Unsupervised Topic Modelling in a Book Recommender System for New Users

Haifa Alharthi & Diana Inkpen & Stan Szpakowicz
EECS, University of Ottawa, Ottawa, Canada
halha060@uottawa.ca,Diana.Inkpen@uottawa.ca,szpak@eecs.uottawa.ca

ABSTRACT

Book recommender systems (RSs) are useful in libraries, schools and e-commerce applications. To our knowledge, no book RS exploits social networks other than book-cataloguing websites. We propose a recommendation component that learns the user's interests from social media data and recommends books accordingly. Our new method of modelling users' interests acquires a user's distinctive topics using tf-idf and represents them as word embeddings. Even though the system is designed to complement other systems, we evaluated it against content-based RS, a traditional book RS, and obtained similar performance. So, the system's new user would receive recommendation as accurate as current users.

KEYWORDS

recommender systems, personalization, user modelling, social media, Twitter

ACM Reference format:
Haifa Alharthi & Diana Inkpen & Stan Szpakowicz. 2017. Unsupervised Topic Modelling in a Book Recommender System for New Users. In *Proceedings of ACM Conference, Tokyo, Japan, August 2017 (SIGIR 2017 eCom),* 8 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The information flood on the Internet makes desirable a wide variety of applications, among them recommender systems (RSs). They help limit users' choices to the possibly most preferred items. To make suggestions, existing RSs exploit users' rating history, product features, user social-media content and relationships, user personality and emotions, and more.

Investigating book RSs is a worthwhile endeavour. They are useful in libraries, schools and e-learning portals, as well as bookstores and e-commerce applications. They can help libraries with abundant unused resources—*e.g.,* 75% of the books in the library of Changsha University of Science and Technology have never been checked out [48]. The practice of reading for pleasure has declined in recent years, especially among children.[1] This decline may affect life quality: readers may be significantly more likely than non-readers to report better health/mental health, to volunteer and feel strongly satisfied with life [18]. Exposure to fiction also correlates with higher ability of communication, empathy and social support [27, 28].

One challenge facing RSs is the user cold start. It happens when new users with no rating history are introduced to the system. A book RS may consider non-readers as new users and recommend books to them, and so help encourage the practice of reading. An issue related to cold start is the lack of explicit feedback from existing users, who may find it burdensome to assign ratings to items.

This paper proposes an automatic personalization module that learns the users' interests from social media data and recommends books accordingly. The Topic-Model-Based book recommendation component (TMB) would help existing RSs deal with new users with no user rating history. For each user, a topic profile is created that summarizes subjects discussed on her social media account. 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, a social book cataloging Web site. We compared the top k recommendations made by TMB and content-based system (CB). Both retrieved a comparable number of books, even though CB relied on users' rating history while TMB only needed their social profiles. We conclude that new users would receive recommendations (made by TMB) as accurate as those for current users (made by CB).

Content-based RS is a standard RS which is widely used for book recommendations [22, 31, 35, 44]. A CB recommender is a classifier that learns the patterns and similarities in the purchase history of one user to predict her future interests.

Many book RSs exploit social media, as explained in section 2, but they are all focused on social networks established mainly for readers such as LibraryThing. Our research investigates the use of a general platform, Twitter, to make book recommendations. Here is why: Twitter is not exclusive to bookworms, so it can help address the issue of new users who have no reading profiles. Moreover, since its establishment, Twitter has been used to survey opinions, report news (more than 85% of the Twitter activities are related to news events), raise awareness, create social and political movements and more; topics discussed on this medium have wide diversity and are up-to-date [11]. This offers a chance to understand the reactions and opinions of active users to their surroundings, *e.g.,* the social and political scene. It allows the capturing of broad topics that the user cares about and may not read about them yet which may help with the over-specialization issue, users receiving non-diverse recommendation lists.

The remainder of the paper is organized as follows. Section 2 summarizes the related work especially in the domains of social RSs for books and news articles. Section 3 gives a high-level description of the system and its components. Section 4 explains the details

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

SIGIR 2017 eCom, Tokyo, Japan

of data collection and preprocessing, as well as the system implementation. Section 5 defines the experiment settings and illustrates its results. Section 6 discusses the results. Section 7 concludes and suggests future work.

2 RELATED WORK

2.1 Topic Modelling of Text in RSs

Topic models have helped estimate preferences in many RSs. To name a few, recommendations were based on the topics extracted from movie plots [4], articles [33, 45], online courses syllabi [2] and trending categories on e-commerce portals [19]. Unlike the previously mentioned work which mainly analyzes textual description of items, TMB model the topics discussed in a user profile to capture their interests and make recommendations accordingly.

Based on topics learned from users' Twitter accounts, RSs could suggest hashtags [16] and friends [36]. 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).

2.2 Social media and the new user issue

Social media have been a great resource to "warm up" the user cold start. A users connections on social network were exploited in [7], [40], [17], [3] and [29]. In addition to using Facebook friends lists, [40] analyzed users' demographics and pages liked by a user.

[32] 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 favourite genres. Later, movies from the most frequent genres are recommended.

2.3 Recommendations of textual items

This section covers social RSs dedicated to recommending textual items, including books and news articles. First, we need to differentiate the characteristics of the book and the news recommendation tasks. News has a short lifetime and may become irrelevant within days or even hours. On the other hand, many books have survived hundreds of years and are still widely read and recommended. Furthermore, news content is dynamic and changes rapidly / daily. That requires the analysis of hashtags and entities such as names and places that may correspond with the news. However, books include broad aspects and are mostly unrelated to present names and actions. Thus, unlike news, book social RSs need to look for users' long-term interests.

2.3.1 Book recommendations using social media. LibraryThing is a social book-cataloguing Web site which allows users to form friendships and catalogue and tag books. [37] match books that a target user likes with books that her friends like. Each book in LibraryThing has a cloud of tags, and the system suggests the most similar tag-represented books, using a word correlation matrix. Another system also uses tags to find similar books in the user's friends list [38]. Books are considered similar when they share one or more tags with friends or are highly rated by a user's most reliable friends.

Another system that exploits LibraryThing, presented by [14], addresses the new item issue. Each book is characterized by tf-idf vectors of social tags (extracted from LibraryThing) and book tags (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 pure RM gives results similar to collaborative filtering. To our knowledge, no book RSs exploit social networks other than book-cataloguing websites.

2.3.2 Twitter-based news recommendations. To make news recommendations, [1] 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 names of people, places and other entities in addition to topics; there is a limitation to 18 different topics (*e.g.,* politics or sports). Entity-based user profiles scored the highest S@k (Success at rank k) at 0.20.

[8] 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 [1, 11, 20] 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 system does not require entity recognition or ontology development.

3 TOPIC-MODEL-BASED BOOK RECOMMENDER SYSTEM

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

3.1 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 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). We also experimented with Latent Dirichlet Allocation (LDA), but did not report the results due to the low performance. This supports the finding of a previous study [39] that NMF performs better than LDA when dealing with tweets.

3.1.1 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 timeline. Inverse document frequency (Equation 1) helps distinguish the terms that are specific to a user/document.

$$idf_t = log\frac{N}{df_t} \tag{1}$$

N is the number of users and df_t is the number of documents where term t occurs. Equation 2 defines the tf-idf weight of term t in document d.

$$tf\text{-}idf_{t,d} = tf_{t,d} * idf_t \tag{2}$$

The terms with highest weights are considered the tf-idf topic model [26].

3.1.2 Non-Negative Matrix Factorization. This dimensionality reduction and topic modelling technique has been found to work well with short text [9, 15, 47]. 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 represent 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 [15].

3.1.3 Topic embeddings. Word embeddings have gained a lot of attention lately thanks to the revival of neural networks. They are word vectors with fixed dimensions. We use the word2vec model, proposed by Mikolov et al. [30]. Terms in book and user profiles are mapped to word embeddings produced by the word2vec model. The model is trained on a very large amount of text and can predict the context of a given word. It represents words in a space where two words occurring in similar contexts are neighbours. We used pre-trained word embeddings developed on the Google News dataset of around 100 billion words. It comprises vectors of 300 dimensions for 3 million words and phrases.[2] Other available pre-trained models (*e.g.,* Global Vectors for Word Representation[3]) have been built using text from Twitter and Wikipedia, but the Google news embeddings are more relevant to both books and tweets. While books have formal descriptions, tweets are casual, with hashtags that require a model which encompasses abbreviations.

3.2 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 3) between UP_{u_i} and BP for every book in B_{u_i}, and suggests the books with k most similar BP.

$$similarity = \frac{UP_{u_i} \cdot BP_j}{\left\|UP_{u_i}\right\| * \left\|BP_j\right\|} \tag{3}$$

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.

4 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.

4.1 Data collection

We collected user data from Goodreads and Twitter, because there are no datasets with *both* users social profiles and their reading lists. The Twitter API was queried to retrieve any review shared by Goodreads users, and more than 1000 tweets were found, from which we accessed their authors and IDs. Twitter API allows the collection of a maximum of 3500 tweets per user. We gathered text, ID and date of creations of tweets for user with Goodreads review. 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 scrapper 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 ISBN, ISBN13, title, authors, language, the average rating of all reader, the number of pages, publisher, publication date, text review count and book description. The read date indicates the time of completion of a read book, while the added date is the time when a book was catalogued.

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 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 paper. 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 [41] which used a dataset of 52 users to test affective-based RS,and [34] which tested a context-aware RS on an 89-user dataset.

[2] https://code.google.com/archive/p/word2vec/
[3] http://nlp.stanford.edu/projects/glove/

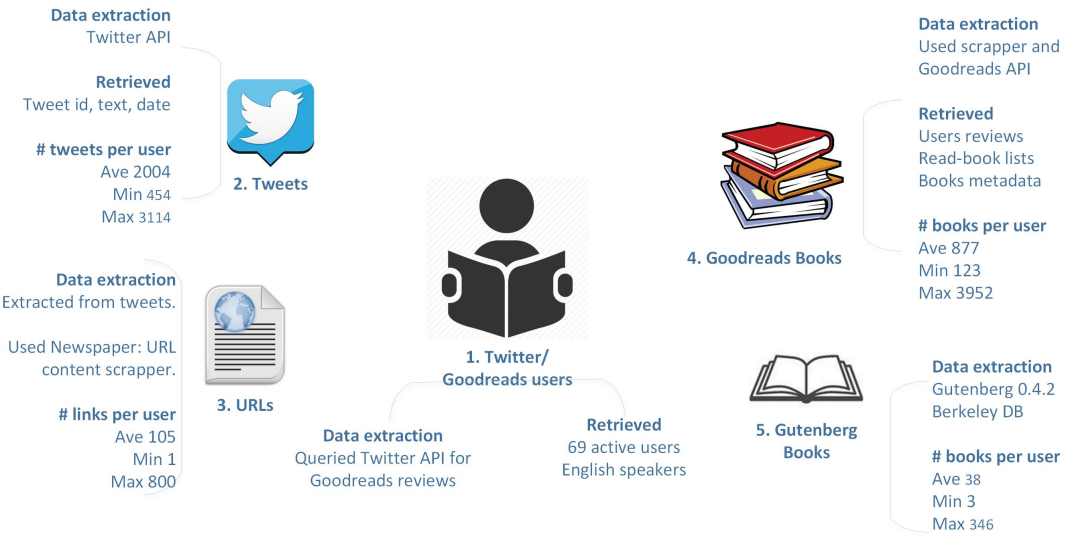

Figure 1: Dataset collection and statistics.

4.2 Data preprocessing

Before topic extraction, text of tweets, link, and book descriptions must be cleaned. The tweets were tokenized using Tweet Tokenizer from NLTK [5], which is Twitter-conscious, and tagged using the GATE Twitter part of speech tagger [12]. Hashtags were checked using aspell.[4] Misspelled words were excluded because they are not useful: the goal is to match them with book descriptions which are spelling-error-free. For links and book descriptions, regular NLTK Word tokenizer [5] and Stanford part-of-speech tagger [42] were applied. Only nouns (singular or plural) were kept, then lemmatized by NLTK WordNet Lemmatizer [5]. 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. In fact, to model user interests based on their social media accounts, other researchers also considered only nouns [10][43].

After building topic models from tweets and links, we noticed that unimportant (generic) terms such as "website" are dominant. Therefore, we went further by excluding NLTK stop words, 100 most common English nouns,[5] and words of fewer than 4 letters. For tweets, we also filtered out the 200 words with lowest idf weights.[6] Repeated content of links is deleted, and so are Web-related terms, *e.g.,* "website" and "Facebook".

4.3 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. 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. The lowest number of ratings needed to develop CB with quality recommendations is 10. This threshold is adopted by many researchers, including [46].

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 [6]. After conducting many trials, the number of NMF topics was set to five, with six words in each, because topics became redundant afterward. The number of tf-idf topics was set to 100. 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 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. The on-line appraisal would provide more performance insights, but it is an expensive option that requires the deployment of a real-time version of TMB. A user study is another option; it was avoided because it usually includes a limited number of users.

5.1 Experiment settings

Strategies of top k recommendations are tested in a similar fashion to the leave-one-out evaluation applied in [13, 21, 23]; it splits the dataset into a training set and a one-item test set, then generates a list of the top N recommendations from the training set. In our setting, however, the training set is made up of all books not included in the recommendation timeframe, so we cannot use it for prediction. This is why we followed a slightly different assessment methodology, used in ranking-based RSs adopted by [24] and [25].

We created one set of 1000 random books that are unique and not rated by any user. For each user, we randomly selected one

[4] http://aspell.net/
[5] http://www.linguasorb.com/english/most-common-nouns/
[6] The idf weights for tweets from all users after the deletion of stop words.

relevant book from the recommendation timeframe, added it to the 1000 books and asked our system 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. Similarly to many related projects, we set k to 10. Metrics adopted are hit-rate (Equation 4), sometimes called recall, and the average reciprocal hit-rank (Equation 5) [13]. To avoid a bias, five trials were conducted, and the reported results averaged. We measured the statistical difference in results using the t-test at a maximum of p-value = 0.05.

$$HR = \frac{\#hits}{\#users} \tag{4}$$

$$ARHR = \frac{1}{\#users} \sum_{i=1}^{\#hits} \frac{1}{f_i} \tag{5}$$

Our approach was compared to a content-based RS and to a random system. 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 4.1). Although we considered a comparison with collaborative filtering (CF), the rating matrix is highly sparse, which means that the results would not reflect a typical CF.

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.

5.1.1 Results. Figure 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 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 different except for the tf-idf-emb of tweets and the tf-idf-emb of tweets and links. There is no consistency in 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.

6 DISCUSSION

The field of RSs is active. Many state-of-the-art recommendation methods have been proposed in the recent years. However, we only compared TMB results with CB which has been around for a long while. TMB gives similar performance to a traditional system, CB without the need for user rating history. Nevertheless, we do not claim that the system works independently. To verify this, more comprehensive experiments are required.

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 eliminate noise, 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 term that is not as popular, like "mythology", may have a high idf value and be in the top tf-idf list.

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 3) the word embeddings of one user profile (b) and his two book profiles (c, d). Section (a) of Figure 3 shows the closeness of word vectors found the UP and BPs. One can notice the variety of topics in the user profile. 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 capturing a wider range of interests. In fact, there is an evident difference in their effect on pure NMF and tf-idf. 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.

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

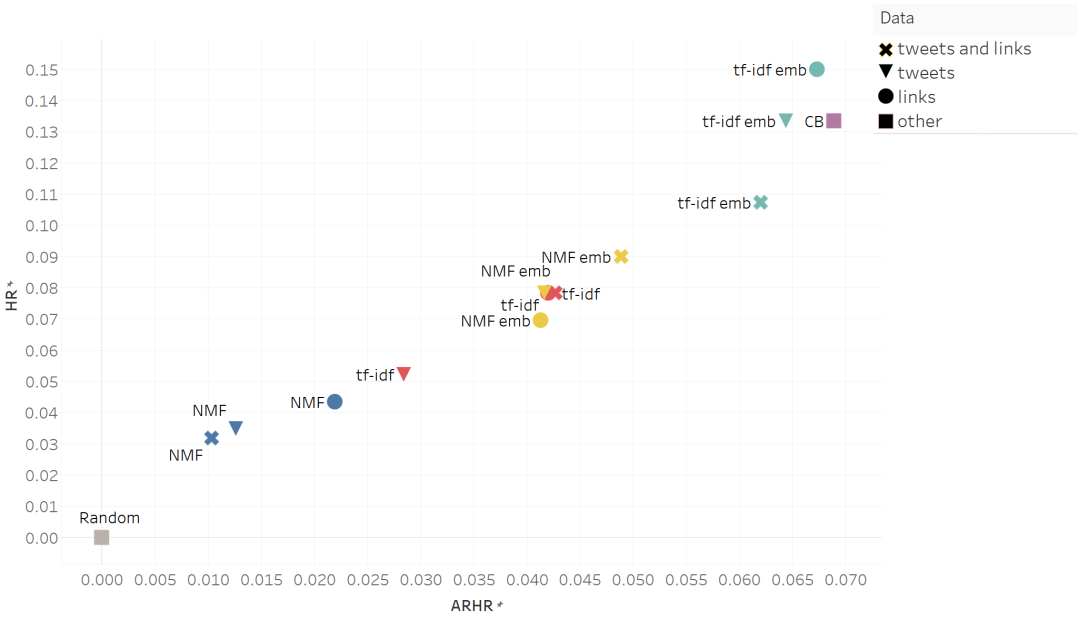

Figure 2: The comparison of TMB approaches, CB and the random system.

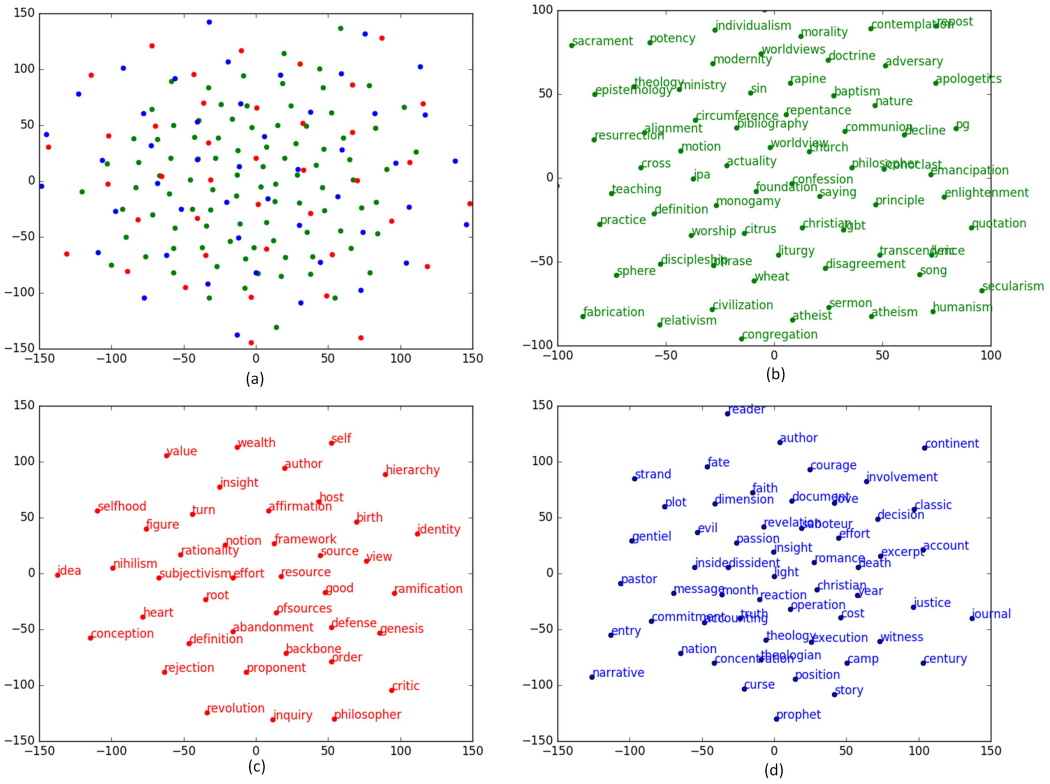

Figure 3: Word embeddings of terms in a user profile and two book descriptions.

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 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*.

7 CONCLUSION

This paper proposes TMB, a system that builds a topic model for a user from textual content shared voluntarily on her social media, and recommends the books most related to these topics. We acquired a user's distinctive topics by tf-idf weighting and represent them as word embeddings in order to capture their context. TMB achieves a recommendation accuracy similar to CB, a commonly used book RS, particularly when word embeddings are deployed. Thus, TMB can aid current RSs in suggesting books to new users without major loss in performance.

For future improvement, we plan to study the temporal effect on topic models, as well as the relationship between the level of user activity and the accuracy of the recommendations. 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, especially verbs and adjectives. To enhance the performance of the system, we will train embeddings for tweets and books.

ACKNOWLEDGMENT

Support for this work has come from the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

[1] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. 2011. Analyzing User Modeling on Twitter for Personalized News Recommendations. In *Proc. 19th International Conference on User Modeling, Adaption, and Personalization (UMAP'11)*. Springer-Verlag, Berlin, Heidelberg, 1–12. http://dl.acm.org/citation.cfm?id=2021855.2021857

[2] Rel Guzman Apaza, Elizabeth Vera Cervantes, Laura Cruz Quispe, and Jose Ochoa Luna. 2014. Online Courses Recommendation based on LDA. (2014).

[3] David Ben-Shimon, Alexander Tsikinovsky, Lior Rokach, Amnon Meisles, Guy Shani, and Lihi Naamani. 2007. Recommender System from Personal Social Networks. In *Advances in Intelligent Web Mastering: Proc. 5th Atlantic Web Intelligence Conference – AWIC'2007, Fontainbleau, France, June 25 – 27, 2007*, Katarzyna M. Wegrzyn-Wolska and Piotr S. Szczepaniak (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 47–55. DOI: https://doi.org/10.1007/978-3-540-72575-6_8

[4] Sonia Bergamaschi and Laura Po. 2015. Comparing LDA and LSA Topic Models for Content-Based Movie Recommendation Systems. In *Web Information Systems and Technologies: 10th International Conference, WEBIST 2014, Barcelona, Spain, April 3-5, 2014, Revised Selected Papers*, Valérie Monfort and Karl-Heinz Krempels (Eds.). Springer International Publishing, Cham, 247–263. DOI: https://doi.org/10.1007/978-3-319-27030-2_16

[5] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python* (1st ed.). O'Reilly Media, Inc.

[6] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 108–122.

[7] Eduardo Castillejo, Aitor Almeida, and Diego López-De-Ipiña. 2012. Alleviating cold-user start problem with users' social network data in recommendation systems. In *Workshop on Problems and Applications in AI, ECAI-12*.

[8] Jilin Chen, Rowan Nairn, Les Nelson, Michael Bernstein, and Ed Chi. 2010. Short and Tweet: Experiments on Recommending Content from Information Streams. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, 1185–1194. DOI: https://doi.org/10.1145/1753326.1753503

[9] Xueqi Cheng, Jiafeng Guo, Shenghua Liu, Yanfeng Wang, and Xiaohui Yan. 2013. Learning Topics in Short Texts by Non-negative Matrix Factorization on Term Correlation Matrix.. In *Proc. SIAM International Conference on Data Mining*. SIAM, 749–757. http://dblp.uni-trier.de/db/conf/sdm/sdm2013.html#ChengGLWY13

[10] D. Choi, J. Kim, E. Lee, C. Choi, J. Hong, and P. Kim. 2014. 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*. 36–40. DOI: https://doi.org/10.1109/IMIS.2014.94

[11] Gianmarco De Francisci Morales, Aristides Gionis, and Claudio Lucchese. 2012. 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)*. ACM, 153–162. DOI: https://doi.org/10.1145/2124295.2124315

[12] Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. 2013. 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.

[13] Mukund Deshpande and George Karypis. 2004. Item-based top-N Recommendation Algorithms. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 143–177. DOI: https://doi.org/10.1145/963770.963776

[14] Sharon Givon and Victor Lavrenko. 2009. Predicting Social-tags for Cold Start Book Recommendations. In *Proc. Third ACM Conference on Recommender Systems (RecSys '09)*. ACM, 333–336. DOI: https://doi.org/10.1145/1639714.1639781

[15] Daniel Godfrey, Caley Johns, Carl Dean Meyer, Shaina Race, and Carol Sadek. 2014. A Case Study in Text Mining: Interpreting Twitter Data From World Cup Tweets. *CoRR* abs/1408.5427 (2014). http://arxiv.org/abs/1408.5427

[16] Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. 2013. Using Topic Models for Twitter Hashtag Recommendation. In *Proc. 22Nd International Conference on World Wide Web (WWW '13 Companion)*. ACM, 593–596. DOI: https://doi.org/10.1145/2487788.2488002

[17] Ido Guy, Naama Zwerdling, Inbal Ronen, David Carmel, and Erel Uziel. 2010. Social Media Recommendation Based on People and Tags. In *Proc. 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10)*. ACM, 194–201. DOI: https://doi.org/10.1145/1835449.1835484

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

[19] Diane J. Hu, Rob Hall, and Josh Attenberg. 2014. Style in the Long Tail: Discovering Unique Interests with Latent Variable Models in Large Scale Social E-commerce. In *Proc. 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*. ACM, 1640–1649. DOI: https://doi.org/10.1145/2623330.2623338

[20] Nirmal Jonnalagedda, Susan Gauch, Kevin Labille, and Sultan Alfarhood. 2016. Incorporating popularity in a personalized news recommender system. *PeerJ Computer Science* 2 (2016), e63.

[21] Zhao Kang, Chong Peng, and Qiang Cheng. 2016. Top-N Recommender System via Matrix Completion. In *Proc. Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*. AAAI Press, 179–184. http://dl.acm.org/citation.cfm?id=3015812.3015839

[22] Hikmet Kapusuzoglu and Sule Gunduz Öguducu. 2011. A Relational Recommender System Based on Domain Ontology. In *Emerging Intelligent Data and Web Technologies (EIDWT), 2011 International Conference on*. 36–41. DOI: https://doi.org/10.1109/EIDWT.2011.15

[23] George Karypis. 2001. Evaluation of Item-Based Top-N Recommendation Algorithms. In *Proc. Tenth International Conference on Information and Knowledge Management (CIKM '01)*. ACM, 247–254. DOI: https://doi.org/10.1145/502585.502627

[24] Yehuda Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *Proc. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*. ACM, 426–434. DOI: https://doi.org/10.1145/1401890.1401944

[25] Qiuxia Lu, Tianqi Chen, Weinan Zhang, Diyi Yang, and Yong Yu. 2012. 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)*. IEEE Computer Society, Washington, DC, USA, 258–265. http://dl.acm.org/citation.cfm?id=2457524.2457692

[26] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

[27] Raymond A. Mar and Keith Oatley. 2008. The Function of Fiction is the Abstraction and Simulation of Social Experience. *Perspectives on Psychological Science* 3, 3 (01 May 2008), 173–192. DOI: https://doi.org/10.1111/j.1745-6924.2008.00073.x

[28] Raymond A. Mar, Keith Oatley, and Jordan B. Peterson. 2009. Exploring the link between reading fiction and empathy: Ruling out individual differences and examining outcomes. *Communications* 34, 4 (1 Dec. 2009), 407–428. DOI: https://doi.org/10.1515/comm.2009.025

[29] Daniel Mican, Loredana Mocean, and Nicolae Tomai. 2012. Building a Social Recommender System by Harvesting Social Relationships and Trust Scores between Users. In *Business Information Systems Workshops: BIS 2012 International Workshops and Future Internet Symposium, Vilnius, Lithuania, May 21-23, 2012 Revised Papers*, Witold Abramowicz, John Domingue, and Krzysztof Węcel (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 1–12. DOI: https://doi.org/10.1007/978-3-642-34228-8_1

[30] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). http://arxiv.org/abs/1301.3781

[31] Raymond J. Mooney and Loriene Roy. 2000. Content-based Book Recommending Using Learning for Text Categorization. In *Proc. Fifth ACM Conference on Digital Libraries (DL '00)*. ACM, 195–204. DOI: https://doi.org/10.1145/336597.336662

[32] P. Nair, M. Moh, and T. S. Moh. 2016. Using Social Media Presence for Alleviating Cold Start Problems in Privacy Protection. In *2016 International Conference on Collaboration Technologies and Systems (CTS)*. 11–17. DOI: https://doi.org/10.1109/CTS.2016.0022

[33] Sergey Nikolenko. 2015. SVD-LDA: Topic Modeling for Full-Text Recommender Systems. In *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*, Obdulia Pichardo Lagunas, Oscar Herrera Alcántara, and Gustavo Arroyo Figueroa (Eds.). Springer International Publishing, Cham, 67–79. DOI: https://doi.org/10.1007/978-3-319-27101-9_5

[34] Ante Odić, Marko Tkalčič, Andrej Košir, and Jurij F. Tasič. 2011. 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.*

[35] Dharmendra Pathak, Sandeep Matharia, and C. N. S. Murthy. 2013. NOVA: Hybrid book recommendation engine. In *Advance Computing Conference (IACC), 2013 IEEE 3rd International*. 977–982. DOI: https://doi.org/10.1109/IAdCC.2013.6514359

[36] Marco Pennacchiotti and Siva Gurumurthy. 2011. Investigating Topic Models for Social Media User Recommendation. In *Proc. 20th International Conference Companion on World Wide Web (WWW '11)*. ACM, 101–102. DOI: https://doi.org/10.1145/1963192.1963244

[37] Maria Soledad Pera, Nicole Condie, and Yiu-Kai Ng. 2011. Personalized Book Recommendations Created by Using Social Media Data. In *Proc. 2010 International Conference on Web Information Systems Engineering (WISS'10)*. Springer-Verlag, Berlin, Heidelberg, 390–403. http://dl.acm.org/citation.cfm?id=2044492.2044531

[38] Maria Soledad Pera and Yiu-Kai Ng. 2011. With a Little Help from My Friends: Generating Personalized Book Recommendations Using Data Extracted from a Social Website. In *Proc. 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01 (WI-IAT '11)*. IEEE Computer Society, Washington, DC, USA, 96–99. DOI: https://doi.org/10.1109/WI-IAT.2011.9

[39] Ankan Saha and Vikas Sindhwani. 2012. Learning Evolving and Emerging Topics in Social Media: A Dynamic Nmf Approach with Temporal Regularization. In *Proc. Fifth ACM International Conference on Web Search and Data Mining (WSDM '12)*. ACM, 693–702. DOI: https://doi.org/10.1145/2124295.2124376

[40] Suvash Sedhain, Scott Sanner, Darius Braziunas, Lexing Xie, and Jordan Christensen. 2014. Social Collaborative Filtering for Cold-start Recommendations. In *Proc. 8th ACM Conference on Recommender Systems (RecSys '14)*. ACM, 345–348. DOI: https://doi.org/10.1145/2645710.2645772

[41] Marko Tkalčič, Andrej Košir, and Jurij Tasič. 2013. The LDOS-PerAff-1 corpus of facial-expression video clips with affective, personality and user-interaction metadata. *Journal on Multimodal User Interfaces* 7, 1 (2013), 143–155. DOI: https://doi.org/10.1007/s12193-012-0107-7

[42] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. 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)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 173–180. DOI: https://doi.org/10.3115/1073445.1073478

[43] Keita Tsuji, Nobuya Takizawa, Sho Sato, Ui Ikeuchi, Atsushi Ikeuchi, Fuyuki Yoshikane, and Hiroshi Itsumura. 2014. Book Recommendation Based on Library Loan Records and Bibliographic Information. *Procedia - Social and Behavioral Sciences* (2014), 478–486. DOI: https://doi.org/10.1016/j.sbspro.2014.07.142 3rd International Conference on Integrated Information (IC-ININFO).

[44] Paula Cristina Vaz, Ricardo Ribeiro, and David Martins de Matos. 2013. Book Recommender Prototype Based on Author's Writing Style. In *Proc. 10th Conference on Open Research Areas in Information Retrieval (OAIR '13)*. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, Paris, France, France, 227–228. http://dl.acm.org/citation.cfm?id=2491748.2491800

[45] Chong Wang and David M. Blei. 2011. Collaborative Topic Modeling for Recommending Scientific Articles. In *Proc. 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. ACM, 448–456. DOI: https://doi.org/10.1145/2020408.2020480

[46] Yiwen Wang, Natalia Stash, Lora Aroyo, Laura Hollink, and Guus Schreiber. 2009. 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)*. CEUR-WS.org, Aachen, Germany, Germany, 16–28. http://dl.acm.org/citation.cfm?id=2889761.2889763

[47] Xiaohui Yan, Jiafeng Guo, Shenghua Liu, Xue-qi Cheng, and Yanfeng Wang. 2012. Clustering Short Text Using Ncut-weighted Non-negative Matrix Factorization. In *Proc. 21st ACM International Conference on Information and Knowledge Management (CIKM '12)*. ACM, 2259–2262. DOI: https://doi.org/10.1145/2396761.2398615

[48] Xuejun Yang, Hongchun Zeng, and Weihong Huang. 2009. ARTMAP-Based Data Mining Approach and Its Application to Library Book Recommendation. In *Intelligent Ubiquitous Computing and Education, 2009 International Symposium on*. 26–29. DOI: https://doi.org/10.1109/IUCE.2009.43