at a maximum of p-value = 0.05. Some of the randomly chosen 1000 books might have topics similar to a user profile. On the other hand, they could share similar content, *e.g.*, author or description with a user's read books. However, we did not filter out such books because this would introduce a bias and favour one system over the other. In addition, for each of the 69 users, we examined five books, so the overall number of tested books is 345. If there is a bias with a few books, it should not affect the majority of test cases.

$$HR = \frac{\#hits}{\#users} \tag{5.4}$$

$$ARHR = \frac{1}{\#users} \sum_{i=1}^{\#hits} \frac{1}{f_i} \tag{5.5}$$

Results

Figure 5.2 shows the HR and ARHR scores of fourteen recommendation techniques. The best-performing methods are CB and tf-idf-emb built with links; it achieved the highest HR results, while CB reached the best ARHR. Tweet-based tf-idf-emb has similar results to CB. The results of these two methods are not statistically significantly different. In general, *tf-idf* gives better results than NMF. This is expected due to the difference in numbers of topic terms. Comparing the algorithms with and without word embedding vectors, the addition of word embedding enhances the performance. The results are statistically significantly different except for the tf-idf-emb of tweets and the tf-idf-emb of tweets and links. There is no consistency in

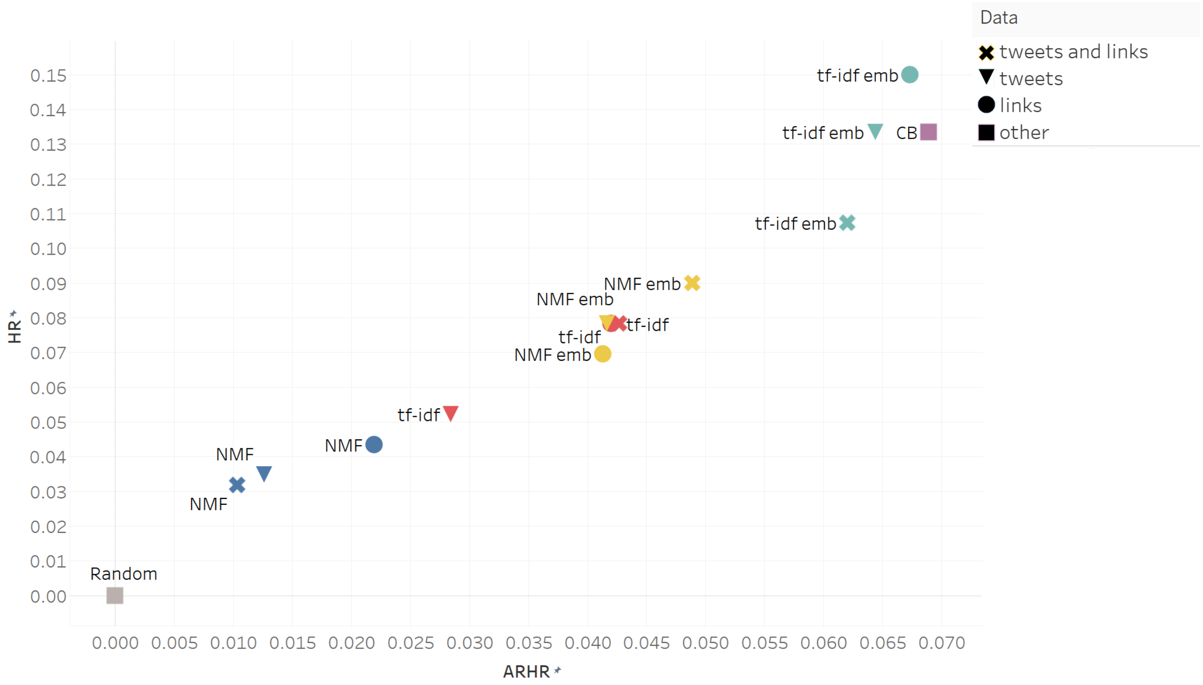


Figure 5.2: The comparison of TMB approaches, CB and the random system.

the effect of using tweets or links. For example, using links with *tf-idf* gives the highest score but with NMF-emb the score is the lowest among all data categories. The random system could not bring any relevant book to the top k.

5.5 Discussion

TMB gives similar performance to a traditional CB system without the need for user rating history. This does not mean that TMB works independently as a standalone system; more comprehensive experiments are required to verify it.

One suggested method, which gives the best results, is to use word embeddings of top *tf-idf* terms. The use of *tf-idf* weighting allows the capturing of distinctive topics frequently discussed by one user in contrast with those discussed by her community. To keep most dis-

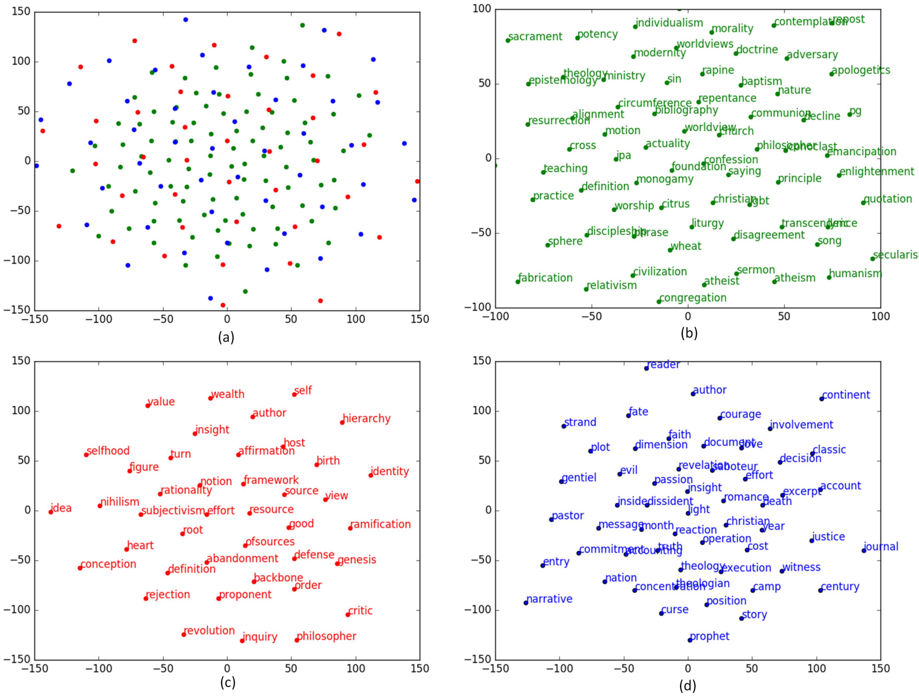


Figure 5.3: Word embeddings of terms in a user profile and two book descriptions.

criminative topics, we only kept the top *tf-idf* words. Otherwise, the average word embedding of all terms in Twitter time-line would be skewed towards less significant terms. We think that this method obtains fine-grained interests not extensively shared among users. For example, a popular interest such as *fiction* would have a low *idf* value and would not be not expected to appear in the top *tf-idf* list. On the other hand, a term that is not as popular, like *mythology*, may have a high *idf* value and be in the top *tf-idf* list. At the same time, many users may share common interests/topics which leads to a low *idf* weight; however, if a topic is predominantly discussed by one user, it will have a high *tf* and consequently a high *tf-idf* value.

All variations of TMB could identify books that interest the user out of a thousand other books, with the link-based tf-idf-emb retrieving the highest number of books. To illustrate how word embeddings contribute to the recommendations made by link-based tf-idf-emb, we plotted (Figure 5.3) the word embeddings of one user profile (b) and his two book profiles (c, d). Section (a) of Figure 5.3 shows the closeness of word vectors found in the UP and BPs. One can notice the variety of topics in the user profile. Also, some topics in the user profile might be semantically similar which may indicate high interest. User interests might be broad and not only related to the books they already preferred.

The textual content of the links can be longer than that of the tweets, and so possibly capture a wider range of interests. Thanks to word embeddings, however, the performance of models that adopt tweets increased dramatically. Word vectors could enrich the topics by including the context of terms. Their improvement of tweet-based algorithms could be due to the presence of hashtags, which summarize a whole subject or event.

5.5.1 Error analysis

We conducted error analysis to investigate the differences in performance between CB and TMB (*tf-idf* based on links and word embeddings). In a leave-one-out evaluation, we tested five books for each user. The two systems retrieved the same number of relevant books when giving recommendations to 31 users. CB could retrieve more relevant books than TMB for 17 users, whereas TMB surpassed CB when dealing with 21 users. For better understanding, we analyzed each system's best recommendations.

CB retrieved three out of five books relevant for users A and B, while TMB suggested only one book to user A and none to B. User A had 512 books in the CB training set, while user B had 542. From the three books recommended to A, only one had the same author as a book in the training set; that is to say, CB relied on book descriptions to make the recommendation. The one book which TMB recommended to user A had cosine similarity of 0.69 and shared words that were semantically close to the user topics (*e.g.*, *drawing* vs. *illustrator*). Like for user A, only one book recommended by CB to user B shared the same author with a book in the CB training set. The possible reason why TMB could not suggest any book to user B is that the user's topics were related to political issues (*e.g.*, *abortion*, *immigration*), while the user's readings were diverse. For example, user B's five relevant books addressed history, romance, philosophy and education. The user's interests were broad, while his discussed topics on Twitter were narrow and related to current issues.

Users C, D and E received five, three and two recommended books by TMB, respectively, while CB could recommend two books for user C and none for user D and E. User C had 140 books in the CB training set. Most of his readings were related to religious matters. The user topic profile reflected these interests: the top *tf-idf* words were *glory*, *theology* and *gospel*. The lowest cosine similarity between the user topic profile and the five retrieved books was 0.72. User D had 13 books in the CB training set. All books in the training and test set were written by distinct authors. The user topics were also related to philosophy and religion, as well as the user readings (see Figure 5.3). User E had 528 books in the CB training set. Her topic profile covered wide interests (*e.g.*, *courtroom*, *femininity*, *mutiny*, and *heroin*) and the two recommended books were slightly similar. One of them, titled “Against the Country”, was described with words such as *offender*, *antihero* and *blast*. The other book was described with such words as *assassination* and *murder*.

5.5.2 Limitations

TMB is a useful solution for cold start issue when users are active on social media and willing to share their data; however, this is not the case for all users. Therefore, a new user can have the option to fill a questionnaire about their reading preferences or share their social media account. Also, a deployed system should allow users to modify their *tf-idf* topics as users may share unpleasant experiences on Twitter and would not like to receive book recommendations on the subject. We considered relying solely on tweets with positive sentiment; however, users may write about their interests in a negative context (*e.g.*, an environment-conscious user discussing climate change).

Another limitation is that we experimented with readers who shared a link to Goodreads, which indicates they may discuss their readings on Twitter. To evaluate TMB for other users who do not address their readings on social media, a user study is required because their reading history is not available.

Also, social media accounts might be utilized for other purposes (*e.g.*, book promotions) other than personal use. In this research, we assume that all users’ data is not fake. However, in a deployed system such issue has to be addressed.

5.6 Conclusion and Future Work

In this chapter, we:

- Proposed TMB, which is a recommendation component that helps existing RSs to overcome the user cold start issue. To our knowledge, this is the first book RS that exploits social media other than book-cataloguing Web sites.

- Automatically represented the dominant topics discussed by a user without searching for predefined concepts, recognizing named entities or developing ontologies.
- Modeled a user's interests is by acquiring her distinctive topics using *tf-idf*, which yields the topics frequently discussed by the user in contrast with those discussed by her community, and then representing them as word embeddings.
- Collected a dataset that contains user profiles in two Web sites namely Twitter and Goodreads.
- Evaluated TMB in top-k recommendation scenario where it retrieved a number of books comparable to CB's, even though CB relied on users' rating history while TMB only needed their social profiles.
- Tested many variations of TMB which could identify books that interest users out of a thousand other books, with the link-based tf-idf-emb retrieving the highest number of books.

This work can be extended in the following ways:

- Since hashtags carry more meaning than other terms on Twitter, an interesting approach would be to create hashtag-based profiles that are enriched with word vectors. Also, user profiles could include other parts of speech, particularly verbs and adjectives.
- Investigate how the use of recent tweets/links vs. old ones affect the recommendation accuracy.

- To enhance the performance of the system, embeddings models can be trained over tweets and books.

- Since the calculation of one user's *tf-idf* topics relies on the number and interests of the other users in the system. It is interesting to investigate the effect of the number of users on the quality of *tf-idf* topics.

- The same approach can be tested for other social media platforms and for other user generated texts.

- It is interesting to test TMB's ability to generate users' interests in real time.

Chapter 6

Conclusion and Future Work

6.1 Summary

Literary book recommender systems (RSs) can promote the practice of reading for pleasure, which has been declining in recent years. Only a few previously proposed book RSs actually take the book texts into consideration. In this thesis, we developed book representations learned from their texts and showed that they lead to improved recommendation performance compared to many state-of-the-art content-based and collaborative filtering systems. The systems we designed are complemented with TMB (Topic Model-Based Book Recommendation Component), which recommends books for new users.

The first system, discussed in chapter 3, represents books as the penultimate layer in a convolutional neural network that learns from book sequences of words to predict their authors. To our knowledge, this is the first work that applies the information learned by an author-identification model to book recommendations. The AuthId book representations are input to a regressor (support vector regression) which finds patterns in books read by the user. We showed that this method outperforms many state-of-the-art CB and CF systems. We also evaluated the recommendation accuracy given AuthId book representations which were built using shorter texts, and concluded that the longer the included texts the more accurate the recommendations. Our qualitative analysis showed that similar book representations are annotated similarly by experts. We also performed error analysis and found that many relevant books ranked in the top ten list are retrieved from a large collection of books written by the same authors.

In chapter 4, we investigated the textual elements that could play a role in generating high-quality book recommendations. Two recommendation algorithms were trained on more than a hundred features learned from the full text of the books, and both algorithms generated more accurate lists of suggested books than many competitive baselines. We highlighted the variables that are considered important using an ensemble of randomized trees and produced book suggestions based on them. Further qualitative analysis showed that similar books generated by our system share the same author and similar descriptions on NoveList. We also compared an approach that uses linguistic features against the AuthId approach and tested their concatenated book representations.

Since both CBs above are prone to the *new user issue*, we proposed TMB (in chapter 5), a module that builds a topic model for users from textual content shared voluntarily on their social media, and recommends books most related to these topics. For each user, we acquired distinctive topics by *tf-idf* weighting, which yields the topics frequently discussed by the user in contrast with those discussed by the community, and represented them as word embeddings to capture their context. For evaluation purposes, we collected a dataset that contains user profiles from two Web sites, namely Twitter and Goodreads. TMB achieved a recommendation accuracy similar to a traditional CB (a commonly used book RS), particularly when word embeddings were deployed.

6.2 Highlights and Challenges

6.2.1 Explainability

As CB systems rely solely on the target user ratings, they overcome many CF associated issues such as rating matrix sparsity, new item issue and shilling attacks. Moreover, CB is a transparent system that provides explainable suggestions. For example, our proposed system, feature-based RS, can provide users with accurate recommendations and a breakdown of the most and least important features that contributed to the accuracy. The system also allows publishers and authors to identify the aspects that matter most for individuals and communities. The same applies to AuthId and TMB, which recommend books because they have similar writing styles or topics, respectively.

6.2.2 Overspecialization

CB recommender systems' dependence on target user ratings results in an issue called overspecialization, which means it generates non-diverse recommendations. In the literature, *overspecialization* is addressed by recommending some random items, eliminating the most similar (*e.g.*, fresh news stories on previously read topics) (Billsus and Pazzani, 2000) and increasing diversity (see (Kaminskas and Bridge, 2016)) (Adomavicius and Tuzhilin, 2005).

One way to help our feature-based RS overcome *overspecialization* is to allow users to retrieve books based on the features that matter most to them. For example, when conducting the qualitative analysis in section 4.4 we eliminated fiction-based features from book representations, and a book by Henry James that was described as similar in terms of character and storyline was not retrieved. Another technique to manage overspecialization is to develop a hybrid system that uses other recommendation methods (*e.g.*, CF) in addition to CB.

6.2.3 Availability of adequate content

Another drawback of CB is the difficulty of accurately suggesting items without adequate content. While the full texts of books provide abundant content for analysis, the trade-off is the cost of copyright limitations. And, as stated in section 1.2.2, though there are many recommendation and retrieval systems that process the full texts of books this is not widely practiced, which could be due to the difficulty of obtaining publisher agreements. The analysis of texts also requires applying NLP techniques that are specific to each natural language.

6.2.4 Books in multiple languages

To generate recommendations to users who read books in multiple languages, the feature-based CB would require tools to analyze texts in each language and create book representations that eventually can be fed to one recommendation algorithm. This is not the case for the authorship-based RS which needs a language-specific AuthId classifier whose generated book representations are input to different recommendation components to generate separate recommendation lists. In term of TMB, if user profiles and/or book profiles contain terms from several languages, all the terms can be translated into one language in order to contribute to the word embedding representations. Alternatively, one can map terms from different languages to multilingual word embeddings, as in (Conneau et al., 2018).

Furthermore, learning about translated book preferences raises questions, such as the book's likes or dislikes being attributed to translation quality rather than content. Translated books are expected to share the same topics and themes with their original versions. Our systems should discover similar books whether they are translated or not. For example, in the feature-based approach, if a book is characterized to have subjective language, rich lexical density and few fictional characters, the RS should find books with similar representations regardless of being translated or not. TMB, on the other hand, considers topics and interests which should not differ from the original to translated books. AuthId representations are also expected to capture style features that reflect an author's topics and word choices but not sentence structure and more advance stylistic aspects.

6.2.5 Limitations in the evaluation

As with most work in RS literature, we conducted an offline assessment using a formerly collected dataset, and reported the RS performance in term of accuracy. However, online evaluations using a deployed RS and user case studies could provide in-depth analysis of the system, and highlight additional benefits and limitations. It would also be useful to have experts perform qualitative analysis of books considered similar. Also, though accuracy profoundly influences user satisfaction (Hijikata et al., 2012), other beyond-accuracy goals can be achieved by RSs to ensure quality recommendations, including diversity, serendipity, and novelty. Diversity in RSs involves recommending a list of various item types, serendipity is recommending items that are surprising to the user and novelty is recommending unknown items. Kaminskas and Bridge (2016) investigated these objectives and surveyed metrics and methods to achieve them. With respect to accuracy, we measured the performance of the system in the top-k scenario, rather than predicting the user level of interest in the items (*e.g.*, five stars). While ranking is useful on e-commerce platforms, multimedia and content recommendations (*e.g.*, news), ratings prediction helps users decide about a product.

Due to copyright issues, texts of books other than Gutenberg texts were not accessible for this study as they were authored in the past 50 years. Gutenberg books are typically categorized as classics; plus, such books might not appeal to a wide range of ages. Furthermore, genres such as graphic fiction and manga, which are included in the reader advisory recommendations, are not considered in our study. It would be interesting to conduct similar research on books from various times and different genres.

6.3 Deployment of Our Systems

Our CB systems can work independently or jointly, by recommending items that were highly rated by multiple systems or generating a combined recommendation list. We believe that users should have the choice of what recommendation methodology they are interested in. According to (Smith et al., 2016), readers who tend to appreciate writing style consider it the most appealing factor. The AuthId approach gives users who are interested in writing style a chance to find books/authors similar to the style they prefer. In addition, users can choose feature-based CB to retrieve books with similar content and/or style. A deployed system can also provide a service to discover the most similar book representations to a queried book/author.

RSs can always complement each other, and it is better to use a hybrid RS. With Netflix, for example, users are presented with lists of movies/shows that include recommendations based on content, popularity, other users' ratings (CF), recency and other aspects. In our system, users could receive lists generated by multiple methods. If permission to use a book's text is not granted, a different RS can include the book in its recommended list. This also applies to TMB, which can be applied if new users have social media accounts and are willing to share their information. The new users might prefer to share their social media information instead of completing a questionnaire on their reading preferences, personality traits, and demographics.

6.4 Future Work

Further investigation of our methods could include applying word sense disambiguation, incorporating more factors in the feature-based approach, trying other author identification methods (which could also consider differences in genre and time) and examining the use of hashtags, verbs, and adjectives when building TMB user profiles. Furthermore, recent more advanced forms of word embedding such as FastText and ELMo (Embeddings from Language Model) showed enhanced performance in various NLP applications, and it could be interesting to use them in TMB or as pre-trained embeddings to create AuthId representations.

6.4.1 Deep-learning-based RSs

Neural networks offer many ways to improve RSs. A basic architecture could apply a feedforward NN in RSs (proposed by (He et al., 2017)) by representing users and items as embedding, merging them, and passing them to hidden layers to learn interactions between users and items. In a book RS, the system would incorporate other lookup tables for authors, fictional characters and user demographic information. Such embeddings can be initialized arbitrarily or pre-built, and those generated by the trained model can contribute to understanding user preferences. For example, a model that captures the interactions between embeddings representing users' gender and fictional characters could provide insights into the fictional character preferences of females or males. We implemented such a system but were unable to achieve high results, which could be due to the limited size of our dataset. However, this is still a promising approach, and further investigation is strongly recommended. It is possible to train the network over multiple large datasets, such as Amazon reviews and BookCrossing. Then, the learned item and author embeddings can be transferred to a model working on smaller datasets with available texts, such as LitRec. Multitask learning is another promising approach: to jointly learn to predict authors (author identification) and user ratings (book recommendations) given book texts.

6.4.2 Graph-based RSs

Graph theory has many benefits for book RS researchers. For digital libraries, Huang et al. (2002) recommend books after searching a two-layer graph and traversing three types of links; namely, user-user (demographic similarity), item-item (similarity in content) and item-user (purchase transaction). Resemblance among books could be extended to include many elements specific to literature. In the case of poetry, the Graph Poem (Tanasescu et al., 2018) represents poems as nodes in a graph and connects them by edge, characterizing topics, themes, metaphors and other shared aspects of poems; the aspects are calculated using tools dedicated to poetic features. It would be exciting to extend graph-based book RSs to include such artistic features. Furthermore, we believe it is worthwhile to investigate how to find similar books using social networks of fictional characters that are linked based on dialogue interactions (as proposed in (Elson et al., 2010)).

6.4.3 The effect of mood on book selection

In one study, 194 participants shared several elements that influence why they choose or reject a particular book. Reader mood was found to be the core element when choosing a book, as it influences the desired reading experience. If readers are stressed and tired they typically read familiar, safe, easy and positive books, while if they are relaxed they usually go for unusual, risky, challenging or dark books (Ross, 2000). The field of emotions and mood-aware RSs is already active, though to our knowledge it is not yet applied to book recommendations.

6.4.4 Books for non-readers

Book RSs can also appeal to non-readers and encourage them to take up reading. An automatic personalized system can understand individual interests and reading-related issues (*e.g.*, readability) using social media data, then make recommendations accordingly. The non-reader issue resembles user cold start in RSs, except that non-readers are not interested in receiving recommendations in the first place. Recognizing what kindles the spark of reading can help attract new users.

Appendix A

Details about the Linguistic Features

A.1 The full list of LIWC features with examples

Table A.1: LIWC Dimensions, labels and examples

LIWC Dimension	**Output Label**	**Example**
Word Count	WC	
Summary Variable		
Analytical Thinking	Analytic	
Clout	Clout	
Authentic	Authentic	
Emotional Tone	Tone	
Language Metrics		
Words per sentence	WPS	
Words>6 letters	Sixltr	
Dictionary words	Dic	

Function Words	function	it, to, no, very
Total pronouns	pronoun	I, them, itself
Personal pronouns	ppron	I, them, her
1st pers singular	i	I, me, mine
1st pers plural	we	we, us, our
2nd person	you	you, your, thou
3rd pers singular	shehe	she, her, him
3rd pers plural	they	they, their, they'd
Impersonal pronouns	ipron	it, it's, those
Articles	article	a, an, the
Prepositions	prep	to, with, above
Auxiliary verbs	auxverb	am, will, have
Common adverbs	adverb	very, really
Conjunctions	conj	and, but, whereas
Negations	negate	no, not, never
Grammar Other		
Regular verbs	verb	eat, come, carry
Adjectives	adj	free, happy, long
Comparatives	compare	greater, best, after
Interrogatives	interrog	how, when, what
Numbers	number	second, thousand
Quantifiers	quant	few, many, much
Affect Words	affect	happy, cried

Positive emotion	posemo	love, nice, sweet
Negative emotion	negemo	hurt, ugly, nasty
Anxiety	anx	worried, fearful
Anger	anger	hate, kill, annoyed
Sadness	sad	crying, grief, sad
Social Words	social	mate, talk, they
Family	family	daughter, dad, aunt
Friends	friend	buddy, neighbor
Female referents	female	girl, her, mom
Male referents	male	boy, his, dad
Cognitive Processes	cogproc	cause, know, ought
Insight	insight	think, know
Cause	cause	because, effect
Discrepancies	discrep	should, would
Tentativeness	tentat	maybe, perhaps
Certainty	certain	always, never
Differentiation	differ	hasn’t, but, else
Perpetual Processes	percept	look, heard, feeling
Seeing	see	view, saw, seen
Hearing	hear	listen, hearing
Feeling	feel	feels, touch
Biological Processes	bio	eat, blood, pain
Body	body	cheek, hands, spit

Health/illness	health	clinic, flu, pill
Sexuality	sexual	horny, love, incest
Ingesting	ingest	dish, eat, pizza
Core Drives and Needs	drives	
Affiliation	affiliation	ally, friend, social
Achievement	achieve	win, success, better
Power	power	superior, bully
Reward focus	reward	take, prize, benefit
Risk/prevention focus	risk	danger, doubt
Time Orientation		
Past focus	focuspast	ago, did, talked
Present focus	focuspresent	today, is, now
Future focus	focusfuture	may, will, soon
Relativity	relativ	area, bend, exit
Motion	motion	arrive, car, go
Space	space	down, in, thin
Time	time	end, until, season
Personal Concerns		
Work	work	job, majors, xerox
Leisure	leisure	cook, chat, movie
Home	home	kitchen, landlord
Money	money	audit, cash, owe
Religion	relig	altar, church

Death	death	bury, coffin, kill
Informal Speech	informal	
Swear words	swear	fuck, damn, shit
Netspeak	netspeak	btw, lol, thx
Assent	assent	agree, OK, yes
Nonfluencies	nonfl	er, hm, umm
Fillers	filler	Imean, youknow
All Punctuation	Allpunc	
Periods	Period	
Commas	Comma	
Colons	Colon	
Semicolons	SemiC	
Question marks	QMark	
Exclamation marks	Exclam	
Dashes	Dash	
Quotation marks	Quote	
Apostrophes	Apostro	
Parentheses (pairs)	Parenth	
Other punctuation	OtherP	

A.2 The full list of importance values generated by ET

Table A.2: All Features ordered according to ET importance values.

Feature	importance value
see	0.013115
author	0.011821
percept	0.011377
auxverb	0.010999
num_char_mention	0.010717
shehe	0.010623
Length of text	0.010545
num_said	0.010375
concrete	0.01029
Percent of text adverbs	0.009883
num_places	0.009876
prep	0.00986
article	0.009777
objective	0.009756
Average variance in sentence length	0.009679
Average length of sentence	0.009588
abstract	0.009544
drives	0.009516
Period	0.009499

bio	0.009443
negate	0.009435
certain	0.009432
Analytic	0.009335
Dic	0.009322
function	0.009265
focusfuture	0.009192
average_syllable_per_word	0.009091
polarity	0.009015
body	0.008949
num_uniuqe_char	0.008926
Average number of commas per sentence	0.008911
sad	0.00891
subjective	0.008905
num_unique_places	0.008901
literary	0.008883
QMark	0.008805
discrep	0.008753
power	0.008701
female	0.0087
Percent dialogue by female characters	0.008679
Parenth	0.008655
average_dialog_length	0.008624

differ	0.008618
ipron	0.008606
Lexical density	0.008599
colloquial	0.008582
achieve	0.008571
reward	0.008555
SemiC	0.008518
feel	0.008514
they	0.008489
conj	0.008487
Authentic	0.008467
relativ	0.008449
Apostro	0.00842
Tone	0.008333
Sixltr	0.008323
focuspresent	0.008321
AllPunc	0.008259
anger	0.008228
Percent of female characters	0.008225
social	0.008191
motion	0.00819
family	0.008173
compare	0.008145

interrog	0.008145
male	0.008143
cogproc	0.008138
Percent of text verbs	0.008034
death	0.008017
space	0.008012
Quote	0.007942
precentage_dialog_text	0.007935
posemo	0.007929
Dash	0.007917
Average variance in paragraph length	0.007913
number	0.007887
i	0.007878
work	0.007852
health	0.00784
Average length of paragraph	0.00782
Exclam	0.007805
insight	0.007796
home	0.007758
Comma	0.007743
quant	0.007738
focuspast	0.007691
hear	0.007641

pronoun	0.007588
flesch_reading_ease_score	0.007559
ppron	0.007513
affect	0.007407
Percent of text adjectives	0.007347
we	0.007288
risk	0.007285
relig	0.007254
nonflu	0.007241
Percent of text nouns	0.007213
negemo	0.007209
you	0.007191
friend	0.007177
affiliation	0.007156
tentat	0.00713
money	0.007112
Colon	0.007068
Percent of text Latinate words	0.007011
time	0.006879
assent	0.006863
Average length of words	0.006849
cause	0.006838
Clout	0.006629

ingest	0.006626
sexual	0.006617
informal	0.006605
leisure	0.006556
netspeak	0.006444
anx	0.00636
swear	0.006266
filler	0.005838

A.3 The correlation values between highly correlated features

Table A.3: Highly correlated features in descending order

feature1	**feature2**	**spearmanr**
number of times places are mentioned	number of unique places	0.945322
relativ	space	0.914507
average_syllable_per_word	Sixltr	0.914247
Average length of sentence	Average number of commas per sentence	0.896296
Average length of sentence	Average variance in sentence length	0.891417
Analytic	article	0.882791
number of fictional characters mentions	number of dialogs	0.873118
objective	Sixltr	0.873
affect	posemo	0.866654
Average number of commas per sentence	Average variance in sentence length	0.85241
number of fictional characters mentions	Length of text	0.851341
bio	body	0.843713

Dic	function	0.837702
AllPunc	Quote	0.829261
percept	see	0.824278
number of fictional characters mentions	number of unique fictional characters	0.823692
number of dialogs	Length of text	0.821914
number of unique fictional characters	Length of text	0.820087
colloquial	AllPunc	0.818877
informal	nonflu	0.818846
Tone	posemo	0.818409
Average length of words	Sixltr	0.816651
Percent of text Latinate words	Sixltr	0.814427
Percent of text Latinate words	average_syllable_per_word	0.811298
Average length of paragraph	Average variance in paragraph length	0.809659
concrete	relativ	0.80277
negate	differ	0.800693
number of unique fictional characters	number of dialogs	0.790596
objective	average_syllable_per_word	0.790023

Clout	social	0.788127
concrete	space	0.787548
Average length of words	Percent of text Latinate words	0.781971
cogproc	insight	0.777673
number of unique places	Length of text	0.772651
abstract	objective	0.772035
number of times places are mentioned	Length of text	0.771608
auxverb	discrep	0.770456
Average number of commas per sentence	Comma	0.765014
relativ	motion	0.762224
cogproc	tentat	0.759638
cogproc	discrep	0.753661
concrete	motion	0.751549
number of unique fictional characters	number of unique places	0.751487
colloquial	Apostro	0.740577
Percent of text Latinate words	objective	0.738185
number of unique fictional characters	number of times places are mentioned	0.73571

Average length of words	objective	0.731746
Average length of words	average_syllable_per_word	0.73095
Percent dialogue by female characters	Percent of female characters	0.727571
cogproc	differ	0.727364
we	affiliation	0.720784
QMark	Quote	0.71701
subjective	affect	0.71085
number of fictional characters mentions	number of times places are mentioned	0.709968
Average length of paragraph	Average length of sentence	0.706295
shehe	social	0.704421
abstract	Sixltr	0.703331
AllPunc	QMark	0.702464
feel	body	0.701584
negemo	anx	0.701336
cogproc	certain	0.699618
Lexical density	Percent of text nouns	0.696846
auxverb	cogproc	0.695565
abstract	average_syllable_per_word	0.688145
drives	affiliation	0.686134
Percentage of dialog text	you	0.685684

social	female	0.678266
function	auxverb	0.67692
literary	relig	0.676712
Dic	pronoun	0.676322
colloquial	Quote	0.674849
percept	feel	0.673496
number of fictional characters mentions	number of unique places	0.672415
drives	power	0.670726
Percent of text Latinate words	abstract	0.669729
subjective	posemo	0.662715
Percent of text verbs	focuspast	0.662678
you	QMark	0.660814
function	pronoun	0.660038
Percent of female characters	female	0.658421
Percentage of dialog text	Quote	0.648597
motion	space	0.64802
auxverb	focuspresent	0.643581
pronoun	i	0.642205
informal	assent	0.640961
feel	bio	0.639746

ipron	cogproc	0.635165
you	focuspresent	0.633838
percept	body	0.631744
family	female	0.627357
function	differ	0.626209
subjective	female	0.62268
informal	AllPunc	0.621785
Dic	auxverb	0.62141
AllPunc	Apostro	0.621276
percept	hear	0.618708
number of dialogs	number of times places are mentioned	0.616882
literary	abstract	0.61599
affect	sad	0.613916
negemo	sad	0.612396
colloquial	informal	0.61116
assent	nonflu	0.608588
Average length of words	Analytic	0.608014
social	family	0.606708
number of dialogs	number of unique places	0.605526
concrete	feel	0.604919
shehe	female	0.60271
negate	cogproc	0.597715

concrete	see	0.59678
cogproc	cause	0.596508
Authentic	we	0.596437
function	negate	0.596292
Average length of sentence	prep	0.595493
affect	negemo	0.594628
Average length of paragraph	Average variance in sentence length	0.594148
pronoun	social	0.591462
Percent dialogue by female characters	female	0.590371
you	Quote	0.590343
see	body	0.589186
concrete	percept	0.588753
shehe	male	0.587868
Average variance in paragraph length	Average variance in sentence length	0.587081
insight	tentat	0.586467
Average length of words	prep	0.586078
we	drives	0.584292
average_syllable_per_word	work	0.581434
auxverb	differ	0.580869
focuspresent	focusfuture	0.579804

Percent of text verbs	auxverb	0.57978
informal	netspeak	0.577181
Dic	cogproc	0.577118
discrep	tentat	0.576683
relativ	time	0.576284
objective	Analytic	0.575388
Percentage of dialog text	focuspresent	0.57354
Average length of words	Percent of text adjectives	0.571221
negemo	anger	0.570031
Average length of sentence	Average variance in paragraph length	0.569986
function	ipron	0.568424
average_dialog_length	Average length of paragraph	0.567453
Average length of paragraph	prep	0.567062
subjective	social	0.565026
auxverb	focusfuture	0.564562
Clout	shehe	0.564429
discrep	differ	0.563602
see	feel	0.563587
discrep	focuspresent	0.56011
percept	bio	0.556452

Average length of paragraph	Average number of commas per sentence	0.555968
Average number of commas per sentence	SemiC	0.554327
compare	quant	0.551965
posemo	friend	0.551721
auxverb	negate	0.551566
Average variance in sentence length	SemiC	0.551301
function	cogproc	0.550061
pronoun	insight	0.549095
bio	health	0.547782
Dic	negate	0.544489
colloquial	QMark	0.54304
Lexical density	colloquial	0.542985
nonflu	AllPunc	0.540903
Percentage of dialog text	AllPunc	0.534266
AllPunc	Period	0.534225
informal	Quote	0.533969
informal	Apostro	0.533773
objective	article	0.531588
Percent of text verbs	discrep	0.531541
Average length of sentence	compare	0.531071

Percentage of dialog text	QMark	0.530546
Percent of text nouns	Analytic	0.528371
quant	certain	0.528215
pronoun	you	0.525025
anger	death	0.523704
pronoun	cogproc	0.523235
Analytic	Sixltr	0.51951
you	AllPunc	0.518611
Average variance in sentence length	literary	0.518331
Percent of text nouns	colloquial	0.51783
ipron	auxverb	0.517657
Average length of sentence	SemiC	0.516775
ipron	insight	0.516529
cogproc	focuspresent	0.514917
Dic	discrep	0.512462
Percent of text adverbs	cogproc	0.511709
pronoun	auxverb	0.511313
flesch_reading_ease_score	Period	0.510352
Average length of words	article	0.509311
Average length of paragraph	compare	0.508174
posemo	female	0.508158

pronoun	discrep	0.508118
concrete	body	0.507745
Average number of commas per sentence	literary	0.506625
Average length of words	abstract	0.506438
posemo	family	0.506266
polarity	Tone	0.505959
Average length of sentence	conj	0.50536
Average variance in sentence length	prep	0.505066
function	discrep	0.50334
Dic	differ	0.502888
ipron	tentat	0.50195
Percent of text nouns	AllPunc	0.50145
Average variance in sentence length	Comma	0.501399
concrete	negate	-0.50199
subjective	motion	-0.5027
Percent of text Latinate words	motion	-0.50286
auxverb	body	-0.50416
Percent of text nouns	differ	-0.50608
cogproc	space	-0.50751

average_syllable_per_word	focuspast	-0.50947
Average variance in paragraph length	Quote	-0.51048
Average variance in paragraph length	Period	-0.51068
Period	SemiC	-0.51092
Sixltr	conj	-0.51179
colloquial	function	-0.51356
concrete	posemo	-0.51413
concrete	cogproc	-0.51553
abstract	feel	-0.51608
social	space	-0.51611
Average length of words	AllPunc	-0.51622
percept	achieve	-0.51675
abstract	see	-0.51699
Average length of words	informal	-0.5209
Average length of words	focuspresent	-0.52257
Tone	anger	-0.52406
prep	Period	-0.52435
Analytic	differ	-0.52578
Percent of text Latinate words	concrete	-0.52953
abstract	space	-0.53026

Percent of text nouns	cogproc	-0.53136
prep	Apostro	-0.53159
Percent of text Latinate words	percept	-0.53195
Average length of sentence	flesch_reading_ease_score	-0.53386
Percent of text nouns	pronoun	-0.53396
Average length of words	colloquial	-0.53673
Percent of text verbs	literary	-0.5371
Average length of words	hear	-0.53733
average_syllable_per_word	conj	-0.53807
Percent of text verbs	Analytic	-0.53855
negate	space	-0.5415
abstract	focuspast	-0.54166
Percent of text adverbs	Analytic	-0.54855
article	social	-0.54883
article	cogproc	-0.55084
posemo	relativ	-0.55387
prep	QMark	-0.55665
affect	relativ	-0.55839
Average number of commas per sentence	flesch_reading_ease_score	-0.562
article	focuspresent	-0.56286

Average length of paragraph	colloquial	-0.56612
Percent of text adverbs	article	-0.56704
Average variance in sentence length	flesch_reading_ease_score	-0.56901
Average length of sentence	colloquial	-0.56922
Analytic	social	-0.57215
abstract	percept	-0.57243
Average length of sentence	Quote	-0.57512
Lexical density	Dic	-0.57882
Average length of sentence	AllPunc	-0.59006
concrete	subjective	-0.59208
Authentic	social	-0.59215
subjective	relativ	-0.59584
Analytic	function	-0.59648
affect	space	-0.59982
Average length of paragraph	Period	-0.60639
article	discrep	-0.60818
prep	Quote	-0.61081
Average length of sentence	QMark	-0.61478
abstract	flesch_reading_ease_score	-0.61522
posemo	space	-0.61974

Percent of text Latinate words	flesch_reading_ease_score	-0.62445
Average length of words	flesch_reading_ease_score	-0.62627
subjective	space	-0.62741
Analytic	cogproc	-0.63047
Analytic	focuspresent	-0.63346
abstract	relativ	-0.63359
flesch_reading_ease_score	average_syllable_per_word	-0.63398
Clout	Authentic	-0.63851
Average length of paragraph	QMark	-0.63989
Analytic	Dic	-0.63992
prep	AllPunc	-0.64475
objective	flesch_reading_ease_score	-0.64976
colloquial	prep	-0.67435
abstract	motion	-0.67524
flesch_reading_ease_score	Sixltr	-0.67541
Average length of paragraph	AllPunc	-0.68092
Analytic	auxverb	-0.68319
Analytic	discrep	-0.68608
we	shehe	-0.6913
pronoun	article	-0.71055

Average length of paragraph	Quote	-0.7361
Lexical density	function	-0.74088
Authentic	shehe	-0.74155
Percent of text nouns	Dic	-0.74807
abstract	concrete	-0.79531
Percent of text nouns	function	-0.79635
Analytic	pronoun	-0.83538
Average number of commas per sentence	Period	-0.84055
Average variance in sentence length	Period	-0.86743
Average length of sentence	Period	-0.91302

Appendix B

Additional TMB Preprocessing Information

B.1 100 most frequent nouns in Oxford English Corpus

time, year, people, way, day, man, thing, woman, life, child, world, school, state, family, student, group, country, problem, hand, part, place, case, week, company, system, program, question, work, government, number, night, Mr, point, home, water, room, mother, area, money, storey, fact, month, lot, right, study, book, eye, job, word, business, issue, side, kind, head, house, service, friend, father, power, hour, game, line, end, member, law, car, city, community, name, president, team, minute, idea, kid, body, information, back, parent, face, others, level, office, door, health, person, art, war, history, party, result, change, morning, reason, research, girl, guy, food, moment, air, teacher.

B.2 Words with low IDF weights

The nouns with the least IDF weighted which means discussed by most users.

book, love, day, review, read, win, time, today, giveaway, page, story, author, blog, romance, life, video, way, work, series, release, kindle, post, feel, writing, hope, star, photo, novel, end, tour, show, copy, amazon, tweet, news, christmas, game, movie, fun, twitter, word, something, fantasy, help.

Appendix C

Additional Experiments

This appendix shows complementary experiments as follows:

C.1 The difference in performance when predicting binary vs. 1-5 ratings

The tables in this section extend tables 3.2 and 4.1, respectively, by showing the performance of predicting different forms of user feedback. In general, a regressor trained on binary output is expected to learn to rank relevant books in top ranks. This is the case of feature-based RSs as shown below. However, SVR with AuthId book representations performs better when trained over 1-5 rated books as shown in the table below.

Table C.1: Extension of the results presented in table 3.2

	Precision@k	Recall@k
SVR_AuthId (1-5)	0.162	0.325
SVR_AuthId (binary)	0.15038	0.3053

Table C.2: Extension of the results presented in table 4.1

	Precision@k	Recall@k
ER_1000_3features (Binary)	0.168756	0.363426
ER_1000_3features (1-5 ratings)	0.168483	0.361962
KNN (Binary)	0.169846	0.372023
KNN (1-5 ratings)	0.167847	0.367564

C.2 Embeddings of fictional characters and places

We have developed embeddings of fictional characters as we intended to use them in deep-learning based RS, which was supposed to learn from interactions between a user demographic information and her favorite characters. However, due to the size of the dataset, the RS could not perform well. Therefore, we tried to retrieve books with similar fictional character and place embeddings as in the table below. Similar to novel2vec (Grayson et al., 2016), which was not used in RSs, we extract the occurrences of characters and create word embeddings for them. In novel2vec, the annotation of characters was done manually. However, to automate the process, we used fiction-aware entity recognition tool (LitNER). The goal is to characterize a book by two vectors representing the average of the embeddings of persons and places, respectively. One could use only the main character to represent a book. However, it is not easy to automatically decide which character is the main one. Is it the most frequently mentioned name? The average of all person/place embeddings allows us to capture the variability in characters with an emphasis on the frequent ones.

First, names of people and places mentioned in a book's text are extracted. To ensure that named entities are book-specific, all occurrences of an identified named-entity in the text of a book are labeled with the book id (*e.g.*, Raj_11924_PERSON). Then, word embeddings are generated for all words occurring in the books corpus (with tagged entities). Then average vectors representing book's characters are calculated to create what we call a person embedding. Place embeddings are set up in the same way.

To develop word2vec embeddings, we filtered out stop words and punctuations. We experimented with multiple word2vec parameters, including 100 or 300 dimensions and 5 or 10 window sizes. We created average book vectors (whether based on persons/ places or all words) and then using cosine similarity, we recommend the *k* most similar books to the user query, which is the average of training set's books vectors. Some books have no mentions of places; hence, we discarded them while ranking books. The results in the table below show that the best performing doc2vec from table 4.1[1] was statistically significantly better than all the presented approaches. Yet, an interesting observation is that person embeddings are performing better than embeddings of all words. One apparent reason behind it is that person embeddings are tagged with their book id, which makes them similar to doc2vec. Even though embeddings of persons and places do not give the best accuracy, they could be applied for users who wish to find books with similar characters to their favorites. To download the fictional character and place embeddings visit `https://tinyurl.com/ybn6uaya`.

[1] Slightly different performance due to the removal of randomly selected items from the ranking list

Table C.3: The accuracy of book recommendations based on different combinations of embeddings

Method	Precision@k	Recall@k
Doc2Vec_300_5	0.135604	0.301843
person_300_10	0.129155	0.286955
person&place_300_10	0.128701	0.285503
person_100_10	0.125341	0.278149
person&place_100_10	0.125159	0.277872
person_300_5	0.117711	0.263297
person_100_5	0.117348	0.262912
person&place_100_5	0.116349	0.261538
person&place_300_5	0.116258	0.261361
all_words_emb_100_10	0.108356	0.245038
all_words_emb_300_10	0.106812	0.240755
no_person_all_words_100_10	0.105995	0.238612
no_person_all_words_300_10	0.105813	0.237855
all_words_emb_100_5	0.105268	0.237189
all_words_emb_300_5	0.10445	0.234977
no_person_all_words_100_5	0.103906	0.23373
no_person_all_words_300_5	0.102634	0.230241
place_100_10	0.088011	0.198061
place_300_10	0.083197	0.188456
place_100_5	0.074024	0.168023
place_300_5	0.069846	0.158104

Bibliography

F. Abel, Q. Gao, G.-J. Houben, and K. Tao. Analyzing user modeling on twitter for personalized news recommendations. In *Proc. 19th International Conference on User Modeling, Adaption, and Personalization*, UMAP'11, pages 1–12, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-22361-7. URL `http://dl.acm.org/citation.cfm?id=2021855.2021857`.

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*, 17(6):734–749, June 2005. ISSN 1041-4347. doi: 10.1109/TKDE.2005.99.

R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc. ISBN 1-55860-153-8. URL `http://dl.acm.org/citation.cfm?id=645920.672836`.

H. Alharthi and D. Inkpen. Study of Linguistic Features Incorporated in a Literary Book Recommender System. In *Proceedings of the 34th ACM/SIGAPP Symposium On Applied Computing (SAC '19)*, New York, NY, USA, April 2019. ACM. doi: https://doi.org/10.1145/3297280.3297382.

H. Alharthi, D. Inkpen, and S. Szpakowicz. Unsupervised topic modelling in a book recommender system for new users. In *SIGIR 2017 Workshop on eCommerce (ECOM17)*, 2017. ISBN 978-1-4503-5022-8. doi: 10.1145/3077136.3084367. URL http://doi.acm.org/10.1145/3077136.3084367.

H. Alharthi, D. Inkpen, and S. Szpakowicz. A survey of book recommender systems. In *Journal of Intelligent Information Systems*, volume 51, pages 139–160. Springer, Aug 2018a. doi: 10.1007/s10844-017-0489-9. URL https://doi.org/10.1007/s10844-017-0489-9.

H. Alharthi, D. Inkpen, and S. Szpakowicz. Authorship identification for literary book recommendations. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 390–400, 2018b. URL https://aclanthology.info/papers/C18-1033/c18-1033.

R. G. Apaza, E. V. Cervantes, L. C. Quispe, and J. O. Luna. Online courses recommendation based on lda, 2014.

M. C. Ardanuy and C. Sporleder. Clustering of Novels Represented as Social Networks. *LiLT (Linguistic Issues in Language Technology)*, 12(4), 2016. URL http://csli-lilt.stanford.edu/ojs/index.php/LiLT.

S. Argamon, M. Koppel, J. W. Pennebaker, and J. Schler. Automatically profiling the author of an anonymous text. *Commun. ACM*, 52(2):119–123, Feb. 2009. ISSN 0001-0782. doi: 10.1145/1461928.1461959. URL http://doi.acm.org/10.1145/1461928.1461959.

B. Athiwaratkun and K. Kang. Feature representation in convolutional neural networks. *CoRR*, abs/1507.02313, 2015.

M. M. Bakhtin. Forms of time and of the chronotope in the novel. In M. Holquist, editor, *The Dialogic Imagination: Four Essays*, pages 84–258+. University of Texas Press, 1981. URL http://books.google.com/books?id=JKZztxqdIpgC.

D. Basak, S. Pal, and D. C. Patranabis. Support Vector Regression. *Neural Information Processing – Letters and Reviews*, 11(10):478–486, 2007.

D. Ben-Shimon, A. Tsikinovsky, L. Rokach, A. Meisles, G. Shani, and L. Naamani. Recommender system from personal social networks. In K. M. Wegrzyn-Wolska and P. S. Szczepaniak, editors, *Advances in Intelligent Web Mastering: Proc. 5th Atlantic Web Intelligence Conference – AWIC'2007, Fontainbleau, France, June 25 – 27, 2007*, pages 47–55. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-72575-6. doi: 10.1007/978-3-540-72575-6_8. URL http://dx.doi.org/10.1007/978-3-540-72575-6_8.

S. Bergamaschi and L. Po. Comparing lda and lsa topic models for content-based movie recommendation systems. In V. Monfort and K.-H. Krempels, editors, *Web Information Systems and Technologies: 10th International Conference, WEBIST 2014, Barcelona, Spain, April 3-5, 2014, Revised Selected Papers*, pages 247–263. Springer International Publishing, Cham, 2015. ISBN 978-3-319-27030-2. doi: 10.1007/978-3-319-27030-2_16. URL http://dx.doi.org/10.1007/978-3-319-27030-2_16.

G. S. Berns, K. Blaine, M. J. Prietula, and B. E. Pye. Short- and Long-Term Effects of a Novel on Connectivity in the Brain. *Brain Connectivity*, 3(6):590–600, 2013.

D. Billsus and M. J. Pazzani. A hybrid user model for news story classification. In J. Kay, editor, *UM99 User Modeling*, pages 99–108, Vienna, 1999. Springer Vienna. ISBN 978-3-7091-2490-1.

D. Billsus and M. J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2-3):147–180, Feb. 2000. ISSN 0924-1868. doi: 10.1023/A:1026501525781. URL http://dx.doi.org/10.1023/A:1026501525781.

S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009. ISBN 0596516495, 9780596516499.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=944919.944937.

M. Braunhofer, I. Fernández-Tobías, and F. Ricci. Parsimonious and adaptive contextual information acquisition in recommender systems. In *Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems co-located with ACM Conference on Recommender Systems (RecSys 2015)*, 2015.

L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, Oct. 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

C. S. Brinegar. Mark twain and the quintus curtius snodgrass letters: A statistical test of authorship. *Journal of the American Statistical Association*, 58(301):85–96, 1963. ISSN 01621459. URL http://www.jstor.org/stable/2282956.

J. Brooke and G. Hirst. Hybrid models for lexical acquisition of correlated styles. In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan,* *October 14-18, 2013*, pages 82–90, 2013. URL http://aclweb.org/anthology/I/I13/I13-1010.pdf.

J. Brooke and G. Hirst. Supervised ranking of co-occurrence profiles for acquisition of continuous lexical attributes. In *COLING*, 2014.

J. Brooke, A. Hammond, and G. Hirst. Gutentag: an nlp-driven tool for digital humanities research in the project gutenberg corpus. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature, CLfL@NAACL-HLT 2015, June 4, 2015, Denver, Colorado, USA*, pages 42–47, 2015. URL http://aclweb.org/anthology/W/W15/W15-0705.pdf.

J. Brooke, A. Hammond, and T. Baldwin. Bootstrapped text-level named entity recognition for literature. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*, 2016. URL http://aclweb.org/anthology/P/P16/P16-2056.pdf.

J. Brooke, A. Hammond, and G. Hirst. Using models of lexical style to quantify free indirect discourse in modernist fiction. *DSH*, 32:234–250, 2017.

D. Brown and A. Krog. Creative non-fiction: A conversation. *Current Writing: Text and Reception in Southern Africa*, 23(1):57–70, 2011. doi: 10.1080/1013929X.2011.572345. URL https://doi.org/10.1080/1013929X.2011.572345.

L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

K. C. Beyard-Tyler and H. J. Sullivan. Adolescent reading preferences for type of theme and sex of character. *Reading Research Quarterly*, 16:104, 01 1980. doi: 10.2307/747350.

E. Castillejo, A. Almeida, and D. López-De-Ipiña. Alleviating cold-user start problem with users' social network data in recommendation systems. In *Workshop on Problems and Applications in AI, ECAI-12*, 2012.

G. Chen. A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation. *ArXiv e-prints*, Oct. 2016.

J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. Short and tweet: Experiments on recommending content from information streams. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1185–1194. ACM, 2010. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753503. URL `http://doi.acm.org/10.1145/1753326.1753503`.

Y.-F. Chen. Herd behavior in purchasing books online. *Computers in Human Behavior*, 24(5):1977–1992, 2008. ISSN 0747-5632. doi: http://dx.doi.org/10.1016/j.chb.2007.08.004. URL `http://www.sciencedirect.com/science/article/pii/S0747563207001458`. Including the Special Issue: Internet Empowerment.

X. Cheng, J. Guo, S. Liu, Y. Wang, and X. Yan. Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In *Proc. SIAM International Conference on Data Mining*, pages 749–757. SIAM, 2013. ISBN 978-1-61197-283-2.

D. Choi, J. Kim, E. Lee, C. Choi, J. Hong, and P. Kim. Research for the pattern analysis of individual interest using sns data: Focusing on facebook. In *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 36–40, July 2014. doi: 10.1109/IMIS.2014.94.

A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou. Word translation without parallel data. *CoRR*, abs/1710.04087, 2018.

G. De Francisci Morales, A. Gionis, and C. Lucchese. From chatter to headlines: Harnessing the real-time web for personalized news recommendation. In *Proc. Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 153–162. ACM, 2012. ISBN 978-1-4503-0747-5. doi: 10.1145/2124295.2124315. URL `http://doi.acm.org/10.1145/2124295.2124315`.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, pages 391–407, 1990.

Y. Deldjoo, M. Elahi, P. Cremonesi, F. Garzotto, P. Piazzolla, and M. Quadrana. Content-based video recommendation system based on stylistic visual features. *Journal on Data Semantics*, 5(2):99–113, Jun 2016. ISSN 1861-2040. doi: 10.1007/s13740-016-0060-9. URL `https://doi.org/10.1007/s13740-016-0060-9`.

L. Derczynski, A. Ritter, S. Clark, and K. Bontcheva. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proc. International Conference on Recent Advances in Natural Language Processing*. Association for Computational Linguistics, 2013.

M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, Jan. 2004. ISSN 1046-8188. doi: 10.1145/963770.963776. URL `http://doi.acm.org/10.1145/963770.963776`.

D. K. Elson, N. Dames, and K. R. McKeown. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguis-*

tics, ACL '10, pages 138–147, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1858681.1858696.

I. Fernández-Tobías, M. Braunhofer, M. Elahi, F. Ricci, and I. Cantador. Alleviating the new user problem in collaborative filtering by exploiting personality information. *User Modeling and User-Adapted Interaction*, 26(2):221–255, Jun 2016. ISSN 1573-1391. doi: 10.1007/s11257-016-9172-z. URL https://doi.org/10.1007/s11257-016-9172-z.

P. A. Flach and N. Lachiche. Confirmation-guided Discovery of First-order Rules with Tertius. *Machine Learning*, 42(1/2):61–95, Jan. 2001. ISSN 0885-6125. doi: 10.1023/A:1007656703224. URL http://dx.doi.org/10.1023/A:1007656703224.

L. Flekova and I. Gurevych. Personality profiling of fictional characters using sense-level links between lexical resources. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1805–1816. Association for Computational Linguistics, 2015. doi: 10.18653/v1/D15-1208. URL http://aclanthology.coli.uni-saarland.de/pdf/D/D15/D15-1208.pdf.

W. Fucks. On mathematical analysis of style. *Biometrika*, 39(1/2):122–129, 1952. ISSN 00063444. URL http://www.jstor.org/stable/2332470.

A. L. Garrido, M. G. Buey, S. Escudero, S. Ilarri, E. Mena, and S. B. Silveira. TM-Gen: A Topic Map Generator from Text Documents. In *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, pages 735–740, Nov 2013. doi: 10.1109/ICTAI.2013.113.

A. L. Garrido, M. S. Pera, and S. Ilarri. SOLE-R: A Semantic and Linguistic Approach for Book Recommendations. In *Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on*, pages 524–528, July 2014. doi: 10.1109/ICALT.2014.155.

P. Geurts and G. Louppe. Learning to rank with extremely randomized trees. In O. Chapelle, Y. Chang, and T.-Y. Liu, editors, *Proceedings of the Learning to Rank Challenge*, volume 14 of *Proceedings of Machine Learning Research*, pages 49–61, Haifa, Israel, 25 Jun 2011. PMLR. URL `http://proceedings.mlr.press/v14/geurts11a.html`.

P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63 (1):3–42, Apr 2006. ISSN 1573-0565. doi: 10.1007/s10994-006-6226-1. URL `https://doi.org/10.1007/s10994-006-6226-1`.

M. Girolami and A. Kabán. On an equivalence between plsi and lda. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 433–434, New York, NY, USA, 2003. ACM. ISBN 1-58113-646-3. doi: 10.1145/860435.860537. URL `http://doi.acm.org/10.1145/860435.860537`.

S. Givon and V. Lavrenko. Predicting Social-tags for Cold Start Book Recommendations. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 333–336, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-435-5. doi: 10.1145/1639714.1639781. URL `http://doi.acm.org/10.1145/1639714.1639781`.

D. Godfrey, C. Johns, C. D. Meyer, S. Race, and C. Sadek. A case study in text mining: Interpreting twitter data from world cup tweets. *CoRR*, abs/1408.5427, 2014. URL `http://arxiv.org/abs/1408.5427`.

F. Godin, V. Slavkovikj, W. De Neve, B. Schrauwen, and R. Van de Walle. Using topic models for twitter hashtag recommendation. In *Proc. 22Nd International Conference on World Wide Web*, WWW '13 Companion, pages 593–596. ACM, 2013. ISBN 978-1-

4503-2038-2. doi: 10.1145/2487788.2488002. URL http://doi.acm.org/10.1145/2487788.2488002.

Y. Goldberg. A primer on neural network models for natural language processing. *J. Artif. Intell. Res.*, 57:345–420, 2016.

S. Grayson, M. Mulvany, K. Wade, G. Meaney, and D. Greene. Novel2vec: Characterising 19th century fiction via word embeddings. In *Proceedings of the 24th Irish Conference on Artificial Intelligence and Cognitive Science, AICS 2016, Dublin, Ireland, September 20-21, 2016.*, pages 68–79, 2016. URL http://ceur-ws.org/Vol-1751/AICS_2016_paper_48.pdf.

S. R. Gunn. Support vector machines for classification and regression, 1998.

S. Gupta and V. Varma. Scientific article recommendation by using distributed representations of text and graph. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 1267–1268, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-4914-7. doi: 10.1145/3041021.3053062. URL https://doi.org/10.1145/3041021.3053062.

D. F. Gurini, F. Gasparetti, A. Micarelli, and G. Sansonetti. A sentiment-based approach to twitter user recommendation. In *Proceedings of the Fifth ACM RecSys Workshop on Recommender Systems and the Social Web co-located with the 7th ACM Conference on Recommender Systems (RecSys 2013), Hong Kong, China, October 13, 2013.*, 2013. URL http://ceur-ws.org/Vol-1066/Paper5.pdf.

I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel. Social media recommendation based on people and tags. In *Proc. 33rd International ACM SIGIR Conference on Research* *and Development in Information Retrieval*, SIGIR '10, pages 194–201. ACM, 2010. ISBN 978-1-4503-0153-4. doi: 10.1145/1835449.1835484. URL `http://doi.acm.org/10.1145/1835449.1835484`.

M. Hahsler, B. Grün, and K. Hornik. arules – A Computational Environment for Mining Association Rules and Frequent Item Sets. *JSS Journal of Statistical Software*, 14(15):1–25, 2005. ISSN 1548-7660. doi: 10.18637/jss.v014.i15.

X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 173–182, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-4913-0. doi: 10.1145/3038912.3052569. URL `https://doi.org/10.1145/3038912.3052569`.

X. He, Z. He, X. Du, and T.-S. Chua. Adversarial personalized ranking for recommendation. In *The 41st International ACM SIGIR Conference on Research &#38; Development in Information Retrieval*, SIGIR '18, pages 355–364, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5657-2. doi: 10.1145/3209978.3209981. URL `http://doi.acm.org/10.1145/3209978.3209981`.

Y. Hijikata, Y. Kai, and S. Nishida. The relation between user intervention and user satisfaction for information recommendation. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 2002–2007, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-0857-1. doi: 10.1145/2245276.2232109. URL `http://doi.acm.org/10.1145/2245276.2232109`.

K. Hill. The Arts and Individual Well-Being in Canada, February 2013. URL http://www.hillstrategies.com/content/arts-and-individual-well-being-canada. [Online; posted 13 February 2013].

P. J. Holcomb, J. Kounios, J. E. Anderson, and W. C. West. Dual-coding, context-availability, and concreteness effects in sentence comprehension: an electrophysiological investigation. *Journal of experimental psychology. Learning, memory, and cognition*, 25 3:721–42, 1999.

D. I. Holmes. The analysis of literary style–a review. *Journal of the Royal Statistical Society. Series A (General)*, 148(4):328–341, 1985. ISSN 00359238. URL http://www.jstor.org/stable/2981893.

D. I. Holmes. A stylometric analysis of mormon scripture and related texts. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 155(1):91–120, 1992. ISSN 09641998, 1467985X. URL http://www.jstor.org/stable/2982671.

Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, Dec 2008. doi: 10.1109/ICDM.2008.22.

Z. Huang, W. Chung, T.-H. Ong, and H. Chen. A graph-based recommender system for digital library. In *Proceedings of the 2Nd ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '02, pages 65–73, New York, NY, USA, 2002. ACM. ISBN 1-58113-513-0. doi: 10.1145/544220.544231. URL http://doi.acm.org/10.1145/544220.544231.

I. Hussain and P. Munshi. Identifying reading preferences of secondary school students. *Creative Education*, 2:6, 2011. ISSN 5. doi: 10.4236/ce.2011.25062. URL http://www.scirp.org/journal/CTA.aspx?paperID=8853.

H. Jeremy. *A glossary of contemporary literary theory*. An Arnold Publication Series. Bloomsbury Academic, 2000. ISBN 9780340761953. URL https://books.google.ca/books?id=oFc6tKi_Bm8C.

M. L. Jockers and D. Mimno. Significant themes in 19th-century literature. *Poetics*, 41(6): 750–769, Dec. 2013. ISSN 0304422X. doi: 10.1016/j.poetic.2013.08.005. URL http://dx.doi.org/10.1016/j.poetic.2013.08.005.

R. Johnson and T. Zhang. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112. Association for Computational Linguistics, 2015. doi: 10.3115/v1/N15-1011. URL http://www.aclweb.org/anthology/N15-1011.

N. Jonnalagedda, S. Gauch, K. Labille, and S. Alfarhood. Incorporating popularity in a personalized news recommender system. *PeerJ Computer Science*, 2:e63, 2016.

M. Kaminskas and D. Bridge. Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Trans. Interact. Intell. Syst.*, 7(1):2:1–2:42, Dec. 2016. ISSN 2160-6455. doi: 10.1145/2926720. URL http://doi.acm.org/10.1145/2926720.

H. Kapusuzoglu and S. G. Öguducu. A Relational Recommender System Based on Domain Ontology. In *Emerging Intelligent Data and Web Technologies (EIDWT), 2011 International Conference on*, pages 36–41, Sept 2011. doi: 10.1109/EIDWT.2011.15.

Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of* *the ACL*, pages 1746–1751, 2014. URL http://aclweb.org/anthology/D/D14/D14-1181.pdf.

J. Koberstein and Y.-K. Ng. Using Word Clusters to Detect Similar Web Documents. In *Proceedings of the First International Conference on Knowledge Science, Engineering and Management*, KSEM'06, pages 215–228, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-37033-1, 978-3-540-37033-8. doi: 10.1007/11811220_19. URL http://dx.doi.org/10.1007/11811220_19.

A. Kohrs and B. Mérialdo. Improving collaborative filtering for new-users by smart object selection. In *Proc. International Conference on Media Futures (ICME 2001)*, Florence, Italy, May 2001.

D. Kokkinakis and M. Malm. Character profiling in 19th century fiction. In *Workshop: Language Technologies for Digital Humanities and Cultural Heritage in conjunction with the Recent Advances in Natural Language Processing (RANLP). Hissar, Bulgaria.*, 2011.

M. Kompan and M. Bieliková. *Content-Based News Recommendation*, pages 61–72. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-15208-5. doi: 10.1007/978-3-642-15208-5_6. URL https://doi.org/10.1007/978-3-642-15208-5_6.

Y. Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proc. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 426–434. ACM, 2008. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401944. URL http://doi.acm.org/10.1145/1401890.1401944.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009. ISSN 0018-9162. doi: 10.1109/MC.2009.263. URL http://dx.doi.org/10.1109/MC.2009.263.

T. Landauer, P. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse processes*, 25:259–284, 1998.

J. H. Lau and T. Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 78–86. Association for Computational Linguistics, 2016. doi: 10.18653/v1/W16-1609. URL `http://www.aclweb.org/anthology/W16-1609`.

Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1188–II–1196. JMLR.org, 2014. URL `http://dl.acm.org/citation.cfm?id=3044805.3045025`.

P. Lops, M. Gemmis, and G. Semeraro. Content-based Recommender Systems: State of the Art and Trends. In F. Ricci, L. Rokach, B. Shapira, and B. P. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer US, Boston, MA, 2011. ISBN 978-0-387-85820-3. doi: 10.1007/978-0-387-85820-3_3. URL `http://dx.doi.org/10.1007/978-0-387-85820-3_3`.

G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts. Understanding variable importances in forests of randomized trees. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, pages 431–439, USA, 2013. Curran Associates Inc. URL `http://dl.acm.org/citation.cfm?id=2999611.2999660`.

Q. Lu, T. Chen, W. Zhang, D. Yang, and Y. Yu. Serendipitous personalized ranking for top-n recommendation. In *Proc. The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '12, pages 258–

265, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-0-7695-4880-7. URL `http://dl.acm.org/citation.cfm?id=2457524.2457692`.

S. Maneewongvatana and S. Maneewongvatana. A recommendation model for personalized book lists. In *Communications and Information Technologies (ISCIT), 2010 International Symposium on*, pages 389–394, Oct 2010. doi: 10.1109/ISCIT.2010.5664873.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

R. A. Mar and K. Oatley. The Function of Fiction is the Abstraction and Simulation of Social Experience. *Perspectives on Psychological Science*, 3(3):173–192, May 2008. ISSN 1745-6924. doi: 10.1111/j.1745-6924.2008.00073.x. URL `http://dx.doi.org.monstera.cc.columbia.edu:2048/10.1111/j.1745-6924.2008.00073.x`.

R. A. Mar, K. Oatley, and J. B. Peterson. Exploring the link between reading fiction and empathy: Ruling out individual differences and examining outcomes. *Communications*, 34 (4):407–428, Dec. 2009. ISSN 0341-2059. doi: 10.1515/comm.2009.025. URL `http://dx.doi.org/10.1515/comm.2009.025`.

P. Matuszyk, J. a. Vinagre, M. Spiliopoulou, A. M. Jorge, and J. a. Gama. Forgetting methods for incremental matrix factorization in recommender systems. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, pages 947–953, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3196-8. doi: 10.1145/2695664.2695820. URL `http://doi.acm.org/10.1145/2695664.2695820`.

J. McAuley and J. Leskovec. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *Proceedings of the 7th ACM Conference on Recommender*

Systems, RecSys '13, pages 165–172, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2409-0. doi: 10.1145/2507157.2507163. URL http://doi.acm.org/10.1145/2507157.2507163.

D. Mican, L. Mocean, and N. Tomai. *Building a Social Recommender System by Harvesting Social Relationships and Trust Scores between Users*, pages 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-34228-8. doi: 10.1007/978-3-642-34228-8_1. URL https://doi.org/10.1007/978-3-642-34228-8_1.

M. Mikawa, S. Izumi, and K. Tanaka. Book Recommendation Signage System Using Silhouette-Based Gait Classification. In *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, volume 1, pages 416–419, Dec 2011. doi: 10.1109/ICMLA.2011.43.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc. URL http://dl.acm.org/citation.cfm?id=2999792.2999959.

R. J. Mooney and L. Roy. Content-based Book Recommending Using Learning for Text Categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, pages 195–204, New York, NY, USA, 2000. ACM. ISBN 1-58113-231-X. doi: 10.1145/336597.336662. URL http://doi.acm.org/10.1145/336597.336662.

L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, and Z. Jin. How Transferable are Neural Networks in NLP Applications? *ArXiv e-prints*, Mar. 2016.

C. Musto, C. Greco, A. Suglia, and G. Semeraro. Ask me any rating: A content-based recommender system based on recurrent neural networks. In *Proceedings of the 7th Italian Information Retrieval Workshop, Venezia, Italy, May 30-31, 2016.*, 2016. URL http://ceur-ws.org/Vol-1653/paper_11.pdf.

P. Nair, M. Moh, and T. S. Moh. Using social media presence for alleviating cold start problems in privacy protection. In *2016 International Conference on Collaboration Technologies and Systems (CTS)*, pages 11–17, Oct 2016. doi: 10.1109/CTS.2016.0022.

V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress. ISBN 978-1-60558-907-7. URL http://dl.acm.org/citation.cfm?id=3104322.3104425.

M. L. Newman, C. J. Groom, L. D. Handelman, and J. W. Pennebaker. Gender differences in language use: An analysis of 14,000 text samples. *Discourse Processes*, 45(3):211–236, 2008. doi: 10.1080/01638530802073712. URL https://doi.org/10.1080/01638530802073712.

R. Nicholsa, J. Lynn, and B. G. Purzycki. Toward a science of science fiction. *Scientific Study of Literature*, 4(1):25–45, 2014. ISSN e-issn 2210–4380. doi: doi10.1075/ssol.4.1.02nic.

S. Nikolenko. Svd-lda: Topic modeling for full-text recommender systems. In O. Pichardo Lagunas, O. Herrera Alcántara, and G. Arroyo Figueroa, editors, *Advances in Artificial Intelligence and Its Applications: 14th Mexican International Conference on Artificial Intelligence, MICAI 2015, Cuernavaca, Morelos, Mexico, October 25-31, 2015, Proceedings, Part II*, pages 67–79. Springer International Publishing, Cham, 2015. ISBN 978-3-319-

27101-9. doi: 10.1007/978-3-319-27101-9_5. URL http://dx.doi.org/10.1007/978-3-319-27101-9_5.

A. Odić, M. Tkalčič, A. Košir, and J. F. Tasič. A.: Relevant context in a movie recommender system: Users' opinion vs. statistical detection. In *In: Proc. of the 4th Workshop on Context-Aware Recommender Systems (2011*, 2011.

D. Pathak, S. Matharia, and C. N. S. Murthy. NOVA: Hybrid book recommendation engine. In *Advance Computing Conference (IACC), 2013 IEEE 3rd International*, pages 977–982, Feb 2013. doi: 10.1109/IAdCC.2013.6514359.

M. J. Pazzani and D. Billsus. Content-Based Recommendation Systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web: Methods and Strategies of Web Personalization*, pages 325–341. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-72079-9. doi: 10.1007/978-3-540-72079-9_10. URL http://dx.doi.org/10.1007/978-3-540-72079-9_10.

M. Pennacchiotti and S. Gurumurthy. Investigating topic models for social media user recommendation. In *Proc. 20th International Conference Companion on World Wide Web*, WWW '11, pages 101–102. ACM, 2011. ISBN 978-1-4503-0637-9. doi: 10.1145/1963192.1963244. URL http://doi.acm.org/10.1145/1963192.1963244.

J. W. Pennebaker, R. L. Boyd, K. Jordan, and K. Blackburn. The development and psychometric properties of liwc2015, 2015a.

J. W. Pennebaker, C. K. Chung, J. Frazee, G. M. Lavergne, and D. I. Beaver. When small words foretell academic success: The case of college admissions essays. *PLOS ONE*, 9 (12):1–10, 12 2015b. doi: 10.1371/journal.pone.0115844. URL https://doi.org/10.1371/journal.pone.0115844.

M. S. Pera. Using Online Data Sources to Make Recommendations on Reading Materials for K-12 and Advanced Readers, 2014. URL `http://search.proquest.com/docview/1545892907?accountid=14701`. Copyright – Copyright ProQuest, UMI Dissertations Publishing 2014; Last updated – 2015-08-21; First page – n/a.

M. S. Pera and Y.-K. Ng. With a Little Help from My Friends: Generating Personalized Book Recommendations Using Data Extracted from a Social Website. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology – Volume 01*, WI-IAT '11, pages 96–99, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4513-4. doi: 10.1109/WI-IAT.2011.9. URL `http://dx.doi.org/10.1109/WI-IAT.2011.9`.

M. S. Pera and Y.-K. Ng. What to Read Next?: Making Personalized Book Recommendations for K-12 Users. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 113–120, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2409-0. doi: 10.1145/2507157.2507181. URL `http://doi.acm.org/10.1145/2507157.2507181`.

M. S. Pera and Y. K. Ng. How Can We Help Our K-12 Teachers?: Using a Recommender to Make Personalized Book Suggestions. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) – Volume 02*, WI-IAT '14, pages 335–342, Washington, DC, USA, 2014a. IEEE Computer Society. ISBN 978-1-4799-4143-8. doi: 10.1109/WI-IAT.2014.116. URL `http://dx.doi.org/10.1109/WI-IAT.2014.116`.

M. S. Pera and Y.-K. Ng. Automating Readers' Advisory to Make Book Recommendations for K-12 Readers. In *Proceedings of the 8th ACM Conference on Recommender*

Systems, RecSys '14, pages 9–16, New York, NY, USA, 2014b. ACM. ISBN 978-1-4503-2668-1. doi: 10.1145/2645710.2645721. URL `http://doi.acm.org/10.1145/2645710.2645721`.

M. S. Pera and Y.-K. Ng. Analyzing Book-Related Features to Recommend Books for Emergent Readers. In *Proceedings of the 26th ACM Conference on Hypertext &#38; Social Media*, HT '15, pages 221–230, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3395-5. doi: 10.1145/2700171.2791037. URL `http://doi.acm.org/10.1145/2700171.2791037`.

M. S. Pera, N. Condie, and Y.-K. Ng. Personalized book recommendations created by using social media data. In *Proc. 2010 International Conference on Web Information Systems Engineering*, WISS'10, pages 390–403, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-24395-0. URL `http://dl.acm.org/citation.cfm?id=2044492.2044531`.

J. M. Phillips and S. Venkatasubramanian. A gentle introduction to the kernel distance. *CoRR*, abs/1103.1625, 2010.

K. Priyanka, A. S. Tewari, and A. G. Barman. Personalised book recommendation system based on opinion mining technique. In *Communication Technologies (GCCT), 2015 Global Conference on*, pages 285–289, April 2015. doi: 10.1109/GCCT.2015.7342668.

C. Qian, T. He, and R. Zhang. Deep learning based authorship identification, 2017.

M. K. M. Rahman, W. Pi Yang, T. W. S. Chow, and S. Wu. A Flexible Multi-layer Self-organizing Map for Generic Processing of Tree-structured Data. *Pattern Recognition*, 40 (5):1406–1424, May 2007. ISSN 0031-3203. doi: 10.1016/j.patcog.2006.10.010. URL `http://dx.doi.org/10.1016/j.patcog.2006.10.010`.

S. Rajpurkar, D. Bhatt, and P. Malhotra. Book Recommendation System. *IJIRST –International Journal for Innovative Research in Science & Technology*, 1(11):314–316, 2015.

A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl. Getting to know you: Learning new user preferences in recommender systems. In *Proc. 7th International Conference on Intelligent User Interfaces*, IUI '02, pages 127–134. ACM, 2002. ISBN 1-58113-459-2. doi: 10.1145/502716.502737. URL `http://doi.acm.org/10.1145/502716.502737`.

A. M. Rashid, G. Karypis, and J. Riedl. Learning preferences of new users in recommender systems: An information theoretic approach. *SIGKDD Explor. Newsl.*, 10(2):90–100, Dec. 2008. ISSN 1931-0145. doi: 10.1145/1540276.1540302. URL `http://doi.acm.org/10.1145/1540276.1540302`.

A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW '14, pages 512–519, Washington, DC, USA, 2014. IEEE Computer Society. ISBN 978-1-4799-4308-1. doi: 10.1109/CVPRW.2014.131. URL `http://dx.doi.org/10.1109/CVPRW.2014.131`.

F. Ricci, L. Rokach, and B. Shapira. Introduction to Recommender Systems Handbook. In F. Ricci, L. Rokach, B. Shapira, and B. P. Kantor, editors, *Recommender Systems Handbook*, pages 1–35. Springer US, Boston, MA, 2011. ISBN 978-0-387-85820-3. doi: 10.1007/978-0-387-85820-3_1. URL `http://dx.doi.org/10.1007/978-0-387-85820-3_1`.

C. S. Ross. Making Choices: What Readers Say about Choosing Books to Read for Pleasure. *The Acquisitions Librarian*, 13(25):5–21, 2000. doi: 10.1300/J101v13n25_02. URL `http://dx.doi.org/10.1300/J101v13n25_02`.

L. Safoury and A. Salah. Exploiting user demographic attributes for solving cold-start problem in recommender system. In *2nd International Conference on Software and Computer Applications*, Paris, 2 June 2013.

R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 791–798, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: 10.1145/1273496.1273596. URL `http://doi.acm.org/10.1145/1273496.1273596`.

B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM. ISBN 1-58113-348-0. doi: 10.1145/371920.372071. URL `http://doi.acm.org/10.1145/371920.372071`.

A. M. J. Schakel and B. J. Wilson. Measuring word significance using distributed representations of words. *CoRR*, abs/1508.02297, 2015.

G. Schröder, M. Thiele, and W. Lehner. Setting goals and choosing metrics for recommender system evaluations. In *Proceedings of the 5th ACM Conference On Recommender Systems*, Chicago, USA, January 2011.

S. Sedhain, S. Sanner, D. Braziunas, L. Xie, and J. Christensen. Social collaborative filtering for cold-start recommendations. In *Proc. 8th ACM Conference on Recommender Systems*, RecSys '14, pages 345–348. ACM, 2014. ISBN 978-1-4503-2668-1. doi: 10.1145/2645710.2645772. URL `http://doi.acm.org/10.1145/2645710.2645772`.

G. Shani and A. Gunawardana. Evaluating Recommendation Systems. In F. Ricci, L. Rokach, B. Shapira, and B. P. Kantor, editors, *Recommender Systems Handbook*, pages 257–297. Springer US, Boston, MA, 2011. ISBN 978-0-387-85820-3. doi: 10.1007/978-0-387-85820-3_8. URL `http://dx.doi.org/10.1007/978-0-387-85820-3_8`.

B. Shao, D. Wang, T. Li, and M. Ogihara. Music recommendation based on acoustic features and user access patterns. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(8):1602–1611, Nov 2009. ISSN 1558-7916. doi: 10.1109/TASL.2009.2020893.

L. A. Sherman. *Analytics of literature: a manual for the objective study of English prose and poetry*. Ginn, 1893. URL `http://hdl.handle.net/2027/uc1.%24b239859`.

M. Smith. Recent experiences and new developments of methods for the determination of authorship. *Association of Literary and Linguistic Computing Bulletin*, 11:73–82, 1983.

S. Smith, S. Warburton, and J. Rutledge. Connecting patrons with library materials: A readers' advisory crash course innovative libraries online conference, 2016. URL `https://www.statelibraryofiowa.org/ld/c-d/continuing-ed/iloc/iloc-2016/handouts/connecting-patrons/connecting-patrons.pdf`.

W. B. Smith. Curves of pauline and pseudo-pauline style i. *Unitarian Review*, pages 452—-60, 1888.

S. S. Sohail, J. Siddiqui, and R. Ali. Book recommendation system using opinion mining technique. In *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, pages 1609–1614, Aug 2013. doi: 10.1109/ICACCI.2013.6637421.

T. Solorio, P. Rosso, M. Montes-y-Gómez, P. Shrestha, S. Sierra, and F. A. González.

Convolutional neural networks for authorship attribution of short texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 669–674, 2017. URL `http://aclanthology.info/papers/E17-2106/convolutional-neural-networks-for-authorship-attribution-of-short-texts`.

E. Stamatatos. A Survey of Modern Authorship Attribution Methods. *J. Am. Soc. Inf. Sci. Technol.*, 60(3):538–556, Mar. 2009. ISSN 1532-2882. doi: 10.1002/asi.v60:3. URL `http://dx.doi.org/10.1002/asi.v60:3`.

X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4:2–4:2, Jan. 2009. ISSN 1687-7470. doi: 10.1155/2009/421425. URL `http://dx.doi.org/10.1155/2009/421425`.

C. Tanasescu, V. Kesarwani, and D. Inkpen. Metaphor detection by deep learning and the place of poetic metaphor in digital humanities. In *Proceedings of the Thirty-First International Florida Artificial Intelligence Research Society Conference, FLAIRS 2018, Melbourne, Florida, USA. May 21-23 2018.*, pages 122–127, 2018. URL `https://aaai.org/ocs/index.php/FLAIRS/FLAIRS18/paper/view/17704`.

J. Tang, X. Hu, and H. Liu. Social recommendation: a review. *Social Network Analysis and Mining*, 3(4):1113–1133, 2013. ISSN 1869-5469. doi: 10.1007/s13278-013-0141-9. URL `http://dx.doi.org/10.1007/s13278-013-0141-9`.

Y. Tausczik and J. Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54, 3 2010. ISSN 0261-927X. doi: 10.1177/0261927X09351676.

A. J.-P. Tixier. Notes on Deep Learning for NLP. *ArXiv e-prints*, Aug. 2018.

M. Tkalčič, A. Košir, and J. Tasič. The ldos-peraff-1 corpus of facial-expression video clips with affective, personality and user-interaction metadata. *Journal on Multimodal User Interfaces*, 7(1):143–155, 2013. ISSN 1783-8738. doi: 10.1007/s12193-012-0107-7. URL `http://dx.doi.org/10.1007/s12193-012-0107-7`.

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073478. URL `http://dx.doi.org/10.3115/1073445.1073478`.

K. Tsuji, N. Takizawa, S. Sato, U. Ikeuchi, A. Ikeuchi, F. Yoshikane, and H. Itsumura. Book recommendation based on library loan records and bibliographic information. *Procedia - Social and Behavioral Sciences*, 147(Supplement C):478 – 486, 2014. ISSN 1877-0428. doi: https://doi.org/10.1016/j.sbspro.2014.07.142. URL `http://www.sciencedirect.com/science/article/pii/S1877042814040531`. 3rd International Conference on Integrated Information (IC-ININFO).

V. Vapnik, S. E. Golowich, and A. Smola. Support Vector Method for Function Approximation, Regression Estimation and Signal Processing. In *Proc. 9th International Conference on Neural Information Processing Systems*, NIPS'96, pages 281–287, Cambridge, MA, USA, 1996. MIT Press. URL `http://dl.acm.org/citation.cfm?id=2998981.2999021`.

P. C. Vaz, D. Martins de Matos, and B. Martins. Stylometric Relevance-feedback Towards a Hybrid Book Recommendation Algorithm. In *Proceedings of the Fifth ACM Work-*

shop on Research Advances in Large Digital Book Repositories and Complementary Media, BooksOnline '12, pages 13–16, New York, NY, USA, 2012a. ACM. ISBN 978-1-4503-1714-6. doi: 10.1145/2390116.2390125. URL http://doi.acm.org/10.1145/2390116.2390125.

P. C. Vaz, D. Martins de Matos, B. Martins, and P. Calado. Improving a Hybrid Literary Book Recommendation System Through Author Ranking. In *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '12, pages 387–388, New York, NY, USA, 2012b. ACM. ISBN 978-1-4503-1154-0. doi: 10.1145/2232817.2232904. URL http://doi.acm.org/10.1145/2232817.2232904.

P. C. Vaz, R. Ribeiro, and D. M. de Matos. LitRec vs. Movielens – A comparative study. In *KDIR 2012 – Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Barcelona, Spain, 4-7 October, 2012*, pages 370–373, 2012c.

P. C. Vaz, R. Ribeiro, and D. M. de Matos. Understanding Temporal Dynamics of Ratings in the Book Recommendation Scenario. In *Proceedings of the 2013 International Conference on Information Systems and Design of Communication*, ISDOC '13, pages 11–15, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2299-7. doi: 10.1145/2503859.2503862. URL http://doi.acm.org/10.1145/2503859.2503862.

C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proc. 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 448–456. ACM, 2011. ISBN 978-1-4503-0813-7. doi: 10.1145/2020408.2020480. URL http://doi.acm.org/10.1145/2020408.2020480.

J. Wang, C. Man, Y. Zhao, and F. Wang. An answer recommendation algorithm for medical community question answering systems. In *2016 IEEE International Conference on Service*

Operations and Logistics, and Informatics (SOLI), pages 139–144, July 2016. doi: 10.1109/SOLI.2016.7551676.

X. Wang, X. He, L. Nie, and T. Chua. Item silk road: Recommending items from information domains to social users. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 185–194, 2017. doi: 10.1145/3077136.3080771. URL `http://doi.acm.org/10.1145/3077136.3080771`.

Y. Wang, N. Stash, L. Aroyo, L. Hollink, and G. Schreiber. Using semantic relations for content-based recommender systems in cultural heritage. In *Proceedings of the 2009 International Conference on Ontology Patterns - Volume 516*, WOP'09, pages 16–28, Aachen, Germany, Germany, 2009. CEUR-WS.org. URL `http://dl.acm.org/citation.cfm?id=2889761.2889763`.

A. T. Wibowo, A. Siddharthan, J. Masthoff, and C. Lin. Incorporating constraints into matrix factorization for clothes package recommendation. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, UMAP '18, pages 111–119, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5589-6. doi: 10.1145/3209219.3209228. URL `http://doi.acm.org/10.1145/3209219.3209228`.

B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolutional network. *CoRR*, abs/1505.00853, 2015.

X. Yan, J. Guo, S. Liu, X.-q. Cheng, and Y. Wang. Clustering short text using ncut-weighted non-negative matrix factorization. In *Proc. 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2259–2262. ACM, 2012. ISBN 978-1-

4503-1156-4. doi: 10.1145/2396761.2398615. URL http://doi.acm.org/10.1145/2396761.2398615.

X. Yang, H. Zeng, and W. Huang. ARTMAP-Based Data Mining Approach and Its Application to Library Book Recommendation. In *Intelligent Ubiquitous Computing and Education, 2009 International Symposium on*, pages 26–29, May 2009. doi: 10.1109/IUCE.2009.43.

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3320–3328, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969197.

G. U. Yule. On sentence-length as a statistical characteristic of style in prose: With application to two cases of disputed authorship. *Biometrika*, 30(3/4):363–390, 1939. ISSN 00063444. URL http://www.jstor.org/stable/2332655.

H. Zhang and T. W. S. Chow. Organizing Books and Authors by Multilayer SOM. *Neural Networks and Learning Systems, IEEE Transactions on*, PP(99):1–14, 2015. ISSN 2162-237X. doi: 10.1109/TNNLS.2015.2496281.

S. Zhang, L. Yao, and A. Sun. Deep Learning based Recommender System: A Survey and New Perspectives. *ArXiv e-prints*, July 2017.

M. Zhou. Book recommendation based on web social network. In *Artificial Intelligence and Education (ICAIE), 2010 International Conference on*, pages 136–139, Oct 2010. doi: 10.1109/ICAIE.2010.5641415.

Z. Zhu and J. yan Wang. Book Recommendation Service by Improved Association Rule Mining Algorithm. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 7, pages 3864–3869, Aug 2007. doi: 10.1109/ICMLC.2007.4370820.

C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving Recommendation Lists Through Topic Diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 22–32, New York, NY, USA, 2005. ACM. ISBN 1-59593-046-9. doi: 10.1145/1060745.1060754. URL `http://doi.acm.org/10.1145/1060745.1060754`.