Book Recommendations Beyond the Usual Suspects

Embedding Book Plots Together with Place and Time Information

Julian Risch, Samuele Garda, and Ralf Krestel

Hasso-Plattner-Institut, Prof.-Dr.-Helmert-Str. 2–3, 14482 Potsdam, Germany
`firstname.lastname@hpi.de`

Abstract. Content-based recommendation of books and other media is usually based on semantic similarity measures. While metadata can be compared easily, measuring the semantic similarity of narrative literature is challenging. Keyword-based approaches are biased to retrieve books of the same series or do not retrieve any results at all in sparser libraries. We propose to represent plots with dense vectors to foster semantic search for similar plots even if they do not have any words in common. Further, we propose to embed plots, places, and times in the same embedding space. Thereby, we allow arithmetics on these aspects. For example, a book with a similar plot but set in a different, user-specified place can be retrieved. We evaluate our findings on a set of 16,000 book synopses that spans literature from 500 years and 200 genres and compare our approach to a keyword-based baseline.

Keywords: Recommender systems, Text mining, Document Embedding

1 Recommending Books Beyond the Usual Suspects

When users want to find a new book to read they typically trust best-seller lists or their favorite author. While this approach is easy to implement, the user will never discover serendipitous results or find hidden gems. We argue that users want to read books similar to those they enjoyed to read in the past. However, beyond the usual suspects, such as books of their favorite author, it is hard to find books with similar plots. Further, for more difficult searches with a lower recall base (i.e. fewer relevant books in the corpus) and increased data sparsity (only few information given for the existing books), there are no usual suspects. For this reason, we consider a recommendation task where the goal is to suggest similar books for a given example book. This task is typical in the scenario of users who just provided an initial book review, for example, at an online shopping platform. The task can also be interpreted as a search, where a user specifies an exemplar query. In that case, the query is an example of the books that the user is interested in. Central to both, book similarity searches and book recommendation tasks, is the way that similarity of books is defined.

Based on metadata information, books of the same author or the same genre can be recommended. However, assessing the semantic similarity of book plots goes beyond metadata comparison and comes with several challenges. Often times, recommender systems have no access to the full text of books, but only to synopses or abstracts. While a synopsis summarizes the entire plot line, an abstract tells only parts of the story and aims to motivate potential readers to buy the book. Besides these challenges, naive keyword-based similarity measures have two major disadvantages with regards to data sparsity: (1) books with similar plot but different wording cannot be found and (2) shorter book synopses reduce the chance of finding any similar books. Further, recommendations by a keyword-based approach are biased towards books of the same series, because they use the same words for, e.g., main characters. However, if a user searches for such books, there is no point in a semantic similarity search. The same results could also be retrieved with a metadata search for other books of the same author. To foster serendipity, a similarity measure for books that aims at retrieval and recommendation tasks should consider semantic similarity of the actual content: the plots, rather than metadata. To compare semantic similarity of different plots, an abstract representation of plots is needed. Even the full text description of two semantically similar books might not have many words in common. With shorter text descriptions (abstracts and synopses) this challenge of data sparsity gets even more difficult.

In this work, we propose a content-based recommender system that recommends similar books beyond books with the same keywords. Further, we allow users to specify aspects of requested similarity, but also of allowed dissimilarity. To this end, we define three aspects that can be searched for: plot, place, and time. We choose these aspects because plot, setting (place and time), and characters compose the three main elements of fiction. We neglect similarity of characters across books for two reasons. First, searches for the exact same character can be performed with keyword searches for their name. Typically such searches lead to books of the same author. Second, to find a similar (but not the same) character, the entire plot needs to be considered and the comparison of plots is already covered.

By embedding plot, place, and time in the same space we enable similarity searches based on these aspects. Further, we allow to search for book plots that are mixtures of two given book plots. To this end, we average embeddings of the two given book plots. The book whose plot embedding is closest to the calculated average is a mix of the two given plots. In addition, we enable arithmetics with books: Users can subtract and add places and times to book plots. As an example, a user might have read a crime story that is set in Greece at the time of 1900 and would like to read a similar crime story, but which is set in Portugal in 2018. Our approach enables such searches, because there is a representation for *Greece, Portugal, 1900, 2018,* and also for the given book's plot. From the vector for the given book, we can subtract the vectors for *Greece* and *1900* and add the vectors for *Portugal* and *2018*. The book that is closest to the result of the former calculation is recommended to the user.

We evaluate our recommendations in comparison to a bag-of-words (BoW) baseline and use the distance in the embedding space as a semantic similarity measure. To evaluate the semantic similarity of book synopses, we use the path distance between synsets (sets of one or more synonyms) in WordNet. Further, we evaluate arithmetics in the embedding space and give examples how this new way of book search retrieves relevant and surprising results — beyond the usual suspects. Our implementation of the embedding and the recommender system is open-sourced and published together with the used datasets online[1]. Our contributions are summarized as:

1. an algorithm to embed book plots, places, and times in the same space;
2. a recommender system based on arithmetics in this space;
3. experiments that compare a BoW approach and our approach at a recommendation task showing an increased WordNet similarity score by 7 percent.

2 Related Work

Relevance aspects for book search requests have been identified by Koolen et al. [7] based on previous work by Reuter [15]: Accessibility, content, engagement, familiarity, known-item, metadata, novelty, and socio-cultural background. With our book recommendation approach, we target the familiarity aspect of relevance, where books similar to known books shall be retrieved. However, for ease of use, familiarity relevance can be reduced to metadata relevance. To this end, the similarity measure for books is reduced to meta data only. As a result, only books by the same author or of the same year of publication are retrieved. Similarity of book titles could still be considered as metadata similarity. However, a comparison of titles becomes challenging if it focuses on semantic similarity. Latard et al. analyze how a search engine could profit from semantic similarity of keywords. However, their approach relies on semantic lexicons (WordNet, VerbNet). Due to limitations of these lexicons with regard to multi-word keywords, they are able to identify correct categories for only 22% of the articles [8]. In contrast, our unsupervised, embedding-based approach does not rely on any lexicons or encyclopedia to identify semantically similar words.

The idea of a semantic web has been extended to a web of books, which could connect logical concepts, figures, tables, and references in a semantic graph [6]. It is an open research question how rich semantic graphs can be automatically extracted from books to facilitate semantic searches. One approach to improve search in digital libraries of scientific publications is to generate additional metadata by applying topic models [11]. Depending on the domain, the scientific objective, used dataset, software, etc., can be extracted and clustered to facilitate semantic search. Similarly, Charalampous and Knoth classify document types to enrich meta data for improved search and recommendation results [5].

Bogers and Petras compare tags and controlled vocabularies (CV) at book searches with different information needs [3]. They find that tags and controlled

[1] `hpi.de/naumann/projects/web-science/book-recommendation.html`

vocabularies complement each other: CV work better if the search request is about a certain mood or reading experience. In contrast, tags work better for content-based search requests and for known-item searches. The authors also find that complex information needs in book search cannot be handled with tags or controlled vocabularies. They conclude that topical information in books needs other representations [4]. With our work, we propose such a representation in the form of an embedding space for plots, places, and times.

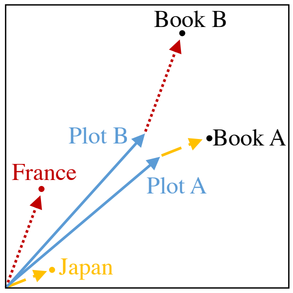
Another field of application for latent similarity measures are domains that use different words to describe similar concepts. For example, cross-collection topic models can reveal latent similarity of patents and scientific papers even if they do not have any words in common [16]. While a document's topic distribution is also a dense representation, our approach uses embeddings as document representations. Word embeddings have become a standard way to encode words for various downstream applications of natural language processing. However, how to obtain an embedding of a full document is still a topic of ongoing research. A naive way is to average all the vectors of all words in a document [17]. However, learning a dense vector representation for a document with paragraph2vec significantly outperforms word vector averaging as well as BoW approaches at information retrieval tasks [9]. A specific task of book recommendation is narrative-driven recommendation, where the users' interests are given as a narrative description [2]. In this work, we consider the plots of books that the user liked as a narrative description of interests.

3 Embedding Plots, Places, and Times in the Same Space

We propose to represent books as a composition of their plot, place, and time in the same embedding space. More specifically, each plot, each place, and each time is represented as a dense vector in the same 300-dimensional space. Further, each book is represented as the sum of its plot, place, and time in the same space. Our approach allows arithmetics in this space, so that the difference of two books can be interpreted with respect to plot, place, and time. This approach extends the idea of Mikolov et al. [13] from arithmetics on word embeddings to document embeddings and abstract concepts, such as plot embeddings. The similarity of two plots is calculated as the cosine similarity of their vector representations. Figure 1 visualizes a book A and a book B, which are composed of similar plots but different places. We can make use of this composition in the following way: When a user searches for a book that is similar to book B, we can recommend book A, because it has a similar plot. Moreover, when the user chooses a particular place, our approach recommends books with similar plots that are set at the specified place. In Figure 1, we search for a book that has a similar plot as book B and is not set in France but in Japan.

The basis of our embedding space are pre-trained word embeddings. We make use of the 300-dimensional *Wikipedia 2014 + Gigaword 5* embeddings published[2] by Pennington et al. [14]. Based on these word embeddings and the para-

[2] `nlp.stanford.edu/projects/glove/`



$$\text{Book A} = \text{Plot A} + \text{Japan}$$

$$\text{Book B} = \text{Plot B} + \text{France}$$

$$\text{Plot A} \approx \text{Plot B}$$

$$\text{Book A} \approx \text{Book B} - \text{France} + \text{Japan}$$

Fig. 1. Our embedding space allows to perform arithmetics on books, their plots, places, and times. Book A and B have a similar plot but are set at different places.

graph2vec approach by Le and Mikolov [9], we calculate document embeddings. Instead of considering all words of a book's synopsis for a book's embedding, we split the synopsis into three parts and generate three separate embeddings. We generate an embedding (1) for the set of words that describe the plot's place, (2) for the set of words that describe the plot's time, and (3) for all other words, which describe the plot independently of its place and time. The book itself is represented as the sum of these three representations.

3.1 Plot Representation

Given a book's synopsis, we apply named entity recognition to separate words that describe place or time. This separation allows us to consider only words that describe neither place nor time for the plot representation. Besides time and place terms, we also remove English stop words, which are not useful in discriminating individual plots. We generate a dense vector representation with paragraph2vec from the remaining words. Figure 2 shows a segment of a 2D-projection of the plot embedding space. Examples that we discuss are highlighted in black, others are grayed out. The vector space visualizations in this paper have been generated based on tensorflow's projector[3] and t-SNE dimensionality reduction [10]. For example, the two adventurous books *Adventures of Huckleberry Finn* and *The Adventures of Tom Sawyer* have similar plots and are close to each other. Further, the plots of *The Brothers Karamazov, Pride and Prejudice, The Sorrows of Young Werther,* and *Hamlet* have unfulfilled love and revenge with elements from tragedies and crime stories in common.

Interestingly, we can mix book plots by averaging their vector representations. In this scenario, a user provides two books as examples. These books can be of different genres. If we lookup the vector representations of the two books' plots in our embedding space, we can calculate their mean vector. This vector represents a mixture of the two books' plots. If we retrieve a book that has a plot vector close to that mean vector, we can recommend interesting mixtures.

[3] `projector.tensorflow.org/`

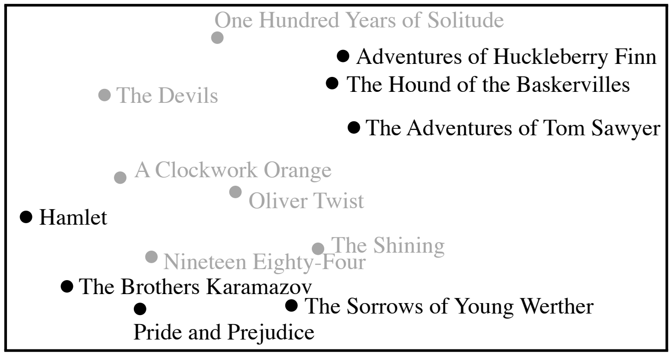


Fig. 2. Books with similar plot are closer to each other in the embedding space.

For example, mixing a crime fiction with a romance novel, we retrieve romantic suspense novels.

3.2 Place Representation

Besides the plot itself, we extract where the plot takes place. With named entity recognition, we extract names of politically or geographically defined locations, such as countries or cities. We lookup the word vector of each mentioned place from pre-trained word embeddings and average all such vectors to obtain an embedding of the book's place. Figure 3 shows a segment of a 2D-projection of the place embedding space. Close neighbors of *Portugal* are *Lisbon*, *Spain*, and *Catalonia.* Presumably because of the frequent term *United Kingdom*, the word *Kingdom* itself is close to *England* and *Britain.* African countries are closer to each other in the embedding space than to European countries. Based on these embeddings, our recommendation approach derives that two books are set in semantically similar places if one is set somewhere in Portugal and the other is specifically set in the Portuguese city Lisbon.

3.3 Time Representation

We extract also time information from book synopses with named entity recognition. For documents that contain no time information explicitly, we propose a different approach to estimate the time the book's plot is set in. To this end, we leverage an external knowledge base: Wikipedia. Every year has its own Wikipedia page[4], which describes important events in this year and also lists births and deaths of public figures. We analyze all these pages and index words that are specific to a subset of years. For example *Apollo 13* occurs only in the page for the year 1970, the year of the mission in the Apollo space program. *Portuguese Republic* occurs first in 1910, matching the proclamation of the first

[4] `en.wikipedia.org/wiki/List_of_years`

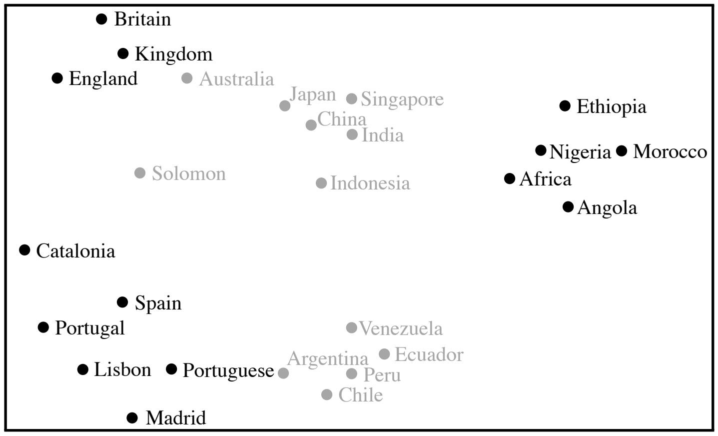


Fig. 3. Geographically close locations are closer to each other in the embedding space.

Portuguese Republic. Further, *Berlin Wall* first occurs in the page for 1961 and occurs for the last time in the page for 1990, which exactly matches the time frame from its construction to destruction. We train a naive Bayes classifier on the set of Wikipedia page texts and their corresponding years. As a consequence, given a text document as an input, such as a Wikipedia page, but also a book's synopses, we can predict a year. According to the training data, this year is likely to be mentioned together with the words in the input document.

Figure 4 shows a segment of a 2D-projection of the time embedding space. Although the years 1918 and 1945 are not consecutive, they are very close in the embedding space. Probably, this is because the two years mark the ends of World War I and II. The semantic similarity of the two years matches the idea of our recommendation approach: A user, who read a book that is set in 1945 might also want to read a book that is set in 1918, because of the similarity of the historic events at that time. As expected, years with a short time distance in between are also close in the embedding space, such as 1898, 1900, and 1912 or 1812, 1820, and 1821.

4 Experiments

With our experiments, we want to evaluate the semantic similarity of a given book and books recommended by our approach. This similarity is difficult to evaluate without a large user study among users, who are familiar with a large number of books. To still be able to evaluate our approach, we propose an automatic evaluation and further provide anecdotal evidence with examples. We consider a book synopses dataset[5] by Bamman and Smith [1]. The dataset describes 16,559 books by 4715 authors extracted from Wikipedia and Project

[5] www.cs.cmu.edu/~dbamman/booksummaries.html

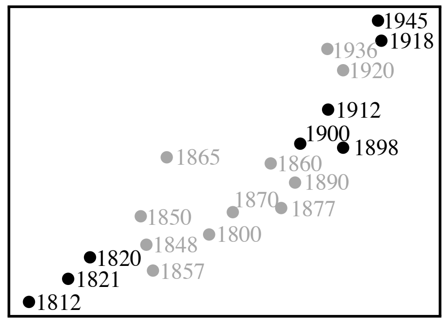


Fig. 4. Time-wise similar years are closer to each other in the embedding space.

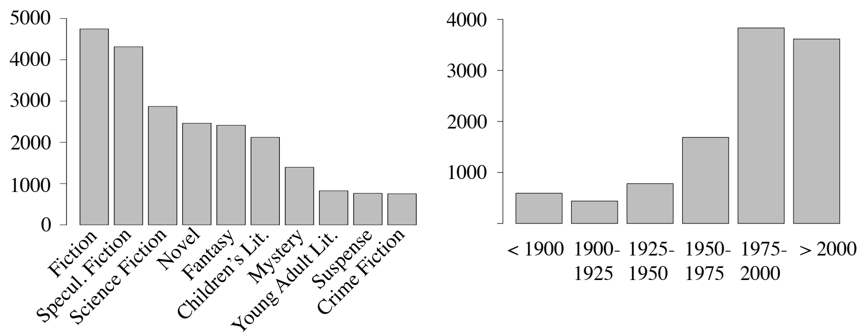


Fig. 5. The distribution of publication years and top 10 genres (out of 227) in the book dataset. Multiple genres can be assigned to the same book.

Gutenberg, along with aligned metadata from Freebase, including book author, title, and genre. Most of these books have been published between 1950 and 2000. 500 books are from the 19th century or older. Each synopsis contains about 430 words and after stop word removal about 260 words remain. Figure 5 visualizes the variety of the dataset with histograms for genre and publication year.

4.1 Evaluation Metric

The used metric is only an approximation of how users would judge semantic similarity of books. To automatically calculate the semantic similarity of an input book and each recommended book, we make use of WordNet[6]. To extend this metric from semantic similarity of pairs of words to pairs of full synopses, we follow the approach of Mihalcea et al. [12]. For each noun, verb, adjective, and adverb, we retrieve its synset (a set of one or more synonyms) from WordNet. For each synset, we identify the most similar synset from the other book and add its similarity score to the overall book similarity score. The synset similarity score is a path-based score in the range 0 to 1. It is based on the shortest path

[6] `wordnet.princeton.edu/`

that connects the synsets in the is-a (hypernym/hyponym) taxonomy. Finally, the book similarity score is the average of all maximum synset similarity scores. Because synset similarity is not a symmetric measure, we define the similarity of two book synopses A and B as:

$$(\mathrm{synset\_sim}(A, B) + \mathrm{synset\_sim}(B, A))/2.$$

4.2 Embedding-based Recommendation

The task of the following experiment is to recommend 10 books for a given input book. As a baseline to compare with, we implement a BoW approach. In particular, we implement a K-nearest-neighbor approach that is based on tf-idf weighted BoW representations of each document. To show that our approach improves on this baseline, we compare recommendations of the baseline against a combined approach of the baseline and our embedding-based approach. The BoW approach is supposed to make good recommendations if there is another book in the dataset with similar wording. However, we assume that our embedding-based approach excels if there are no such books in the dataset — or if they are considered non-relevant because such books are usual suspects. Therefore, if a recommendation of the BoW approach has only few words in common with the input book, we replace this recommendation by one of our embedding-based recommendations in the combined approach.

For a set of 50 randomly sampled books, both, the baseline approach and our combined approach, make 10 ranked recommendations each. As a result, the average score of the first recommendation is 0.467 for the BoW approach compared to 0.501 for our proposed approach (7% improvement). For the first 10 recommendations the score is 0.454 for BoW compared to 0.478 for our embedding-based approach. Our approach improves the semantic similarity of the input book and the top recommendations compared to a BoW baseline.

4.3 Plot Representation

To evaluate the dense vector representation of plots, we consider mixed book plots. For the following experiment, we sampled 10 pairs of books of genre *Crime Fiction* and *Romance Novel.* For each pair, we predict 5 recommendations with our embedding-based approach. The BoW baseline is not able to mix book plots.

Given the romance novel *Waking the Dead* and the crime fiction *Bones to Ashes* as input books, the fiction *Sons of Fortune* is the third-closest neighbor to the average vector of their plot embeddings. In *Sons of Fortune,* there are two twin brothers who fall in love with the same girl. Moreover, one of them is a lawyer and defends the other one on the charge of murder. A second example is the mix of the romance novel *A Passage to India* and the mystery, suspense, crime fiction *2nd Chance.* The closest book to their averaged plot embeddings is *Houseboy.* This book is both a love story and a crime story, but no genre information is designated in the dataset. Therefore, another application of our approach could be to automatically assign genres to books without any labels. In our dataset, 3718 books do not have any genre assigned.

4.4 Place Representation

With the following experiment, we examine how place embeddings affect book recommendations. Given the embedding of the book *Oliver Twist,* we subtract the embedding of its places and add the place embedding for *China.* The resulting vector's closest neighbor is the book *Spilled Water,* which is set in China. Further, both books, *Oliver Twist* and *Spilled Water* are about an orphan who is forced to work. Our approach correctly recommends a book with a similar plot that is set at a user-specified place. The average WordNet semantic similarity of the first 5 recommendations is very similar for our approach (0.561) and a BoW baseline (0.563). Our approach has the advantage that the place can be user-specified. Another input example is the book *Nineteen Eighty-Four.* If we search for books with similar plot but specify the location as *China,* our approach recommends *When the People Fell.* The latter is a Science Fiction story about the colonization of Venus by a future Chinese government. Although the location is not as requested, the recommendation is interesting because of the connection to China. Further, both books are about obedience to authority and therefore have similar plot. Given the book *The Whiskey Rebels* and adding the vector for *Italy,* our approach recommends *Wings of the Falcon,* which is set in Italy. Both books are about rebelling against power.

4.5 Time Representation

Besides the place of a book's plot, users can specify its time. As an input example, we consider *A Farewell to Arms* by Hemingway, which is about a soldier in World War I. The first recommendation of our approach, if we add the vector for *1944,* is *The Wolf's Hour.* This book has a similar plot, but is set in World War II. Another example is *Adventures of Huckleberry Finn,* which is about the adventures of a child around 1850. If we add the vector for *1960*, the first recommendation is *Summer of Night.* This book is about children's adventures set in 1960s. If we request a specific year, for example *2010*, not all recommended books mention exactly this year, but very close ones, such as *2008.* Our approach correctly derives that a plot set in 2008 is time-wise similar to a plot set in 2010.

4.6 Movie Recommendation as a Similar Task

To show that our approach is applicable to other data, we run additional experiments for the similar task of movie recommendation. Similar to the book dataset, we extracted a movie dataset from Wikipedia pages and published it online[7]. The dataset contains 6456 movies from the years 2000 to 2016 extracted from Wikipedia. Each movie is described by about 1340 words. Figure 6 visualizes a segment of a 2D-projection of the movie plot embedding space. *Star Trek* and *Star Wars* both are about space adventures and are therefore located close to each other. Interestingly, *The Ring* and *The Lord of the Rings* are separated

[7] hpi.de/naumann/projects/web-science/book-recommendation.html

from each other, although the titles have the word *Ring* in common. Indeed, the plots of the two movies are very different and while a ring is centric to the story of *The Lord of the Rings, Ring* has only a symbolic meaning in *The Ring.* A keyword-based approach would assume that both movies are similar, because a word in their titles overlaps. However, in the embedding space, *The Ring* is close to *Harry Potter and the Philosopher's Stone,* which makes sense because both movies are about a child with supernatural or magical power.

Similar to the experiments on the book dataset, we compare WordNet similarity scores of a BoW baseline with our combined, embedding-based approach. For 50 movies, the average score of the first recommendation is 0.547 for the BoW approach compared to 0.566 for our proposed approach. For the first 10 recommendations the score is 0.540 for BoW compared to 0.562. The improved semantic similarity of recommendations and the input document shows that our approach is applicable to other data beyond books.

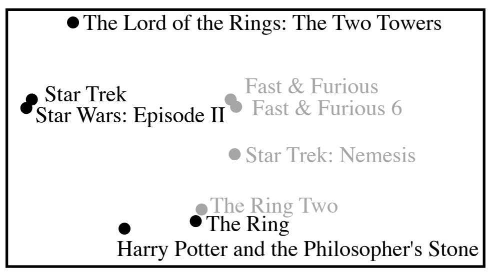


Fig. 6. Movies with similar plot are closer to each other in the embedding space.

5 Conclusions and Future Work

We proposed to embed book plots and their setting in the same space. In this space, we can do arithmetics to express book searches and to extend recommendations beyond the usual suspects. In contrast to a BoW baseline, our embedding-based approach is able to retrieve similar books that do not have any words in common. We find that embeddings achieve semantically more similar recommendations on datasets of books and movies. The semantic similarity of book synopses is evaluated based on the path distance between synsets (sets of one or more synonyms) in WordNet. Last but not least, we allow users to specify place and time when they search for books with similar plot.

Future Work could improve the extraction of place and time information or could add more aspects to the embedding space. For example, for crime stories the murder weapon could be extracted and represented in the same space. Another idea is to use hierarchical word embeddings as the basis for places and times. Thereby, the hierarchical relationship of *21st century* and *2018* or *Asia* and *Japan* could be represented. Last but not least, a detailed user study could evaluate how users interact with and search in the proposed embedding space and how satisfying our recommendations are.

References

1. Bamman, D., Smith, N.A.: New alignment methods for discriminative book summarization. In: arXiv:1305.1319 (2013)
2. Bogers, T., Koolen, M.: Defining and supporting narrative-driven recommendation. In: Proc. of the Conf. on Recom. Systems (RecSys). pp. 238–242. ACM (2017)
3. Bogers, T., Petras, V.: An in-depth analysis of tags and controlled metadata for book search. iConference 2, 15–30 (2017)
4. Bogers, T., Petras, V.: Supporting book search: A comprehensive comparison of tags vs. controlled vocabulary metadata. Data and Information Management 1, 17–34 (2017)
5. Charalampous, A., Knoth, P.: Classifying document types to enhance search and recommendations in digital libraries. In: Proc. of the Int. Conf. on Theory and Practice of Digital Libraries (TPDL). pp. 181–192. Springer (2017)
6. Khusro, S., Ullah, I.: Towards a semantic book search engine. Proc. of the Int. Conf. on Open Source Systems & Technologies (ICOSST) pp. 106–113 (2016)
7. Koolen, M., Bogers, T., van den Bosch, A., Kamps, J.: Looking for books in social media: An analysis of complex search requests. In: Advances in Information Retrieval. pp. 184–196. Springer (2015)
8. Latard, B., Weber, J., Forestier, G., Hassenforder, M.: Towards a semantic search engine for scientific articles. In: Proc. of the Int. Conf. on Theory and Practice of Digital Libraries (TPDL). pp. 608–611. Springer (2017)
9. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proc. of the Int. Conf. on Machine Learning (ICML). pp. 1188–1196. JMLR (2014)
10. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research 9, 2579–2605 (2008)
11. Mesbah, S., Fragkeskos, K., Lofi, C., Bozzon, A., Houben, G.J.: Facet embeddings for explorative analytics in digital libraries. In: Proc. of the Int. Conf. on Theory and Practice of Digital Libraries (TPDL). pp. 86–99. Springer (2017)
12. Mihalcea, R., Corley, C., Strapparava, C., et al.: Corpus-based and knowledge-based measures of text semantic similarity. In: AAAI. vol. 6, pp. 775–780 (2006)
13. Mikolov, T., Yih, W.t., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proc. of the Conf. of the North American Chapter of the ACL. pp. 746–751. ACL (2013)
14. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014)
15. Reuter, K.: Assessing aesthetic relevance: Children's book selection in a digital library. Journal of the Association for Information Science and Technology 58(12), 1745–1763 (2007)
16. Risch, J., Krestel, R.: What should i cite? cross-collection reference recommendation of patents and papers. In: Proc. of the Int. Conf. on Theory and Practice of Digital Libraries (TPDL). pp. 40–46. Springer (2017)
17. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP). pp. 1631–1642. ACL (2013)