Authorship Identification for Literary Book Recommendations

Haifa Alharthi, Diana Inkpen, Stan Szpakowicz
University of Ottawa, Ottawa, Ontario, Canada
{halha060,Diana.Inkpen,szpak@eecs}.uottawa.ca

Abstract

Book recommender systems can help promote the practice of reading for pleasure, which has been declining in recent years. One factor that influences reading preferences is writing style. We propose a system that recommends books after learning their authors' style. To our knowledge, this is the first work that applies the information learned by an author-identification model to book recommendations. We evaluated the system according to a top-k recommendation scenario. Our system gives better accuracy when compared with many state-of-the-art methods. We also conducted a qualitative analysis by checking if similar books/authors were annotated similarly by experts.

1 Introduction

Recommender systems (RSs) are useful for internet users who may find it hard to choose from the multitude of available products and services. RSs predict how likely the target user is to be interested in an item which might have been unknown to her. In this work, we consider book recommender systems, which could be useful in libraries, schools, and on e-learning portals. Nowadays, with the introduction of e-books, readers can access inexpensive resources with little effort. It was expected that the act of reading for pleasure would become widespread, but statistics demonstrate the opposite; it is declining, particularly among young people.[1]

A number of studies have shown the benefits of reading. In a comparison between the well-being of 7,500 Canadian adult readers and non-readers, the former have been found statistically significantly more likely to report better health or mental health, to volunteer, and to feel strongly satisfied with life (Hill, 2013). Fiction has been shown to stimulate profound social communication (Mar and Oatley, 2008). Exposure to fiction correlates with a greater ability for empathy and social support (Mar et al., 2009). It also has a lingering biological effect, notably in the connectivity of the brain (Berns et al., 2013). It is therefore a worthwhile endeavor to deploy artificial intelligence techniques to spark people's interest in books by recommending the right type of books.

There are two main types of traditional recommender systems: collaborative filtering (CF) and content-based recommendation (CB). When a CF system predicts if a user likes a book, it relies on the ratings of all active users in the system. CF depends on a rating matrix which could be large and sparse, and that might lead to inferior recommendations. Sparsity occurs when items/users do not have enough ratings. It can be noticed in library records that many books are never checked out (as is, *e.g.*, the case of 75% of the books in the library of Changsha University of Science and Technology (Yang et al., 2009)). A few other problems face CF systems: *new items* which are difficult to recommend until enough ratings have been received, *the long tail* when most items are overlooked because of few very popular ones, *shilling attack* when items are rated as a way of promotion, and the *gray sheep problem* of users who have tastes entirely different from other users (Alharthi et al., 2017).

Content-based recommendation, on the other hand, relies on the content of items (*e.g.*, a book's genre and author). It is a classifier that finds patterns in the target user's past reading preferences, and predicts

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

Proceedings of the 27th International Conference on Computational Linguistics, pages 390–400
Santa Fe, New Mexico, USA, August 20-26, 2018.

future interests. CB does not require a rating matrix; that eliminates all the aforementioned issues with CF. A newly added book is recommended if it is similar to the target user's preferences. CB also makes recommendations that are interpretable (*e.g.*, a book is recommended because the user tends to like a specific genre). However, it is difficult for CB to recommend items that have inadequate content. Another weakness, *overspecialization*, occurs when the system makes non-diverse recommendations (Alharthi et al., 2017). Hybrid RSs can combine CF and CB to tackle such issues and enhance the recommendation accuracy.

A content-based RS is proposed that analyzes the texts of books to learn users' reading interests. A survey on book recommender systems (Alharthi et al., 2017) describes only a few RSs that take the actual text of books into account. There is a lot of work that explores literary works, including automatic genre identification (Ardanuy and Sporleder, 2016) and learning the narrative structure (Elson et al., 2010). It is quite surprising, then, that the analysis of the textual content of books to improve their recommendations is still quite limited. A user's reading preferences might be influenced by many elements specific to books. For example, recommendations given by the readers' advisory (a library service that suggests books to patrons) rely on multiple appeal factors. The factors are *characterization, frame, language and writing style, pacing, special topics, storyline,* and *tone*. To obtain information about books' appeal factors, librarians can subscribe (at a fee) to reader-advisory databases such as NoveList, established by professionals (Pera and Ng, 2014a).

Commercial applications that perform natural language processing (NLP) on the text of books have begun to appear; one example is BookLamp which was acquired by Apple in 2014.[2] Also, the well-known Google Books provide a retrieval system that searches inside the full text of books for terms that appear in a given query. Moreover, e-book recommendations are already provided via WIFI-connected Kindle, Kobo and other e-readers. Our proposed system could therefore be deployed by e-book providers, and could help current online stores to boost their deployed collaborative filtering systems.

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. More advanced document retrieval methods such as latent Dirichlet allocation (LDA) (Blei et al., 2003) and latent semantic indexing (LSI) (Deerwester et al., 1990) also take advantage of term frequencies. Recently, doc2vec models (Le and Mikolov, 2014) have achieved state-of-the-art results in document classification and recommendations (Gupta and Varma, 2017; Wang et al., 2016).

In this paper, we focus on learning authors' writing styles in aid of book recommendation. 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. 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 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 only for tasks that are mutually semantically similar.

The system has two components: authorship identification and book recommendation. For the former, we adopt two approaches similar to (Solorio et al., 2017); they use convolutional neural networks (CNN) over a sequence of words and sequence of character bigrams. 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 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

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

features associated with the target user ratings to generate a list of ranked books.

This paper makes several contributions. To our knowledge, this is the first work to transfer the information learned from an author identification model to book recommendations. We evaluated the system according to a top-k recommendation scenario. Our system gives better accuracy when compared against many state-of-the-art methods, namely the vector space model (VSM) (Salton and McGill, 1986), latent Dirichlet allocation (LDA) (Blei et al., 2003), latent semantic indexing (LSI) (Deerwester et al., 1990) and paragraph2vec (Le and Mikolov, 2014). Moreover, qualitative analysis was conducted on the vectors generated by the author identification model. The books with similar vectors have been found to have similar description in NoveList.

The remainder of this paper is organized as follows. Section 2 surveys related work in recommender systems and author identification. Section 3 explains the proposed system in detail. Section 4 describes the evaluation process. Section 5 illustrates the results and analyzes them. Section 6 concludes, and highlights future work.

2 Related work

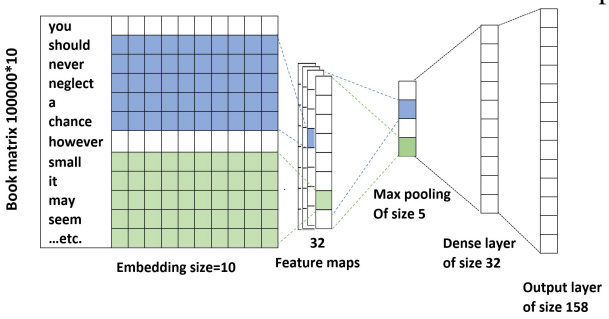
Only a few of more than thirty book RSs surveyed in (Alharthi et al., 2017) take the actual text of books into account. In addition to library circulation analysis, many researchers took advantage of the textual reviews shared by individuals and communities. Another category of book RSs considers stylometry features; they are learned from an author's writing, and can help in authorship identification. Stamatatos (2009) categorizes stylometry features into five classes: lexical, syntactic, semantic, application-specific and character-based. In the stylometric content-based (CB) book RS proposed by (Vaz et al., 2012a), a few features are taken into account. Books were represented in two ways: (1) as vectors of LDA topics, (2) as vectors of stylometric features (document length, vocabulary richness, part-of-speech bigrams and most frequent words). Experiments conducted using the Rocchio algorithm (Manning et al., 2008) on LitRec dataset (Vaz et al., 2012c) showed that the combination of stylometric CB with collaborative filtering (CF) provides better accuracy than an individual CF or CB system. Compared to other representations, the LDA-based vectors showed the best results. Unlike the method we propose, (Vaz et al., 2012a) does not associate stylometric features found in books with authors.

The author's writing style was also considered in (Pera and Ng, 2014a; Pera and Ng, 2014b; Pera and Ng, 2015), yet it is learned from the online reviewers' point of view and not automatically from the textual content of books. Appealing terms (from a fixed number of terms found in (Pera and Ng, 2014a)) are extracted from readers' reviews automatically collected from websites such as Amazon.com. The resemblance between vectors of appealing terms of the preferred and candidate books is calculated, and a recommendation is made accordingly.

Upon receiving the name of the user's favourite author, (Zhang and Chow, 2015) exploit the whole text of an e-book to suggest authors and e-books. To overcome the spatial distribution issue—when words are treated without taking into consideration their order—every author is represented in a hierarchical structure which consists of four layers. The first layer is reserved for the author's related information (*e.g.*, education and political views), while the following three levels are dedicated to the author's books, pages of every book, and paragraphs on each page. The author representations are developed using a multilayer self-organizing map (MLSOM) (Rahman et al., 2007). To recommend books, the system considers the last three layers. On the other hand, to suggest authors, the RS exploits all the layers. The evaluation is conducted by computing the relevance of two books. If a queried book has the same genres as the recommended book, this is considered a relevant recommendation. The results show higher performance when compared with other CB systems, including those using VSM and LSI.

The baseline systems are commonly used in recommender systems. The vector space model, which is the standard approach in information retrieval, was applied to descriptions of books in (Tsuji et al., 2014; Pera and Ng, 2014b), but not to the whole text of books. Both latent semantic indexing and latent Dirichlet allocation are topic-modelling techniques widely used in information retrieval and recommender systems. To name a few, they were used to recommend movies (Bergamaschi and Po, 2015) and articles (Lin, 2017; Nagori and Aghila, 2011). Recently, paragraph2vec—or, as it is frequently

Figure 1: The use of a convolutional neural network for authorship identification



named, Doc2vec—was proposed in (Le and Mikolov, 2014) to offer a fixed-size representation for texts (paragraphs or documents). An unsupervised algorithm predicts the words in a document and generates a vector for every document. Doc2vec was applied in RSs to suggest scientific articles in (Gupta and Varma, 2017) and answers in question-answering systems in (Wang et al., 2016).

A simple system using CNN is proposed in (Solorio et al., 2017). It takes a sequence of characters of a tweet as input and predicts its author, a Twitter user. The system performance with different inputs (including character bigrams, character unigrams and words) is evaluated. The character bigram input gave the best accuracy. Although it has been proposed for short texts, we noticed that it gives acceptable accuracy when predicting authors of books; that is why we have adopted it. As explained in section 3.1, we modified the network in (Solorio et al., 2017) by adding a dense layer which helps extract book AuthId features.

Recent research also shows how the use of recurrent neural networks (RNN) 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 (Lahiri, 2014). Their model represents a sequence of words (initialized as GloVe pretrained embeddings) in a sequence followed by average pooling, and then another GRU that represents the sequence of sentences in an article. A simple recursive neural network adopted in (Macke and Hirshman, 2015) showed high accuracy only when the number of classes was small: 10 authors. Our preliminary experiments show that RNN-based author identification models have poor accuracy, so we do not use such models in this work.

3 Methodology

This section describes the two components of the system. It first illustrates and explains the author identification system, which was proposed by (Solorio et al., 2017), and our modifications to the system. Next, the recommendation procedure is described.

3.1 Author Identification

As shown in Figure 1, drawn similarly to (Kim, 2014), the neural network has the following layers.
Embedding layer (also called *lookup table*). It takes a sequence of words or character bigrams as input, and maps each word or bigram to an embedding of size *k*. The embedding of the *i*th word/character is a dense *k*-dimensional vector $x_i \in R^k$. Each book is represented as a matrix, with one word/character embedding per row. A book has sequence length *n* (*n* is the number of words/characters) where a sequence $x_{1:n} = x_1 \oplus x_2 ... \oplus x_n$ is a concatenation of all words from x_1 to x_n.
Convolution layer. This layer consists of one or more filters (also called kernels) that are applied to windows of words 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 (non-linear), and *b* is a bias. A feature map $\mathbf{c} = [c_1, c_2, ...c_{n-h+1}]$ is created by a filter *w* that slides over all the possible windows of words in a book.
Pooling layer. This layer decreases the dimensionality of the feature map. One approach is max-over-time pooling. Given a feature map **c**, it returns $max\{\boldsymbol{c}\}$; as a result, only the features with the highest importance (maximum value) are kept (Kim, 2014).

Fully-connected layer (also called *dense layer*). Each neuron in this layer is connected to each neuron in the previous layer. This layer is essential in our framework, because it generates the book representation. The pooling layer's output is flattened first, then fed to the dense layer which produces a fixed-size vector with dimensions equal to the number of neurons in the layer.
Output layer. It is a fully-connected softmax layer which outputs vector with dimensions equal to the number of authors (labels). Each dimension represents the probability that the input book belongs to a specific author. The values of all elements of one vector sum to 1.

After the model has been trained and achieved accurate predictions, it is used to extract the features of an intermediate layer. Given the text of a book as an input, the fixed-size vector generated by the fourth layer is extracted. This vector is considered as the AuthId book representation which is used in the next step to make recommendations.

3.2 Book Recommendations

Given a user's reading history associated with AuthId book representations, a regressor predicts her future ratings. Book recommendations are ranked according to the predicted rating values. We applied Support Vector Regression (SVR) 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 Radial Basis Function (RBF)—see Equation 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 2 with α_i^*, where α_i are Lagrange multipliers and N is the number of data points (Gunn, 1997; Basak et al., 2007).

$$k(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{1}$$

$$f(x) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)k(x_i, x) + b \tag{2}$$

4 Evaluation

4.1 Dataset and Preprocessing

We use the Litrec dataset (Vaz et al., 2012c) which contains data on 1,927 users who rated 3,710 literary works. The dataset incorporates data from Goodreads and from Project Gutenberg. It also has the complete texts of books, labelled 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. 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). Also, in order to train the author identification network, we need multiple books per author. Thus, we only kept authors with a minimum of three books. Data for users with fewer than 10 ratings are also deleted. The lowest number of ratings needed to develop CB with quality recommendations is 10, 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 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.

4.2 The Experimental Setting

A recent survey (Zhang et al., 2017) states that the percentage of 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. Books rated by a target user are divided into training and test set. The system learns from

the books in the training set and ranks the remaining books in the dataset (we call it the ranked list). The books in the test set 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, we set k to 10. Three-fold cross-validation is adopted per user, and the results are averaged. We measure the statistical difference in results, using the t-test at a p-value of 0.05 or 0.01.

Metrics. We compute the recommendation accuracy by precision at k (P@k) and recall at k (R@k)—Equations 3-4—where a *relevant* book means to a preferred book, and *recommended* means ranked in the top k list.

$$P@k = \frac{\#\ relevant\ books\ in\ top\ k\ recommended}{k} \tag{3}$$

$$R@k = \frac{\#\ relevant\ books\ in\ top\ k\ recommended}{\#\ of\ relevant\ books} \tag{4}$$

Baselines. To implement LDA, LSI, VSM and Doc2vec, we used gensim,[3] a Python library. The texts of books were tokenized and down-cased, and NLTK stopwords and least frequent words were filtered out. In the first three systems, relevant books in the user training set are compiled and considered as one query. Using cosine similarity, the top-k books most similar to the query are recommended.

For LDA and LSI, we experimented with 10, 50, 100 and 200 topics; the best accuracy is reported. The Doc2vec model was trained on book texts; a book id is considered as the label. For training the model, we tried dimensions of 100 versus 300 and window size of 10 versus 5, and we report the best results. The average of document vectors of relevant books in the user training set is considered as a query. The books with highest cosine similarity to this vector are recommended. We also compare with a plain author-based RS which uses SVR similarly to our proposed system, with one difference: instead of book AuthId representations, author ids are used.

Parameter Settings. One question usually raised with transfer learning is this: when does one stop training the source model? 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 becomes close to 100%. The network is trained using the RMSprop optimizer over 32 batches. For the non-linearity, rectified linear units are adopted. The number of neurons in the first fully connected layer is decided empirically to be 32. We also experimented with various combinations of parameters: embedding size = 1, 5, 10, number of filters = 16, 32, kernel size = 2, 5, and maximal pooling size = 2, 5.

In the char-bigram CNN, the combination that provides the best validation accuracy (64%) is reached at the 7^{th} epoch with embedding size of 10, 16 filters and kernel size and max-pooling sizes of 2. The sequence character bigrams can be very lengthy (this causes the system to crash); therefore, a maximum of 150,000 tokens is imposed. For the word CNN, a validation accuracy (65%) was achieved at the 17^{th} epoch when using embedding size of 10 with 32 filters and 5 for kernel size and max-pooling size. These experiments were implemented using keras,[4] a Python library, on a NVIDIA GeForce Titan X Pascal GPU with memory of 12,184 MiB.

The SVR was developed using scikit-learn.[5] The Radial Basis Function (RBF) is used with gamma=0.001. 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). The final number of irrelevant books in the training set is the same as relevant ones.[6]

5 Results and Analysis

Figure 2 illustrates how the proposed system, whether based on words (AuthId_words) or character bigrams (AuthId_char), retrieves relevant books more than the baselines, with the former achieving statis-

[3]https://radimrehurek.com/gensim/

[4]https://keras.io/

[5]http://scikit-learn.org/stable/

[6]To download the code and dataset, visit `https://tinyurl.com/ybsdxg5a`

Figure 2: Precision@10 and recall@10 generated by our system (AuthId_words and AuthId_char), and the baselines.

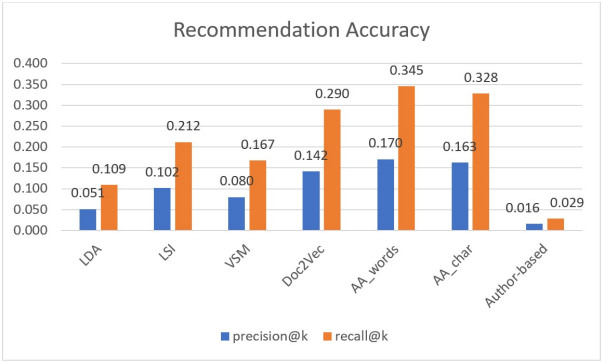
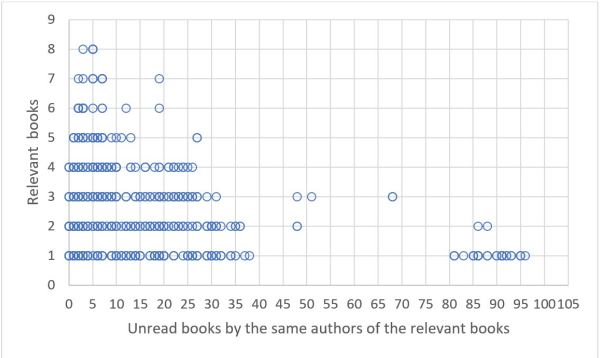


Figure 3: Users' relevant recommended books versus unread books by the same authors



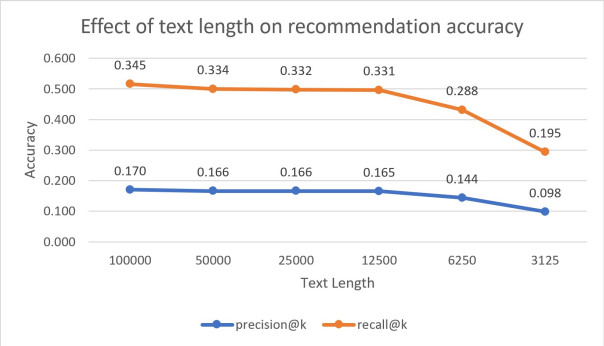
tically higher accuracy at a p-value of 0.01. In fact, AuthId_words score better precision and recall at a p-value of 0.05 when compared with AuthId_char. One possible reason is that in the latter we capped sequence length at 150,000. We also believe that the use of words could capture topical similarity, as highlighted by the qualitative analysis (Table 1).

Many users have fewer than 10 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 10 books in their test set, making the R@10 very low even for an ideal system. The best baseline is doc2vec, followed by LSI. LDA's low accuracy is surprising, yet it is possible that more preprocessing is required for LDA to work properly. We expected the author-based system to have high recall@k 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 read by the user).

This observation has led us to ask how many unread books there are for the authors of books retrieved by our system. 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 where one circle refers to one test case (one fold for one user). The y-axis represents the number of relevant books in the top 10 recommended list. The x-axis refers to the number of unread books by the same authors who wrote the relevant books. For example, the mark at (3, 68) means that the AuthId_words system could recommend three books relevant to a target user from a list of items containing 68 books written by the same authors as the three retrieved books. In 302 cases, the system could retrieve one or more relevant books from a list with more than ten irrelevant books by the same authors. In 12 cases, the number of unread books exceeded 80. The system recommended at most eight relevant books; that was achieved when three and five irrelevant books were on the list of books to rank.

To assess the effect of text length on the accuracy of recommendations, we developed book AuthId

Figure 4: Effect of text length on recommendation accuracy



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 embedding size, number of filters, kernel size and pooling size as in section 4.2. In the author identification model, the validation accuracy of the 100,000 words is similar to 3125 words. However, figure 4 shows that *the shorter the text length, the less accurate the recommendations*. Yet, a statistical difference only occurs when a great deal of the text is removed (*i.e.*, 6250 and 3125 words).

We went further to analyze the quality of AuthId book representations by measuring if similar representations have overlapping descriptions in NoveList. Using cosine similarity, we studied the 10 books most similar to "The Snow-Image: A Childish Miracle" by Nathaniel Hawthorne. We selected this book because NoveList has information on all its related authors. In Tables 1-2, which represent book AuthId using CNN_words and CNN_char_bigrams respectively, one can see Gutenberg book IDs in descending order according to their similarity values, as well as author information.

In both tables, Hawthorne himself authored the first four books. The following list of books by the two methods, however, are entirely different from each other. Table 1 has books by only three authors with many common points with the description of Hawthorne's profile (in Bold). The authors of the first four books and the last five books write mostly about related topics (subject headings). On the other hand, Table 2 contains information on seven unique authors, one sharing the same genre and two having the opposite storyline and pace. Yet, most of them write about different topics than Hawthorne. It is noticeable that the use of words rather than characters captures similarity at the topic level. Solorio et al. (2017) further assess the AuthId model.

6 Conclusion

In the work discussed in this paper, we represent books as vectors learned in relation to authors. Such representations are not only useful in content-based RSs as the results have shown, but can be used to enrich current collaborative filtering systems. The experiments show that AuthId book representation gives better precision and recall compared to LSI, LDA, VSM and Doc2vec. 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. Other ways for authorship identification should be investigated in the context of book recommendations. More advanced ranking methodologies could be adopted, such as pairwise or list-wise approaches. One possible approach is to adopt multi-task learning where a neural network has two outputs, namely 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 the system could work on a larger dataset with more user-item interactions.

Table 1: Author information of books similar to book #30376 using CNN_words

Book id (similarity)	Author information on NoveList
30376 (1)	Author: Nathaniel Hawthorne Genre: **Classics**; **Historical fiction** Character: Brooding; **Complex**; Flawed Storyline: Intricately plotted Pace: Leisurely paced; Tone: **Atmospheric**; Melancholy; **Thought-provoking**; Writing Style: **Descriptive**; **Richly detailed**; Stylistically complex Subject headings: Married women, Puritans – New England, Revenge,Sin, Physicians, Husband and wife, Pariahs, Clergy, Extramarital relations – New England, **Love triangles**, Atonement, Secrets, Ostracism, Villages – New England, Prynne, Curses, **Families**, Haunted houses Location:Massachusetts–History–Colonial period, Massachusetts, **New England**
1916 (0.98) 13707 (0.92) 25344 (0.95)	
2015 (0.88)	Author: G. K. Chesterton Genre: **Classics**; Literary fiction; Mysteries; Mystery classics; Short stories; Surrealist fiction Storyline: Unconventional Tone: **Thought-provoking** Writing Style: Compelling; **Descriptive**; Witty Subject headings: Anarchists, Conspiracies, Secret societies ... etc; Location: London, England, England
11104 (0.87) 4519 (0.87) 4518 (0.86) 4517 (0.86) 1263 (0.85)	Author: Edith Wharton Genre: Literary fiction; Modern classics Character: **Complex**; Storyline: Character-driven Tone: **Atmospheric**; Bittersweet; Strong sense of place Writing Style: **Descriptive**; Lyrical; **Richly detailed** Subject headings: Men/women relations, **Love triangles**, Married men, Marriage, Socialites, Separated women (Marital relations), Family relationships, Manners and customs Location: New York City – Social life and customs – 19th century, **New England**

Table 2: Author information of books similar to book #30376 using CNN_char_bigrams

Book id (similarity)	Author info on NoveList
30376 (1) 1916 (0.96) 25344 (0.959) 13707(0.914)	Author: Nathaniel Hawthorne previous table
22629 (0.775)	Author: Edward Elmer Smith Genre: Science fiction; Science fiction classics Storyline: Plot-driven; World-building Pace: Fast-paced Tone: Dramatic Subject headings: Space warfare, Human/alien encounters, Genocide, Life on other planets, Sexism ... etc.
2729 (0.759)	Author: H. Rider Haggard Genre: Adventure stories; **Classics; Historical fiction**; Science fiction; Science fiction classics Storyline: Action-packed; Plot-driven Pace: Fast-paced Tone: Strong sense of place Writing Style: **Richly detailed**
3735 (0.753)	Translated from author: Nicolay Gogol Genre Anthologies, **Classics**, Short stories, Translations Location: Russia, St. Petersburg, Russia
11605 (0.748)	Author: G.K. Chesterton previous table
2786 (0.739)	Author: Louisa May Alcott Genre: **Classics** Storyline: Character-driven Tone: Feel-good; Moving Subject headings: **Families**, Sisters, Young women, Girls – New England–History Location: **New England** – Social life and customs – 19th century
12204 (0.73)	Author: W.W. Jacobs Genre: **Classics**; Ghost stories; Horror; Horror classics Subject headings: Supernatural, Ghosts, Wishing and wishes, Fate and fatalism, Dead, Mummified animals . . . etc.

References

Haifa Alharthi, Diana Inkpen, and Stan Szpakowicz. 2017. A survey of book recommender systems. *Journal of Intelligent Information Systems*, pages 1–22, 9.

Mariona Coll Ardanuy and Caroline Sporleder. 2016. Clustering of Novels Represented as Social Networks. *LiLT (Linguistic Issues in Language Technology)*, 12(4).

Debasish Basak, Srimanta Pal, and Dipak Chandra Patranabis. 2007. Support Vector Regression. *Neural Information Processing – Letters and Reviews*, 11(10):478–486.

Sonia Bergamaschi and Laura Po. 2015. Comparing lda and lsa topic models for content-based movie recommendation systems. In Valérie Monfort and Karl-Heinz Krempels, editors, *Web Information Systems and Technologies: 10th International Conference, 2014, Revised Selected Papers*, pages 247–263. Springer.

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

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022, March.

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

David K. Elson, Nicholas Dames, and Kathleen R. McKeown. 2010. Extracting Social Networks from Literary Fiction. In *Proc. 48th Annual Meeting of the ACL*, pages 138–147, Stroudsburg, PA, USA. Association for Computational Linguistics.

Steve R. Gunn. 1997. Support Vector Machines for Classification and Regression (Image Speech & Intelligent Systems Group, University of Southampton). `http://m.svms.org/tutorials/Gunn1997.pdf`.

Shashank Gupta and Vasudeva Varma. 2017. Scientific Article Recommendation by Using Distributed Representations of Text and Graph. In *Proc. 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 1267–1268.

Kelly Hill. 2013. The Arts and Individual Well-Being in Canada, February. [Online; posted 13 February 2013].

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proc. 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.

Shibamouli Lahiri. 2014. Complexity of Word Collocation Networks: A Preliminary Structural Analysis. In *Proc. Student Research Workshop at the 14th Conference of the European Chapter of the ACL*, pages 96–105.

Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proc. 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1188–II–1196. JMLR.org.

Sheng-Ting Lin. 2017. *Latent semantic analysis for retrieving related biomedical articles*. Ph.D. thesis, University of British Columbia.

Stephen Macke and Jason Hirshman. 2015. Deep sentence-level authorship attribution. `https://cs224d.stanford.edu/reports/MackeStephen.pdf`.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

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):173–192.

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):407–428.

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

R. Nagori and G. Aghila. 2011. LDA-based integrated document recommendation model for e-learning systems. In *Proc. 2011 International Conference on Emerging Trends in Networks and Computer Communications*, pages 230–233.

Maria Soledad Pera and Yiu-Kai Ng. 2014a. Automating Readers' Advisory to Make Book Recommendations for K-12 Readers. In *Proc. 8th ACM Conference on Recommender Systems*, RecSys '14, pages 9–16.

Maria Soledad Pera and Yiu Kai Ng. 2014b. How Can We Help Our K-12 Teachers?: Using a Recommender to Make Personalized Book Suggestions. In *Proc. 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) – Volume 2*, pages 335–342. IEEE Computer Society.

Maria Soledad Pera and Yiu-Kai Ng. 2015. Analyzing Book-Related Features to Recommend Books for Emergent Readers. In *Proc. 26th ACM Conference on Hypertext & Social Media*, HT '15, pages 221–230.

Chen Qian, Tianchang He, and Rao Zhang. 2017. Deep Learning based Authorship Identification (report, Stanford University.

M. K. M. Rahman, Wang Pi Yang, Tommy W. S. Chow, and Sitao Wu. 2007. A Flexible Multi-layer Self-organizing Map for Generic Processing of Tree-structured Data. *Pattern Recognition*, 40(5):1406–1424, May.

Gerard Salton and Michael J. McGill. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

Thamar Solorio, Paolo Rosso, Manuel Montes-y-Gómez, Prasha Shrestha, Sebastián Sierra, and Fabio A. González. 2017. Convolutional Neural Networks for Authorship Attribution of Short Texts. In *Proc. 15th Conference of the European Chapter of the ACL, EACL 2017, Volume 2: Short Papers*, pages 669–674.

Efstathios Stamatatos. 2009. A Survey of Modern Authorship Attribution Methods. *J. Am. Soc. Inf. Sci. Technol.*, 60(3):538–556, March.

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*, pages 478–486. Proc. 3rd International Conference on Integrated Information.

Vladimir Vapnik, Steven E. Golowich, and Alex Smola. 1996. 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. MIT Press.

Paula Cristina Vaz, David Martins de Matos, and Bruno Martins. 2012a. Stylometric Relevance-feedback Towards a Hybrid Book Recommendation Algorithm. In *Proc. Fifth ACM Workshop on Research Advances in Large Digital Book Repositories and Complementary Media*, BooksOnline '12, pages 13–16.

Paula Cristina Vaz, David Martins de Matos, Bruno Martins, and Pavel Calado. 2012b. Improving a Hybrid Literary Book Recommendation System Through Author Ranking. In *Proc. 12th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '12, pages 387–388, New York, NY, USA. ACM.

Paula Cristina Vaz, Ricardo Ribeiro, and David Martins de Matos. 2012c. LitRec vs. Movielens – A Comparative Study. In *KDIR 2012 – Proc. International Conference on Knowledge Discovery and Information Retrieval, Barcelona, Spain, 4-7 October, 2012*, pages 370–373.

Yiwen Wang, Natalia Stash, Lora Aroyo, Laura Hollink, and Guus Schreiber. 2009. Using Semantic Relations for Content-based Recommender Systems in Cultural Heritage. In *Proc. 2009 International Conference on Ontology Patterns - Volume 516*, WOP'09, pages 16–28, Aachen, Germany, Germany. CEUR-WS.org.

J. Wang, C. Man, Y. Zhao, and F. Wang. 2016. An answer recommendation algorithm for medical community question answering systems. In *Proc. 2016 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pages 139–144, July.

Xuejun Yang, Hongchun Zeng, and Weihong Huang. 2009. ARTMAP-Based Data Mining Approach and Its Application to Library Book Recommendation. In *Proc. 2009 International Symposium on Intelligent Ubiquitous Computing and Education*, pages 26–29, May.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How Transferable Are Features in Deep Neural Networks? In *Proc. 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3320–3328, Cambridge, MA, USA. MIT Press.

Haijun Zhang and Tommy W. S. Chow. 2015. Organizing Books and Authors by Multilayer SOM. *Neural Networks and Learning Systems, IEEE Transactions on*, PP(99):1–14.

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