SIRUP: Search-based Book Recommendation Playground

Ghazaleh Haratinezhad Torbati
ghazaleh@mpi-inf.mpg.de
Max Planck Institute for Informatics
Saarbrücken, Germany

Anna Tigunova
tigunova@mpi-inf.mpg.de
Max Planck Institute for Informatics
Saarbrücken, Germany

Gerhard Weikum
weikum@mpi-inf.mpg.de
Max Planck Institute for Informatics
Saarbrücken, Germany

ABSTRACT

This work presents a playground platform to demonstrate and interactively explore a suite of methods for utilizing user review texts to generate book recommendations. The focus is on search-based settings where the user provides situative context by focusing on a genre, a given item, her full user profile, or a newly formulated query. The platform allows exploration over two large datasets with various methods for creating concise user profiles.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

recommender system, search-based recommendation, user profile

1 INTRODUCTION

Motivation: Content-based recommenders have been extensively researched, with utilization of item features like categories or genres, as well as user features, particularly user-provided reviews of items. The latter is of prime interest for domains where simpler interaction signals (likes, ratings, clicks) are sparse, the item distribution has a long tail, user interests are diverse, and (some) user reviews are textually rich. This work focuses on book domain, that has the aforementioned characteristics. The key issue here is how to distill informative cues from the noisy, lengthy and sometimes irrelevant texts in user reviews, represent them in a concise user profile, and integrate these signals into an end-to-end recommendation model.

Prior works have studied only simple techniques for coping with noisy reviews, such as feeding the entire sequence through a CNN or running multiple text chunks through a transformer with subsequent pooling. Experimental results on this setting thus cover limited ground and are scattered in the literature, leaving big gaps in comparing different methods.

Contributions: This work presents an interactive playground for exploring and qualitatively comparing a suite of methods for leveraging user reviews in book recommendations with search-based context. While this does not address the need for comprehensive experimental studies, our framework and platform are important steps for creating, altering and comparing different ways of building user profiles, and obtaining qualitative insights on how they affect the quality of search-based recommendations. Our methodology comprises several novel techniques for distilling noisy user reviews into concise profiles.

We present the SIRUP (Search-based Interactive Recommendations with User Profiles) platform for book recommendations using review texts of the readers in large online communities, available at https://sirup.mpi-inf.mpg.de/. A distinctive feature of SIRUP is the profile construction step, which builds a concise profile (of bounded length, like a few hundreds tokens) from the entirety of a user's reviews (often spanning thousands of tokens). Such a compact representation is useful to limit the computational and energy cost of running text through a transformer-based model, and also to distill the full user text into its informative gist [11]. The SIRUP platform offers a wide range of profile construction methods, including selection of informative sentences or n-grams, as well as methods that use large language models (GPT or T5) to generate readable summaries of user reviews.

We focus on the less explored setting where the recommendation is conditioned on a situative search by the user. This can take different forms: an explicit query with keywords or fully narrative text, referring to a specific book, or asking the system to use the automatically constructed concise user profile. The system can feed such context into an IR engine to restrict the space of recommendation candidates before transformer-based re-ranking.

2 ARCHITECTURE AND IMPLEMENTATION

2.1 Profile Construction

Book reviews are often long and noisy, containing informative cues about book contents and why the reader likes it, along with personal remarks (e.g., "I am busy with my three kids") and sentiments (e.g., "I kept reading the whole night") that are useless for capturing the reader's interests and taste. Thus, judiciously selecting the best pieces from a reader's collection of reviews and generating a concise gist is the key task here. SIRUP provides a suite of techniques for this purpose, covering both simple heuristics and more advanced options, inspired by [10].

The profile construction takes as input all reviews of a reader, together with the metadata of the corresponding items (titles and categories/genres), and outputs a concise text of the desired target length (128 tokens in our current configuration). In the demonstration we showcase the following profiling methods:

- **genres**: set of genres/categories for items of the reader;
- **N-gram**: a set of word-level N-grams with the highest idf scores based on Google N-grams;
- **idf**: top sentences by their normalized unigram idf scores;
- **SBERT**: top sentences by Sentence-BERT [8] similarities with the corresponding item of each review;
- **T5-keywords**: T5-generated keywords, from all reviews of the reader;
- **ChatGPT**: ChatGPT-generated summaries, from all reviews of the reader;

2.2 Transformer-based Item Ranking

To generate recommendations, we use a *two-tower* architecture with a BERT encoder, a well-established approach to neurally represent readers and items. We initialize BERT with the pre-trained model and specifically fine-tune it on our book review data.

Note that the training labels are binary: positive if the reader wrote an appreciative review, or negative, by default, if the item is not in the reader's book collection. The reason is that almost all reviews in book communities are positive and have only minor variations in their numeric ratings.

The latent representations for the readers and the items are produced by the two towers separately, and the relevance score of an item for a given reader is produced by calculating a dot product between their representations. The top items by the dot product scores serve as selected recommendations. For a more detailed description of the model architecture, see [11].

2.3 Search-based Recommendation

In addition to the long-term reader profile, which is used for tailoring predictions to the reader's tastes, SIRUP allows the users to provide situative context, reflecting current information needs. This context is used to short-list a pool of items that is finally re-ranked by the recommendation model.

To get the context-based pool of candidates, we use BM25 algorithm to obtain top-100 items by the lexical match scores between the provided context and the item texts. In Section 3.2 we provide details on the different search modes.

2.4 Data

SIRUP covers two datasets with book reviews: *Amazon (AM)* and *Goodreads (GR)*, both from the UCSD repository [4]. To train the two-tower recommender model, we selected 1k text-rich users by average review length, and all items associated with them (resulting in 16k and 45k items for AM and GR, respectively).

To increase the selection of items for recommendation, we sample most reviewed items, expanding both datasets to contain around 100k items each. With strongly skewed distribution in popularity, this means that most items are completely unseen at training time: a challenging case for a recommender system, underlining the necessity of having informative and concise user profiles.

2.5 Experiments

To illustrate how different profile selection methods influence the quality of recommendations, we benchmarked their performance on our datasets. Table 1 shows micro-averaged NDCG@5 results,

with drill-down into items seen during training (for some reader, but not the current test reader) and completely unseen items. Note that the data is very sparse, so the absolute NDCG@5 numbers are much lower than typical numbers for recommender benchmarks with dense data (e.g., MovieLens). Table 1 shows that different methods perform similarly, with generative methods slightly superior (T5-generated keywords on Amazon and ChatGPT on Goodreads). Notably, results on unseen items do not degrade much. We attribute this nice effect to the expressiveness of training-time reviews (for other books) and the constructed profiles which can be compared against candidate items at prediction-time.

Table 1 also includes a baseline, denoted "BM25 kw", that uses the T5-generated keywords as reader profile and simply ranks exact text matches by a BM25 scoring model, widely used in information retrieval, without any neural inference at run-time. We see that this alone does not work well, so transformer inference for the final predictions is essential.

Method	Amazon			Goodreads		
	All	Seen	Unseen	All	Seen	Unseen
genres	7.0	8.98	6.84	11.18	18.16	9.71
3-gram	8.98	11.11	8.8	13.19	16.61	12.47
idf	9.14	10.94	8.99	13.76	18.36	12.79
SBERT	9.0	12.68	8.7	13.51	18.03	12.55
T5-kw	9.5	12.65	9.24	13.76	20.28	12.39
ChatGPT	8.93	10.99	8.76	13.78	21.28	12.2
BM25 kw	3.8	2.44	3.91	1.44	1.0	1.54

Table 1: NDCG@5 for Search-based Evaluation

The SIRUP playground is optimized for speed and footprint, as the concise reader profiles reduce the model's input to 128 tokens. Experiments have shown that this size limit results in nearly no loss in quality compared to, for example, processing multiple 512-token chunks with max-pooling or other aggregation [11].

3 DEMONSTRATION PLATFORM

In this section we describe the SIRUP playground, available at https://sirup.mpi-inf.mpg.de/. We will cover profile construction for existing and new readers, as well as specifying situative search context and interpreting the recommendation results.

3.1 Constructing User Profiles

The reader's profile, based on her history of book reviews, is used to personalize the ranking of recommendations. In our demo, such a profile can be defined either by selecting an existing reader from the original datasets (from a drop-down menu), or by manually creating a customized profile for a new user.

Existing User: Figure 1 shows the setup for selecting an existing reader, where each reader in the drop-down menu is characterized by a one-line punchy description. These descriptions, generated by ChatGPT from all reviews of a reader, help to get a feel for the selected reader's interests and tastes.

For each reader, the demo also shows how much they have read: we construct three user groups: *novice* (< 5 books for AM and < 13 books for GR), *fan* (between 5 and 20 books for AM, between 13
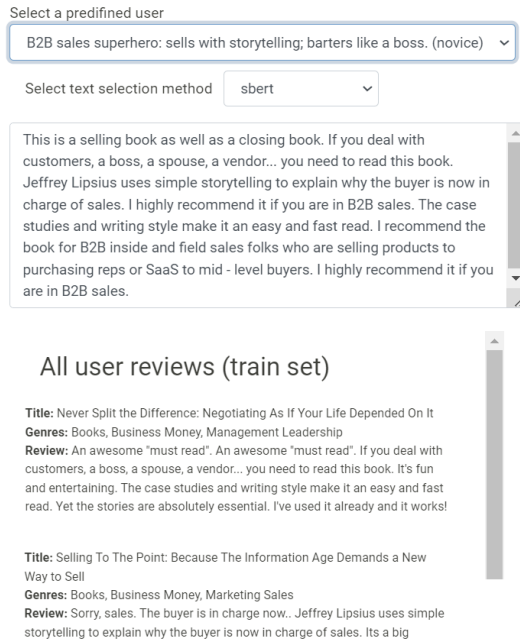
Figure 1: Profile Construction for Existing Users.

and 71 books for GR), and *ambassador* (with more than 20 or 71 books, respectively).

For instance, reader Zoe, who reviewed the books "Selling To The Point: Because The Information Age Demands a New Way to Sell" and "Never Split the Difference: Negotiating As If Your Life Depended On It", is described by ChatGPT as "*B2B sales superhero: sells with storytelling; barters like a boss*". In total Zoe has read 3 books and is considered a *novice* in the playground. We take Zoe as our running example throughout this section.

To distill all reader's reviews into a concise profile and eliminate useless content (e.g. Zoe's emotional phrases, like "…an awesome 'must read' … "), the demo offers several methods for profile construction, as described in Section 2.1. In our example we pick the SBERT method, which selects review sentences that are semantically closest to the descriptions of the books themselves (provided by the AM and GR platforms).

New User: In the manual profile mode, a user has two options to create the profile: i) typing a concise text about the user's interests and taste, or ii) entering books that the user has read and liked, providing short review texts for them.

In the latter option the user can select existing items by title (using auto-complete suggestions). When all books and reviews are entered, the user can trigger the construction of the concise profile using one of the available methods. The user can also save the entered books and reviews as a CSV file, and later load it again for a new session with the constructed profile.

In both modes, the reader's profile can be edited, allowing users to modify it and add information. The text of the concise profile serves as input to the transformer-based personalized recommender model.
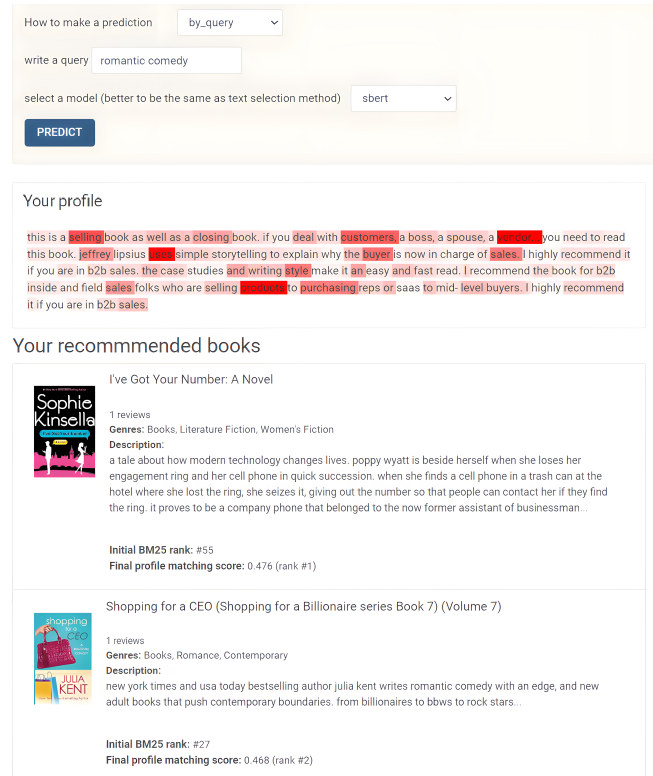

Figure 2: Query-based context and predictions.

3.2 Defining Search Contexts

In addition to leveraging the system-generated profiles, users can define a short-term search context, to select a matching subset of items, which are subsequently ranked by our trained transformer. The interface for defining search context is shown in Figure 3. The user can choose among different search modes.

Query-based: In this mode, the user can specify a free-form textual query. The resulting candidate pool of top-100 results by BM25 is then re-ranked using the transformer model. For example, Zoe queried for "romantic comedy", getting as top recommendations some light-reading romantic novels, which connect to business topics that are expressed in Zoe's profile (see Figure 2).

Genre-based: The user can filter books by categories/genres: for Amazon the top-50 most popular genres, for Goodreads all 10 genres available in the dataset. For instance, when Zoe selects the genre "Biographies, Memoirs" as a filter (see Figure 3), the system recommends biographies of business innovators, which fit well with her interests.

Item-based: In the item-based mode, the user selects an item from a drop-down menu, using the entire catalog of 100K books. The user can type (prefixes of) words in book titles, and suggestions are shown by auto-completion.

Profile-based: This mode uses the generated reader profile as a query to form the initial pool of candidates, which are then ranked by the trained transformer model.

Figure 3: Genre-based context and predictions.

All Items: In this mode the two-tower model ranks all items in the book collection for the given reader.

3.3 Recommendations

The results shown to the user consist of two parts (see Figure 3): i) the input profile with highlighted salient keywords identified by the model, and ii) the list of recommended books.

Weighted Input Terms: We compute the significance of the user's input terms by utilizing BERT attention weights from the last layer. To obtain the final weights, we max-pool the weights across all attention heads and sum the resulting matrix row-wise. Note, that these weights are just an intermediate product of the two-tower model and are not directly used for matching against terms of the recommended books.

For our reader Zoe, words like "selling" and "customer" will be identified among the most significant terms. Note that different text selection methods influence the choice of the important terms (ChatGPT summary in Figure 3 vs. SBERT sentences in Figure 2).

List of Book Recommendations: The list of recommendations consists of books, ranked by their similarity scores with respect to reader's profile, calculated by the two-tower transformer. In addition to these scores, we show the books' ranks from the initial BM25 scoring when applicable (in query-, item- or profile-based search modes). Comparing the two rankings side-by-side can provide insights on how the text-similarity re-ranking changes with

profile-based personalization. For example, for Zoe's query (Figure 2), the top items after personalized re-ranking had ranks 55 and 27 by the initial candidate search with BM25.

Each item in the resulting recommendation list has metadata, including title, genres, description and the cover image of the book. For each item we specify the total number of reviews available for it; 0 reviews means that the item was *unseen* during model training.

4 RELATED WORK

Text-based Recommendations: The methods for content-based recommendations use a variety of inputs, such as item metadata, with textual user reviews being the most descriptive and rich. Previous works utilizing simple neural architectures, such as CNNs, proved to work very well [12]; however, currently most text-based models are based on transformer architectures, including BERT [6] or T5 [3]. Following the recent studies we select BERT as our backbone text similarity model.

Demo Systems: Most existing demonstration platforms showcase the interaction-based recommender systems [7, 9]; in such systems the user profile is defined as a set of the items they interacted with [9] or a set of tags [2]. Few text-based demonstrations allow the users to enter their text in a query-like form [1, 5], which is used to return top recommendations, but do not build a persistent user profile. To the best of our knowledge, we propose the first platform that recommends the items based on both long-term textual profiles and short-term situative context.

5 CONCLUSION

We presented the SIRUP playground for text-based book recommendations, accessible at https://sirup.mpi-inf.mpg.de. SIRUP supports exploring a suite of techniques for constructing reader profiles and how they influence recommendations based on various plug-in models. The architecture of SIRUP allows straightforward transfer from books to other domains of interest.

REFERENCES

[1] Veronika Arefieva, Roman Egger, Michael Schrefl, and Markus Schedl. 2023. Travel Bird: A Personalized Destination Recommender with TourBERT and Airbnb Experiences. In *WSDM '23*.
[2] Joey De Pauw, Koen Ruymbeek, and Bart Goethals. 2022. Who do you think I am? Interactive User Modelling with Item Metadata. In *Recsys '22*.
[3] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *RecSys '22*.
[4] Julian McAuley. 2022. *Recommender Systems and Personalization Datasets*. https://cseweb.ucsd.edu/~jmcauley/datasets.html, accessed on Sep 29, 2022.
[5] Diana Andreea Petrescu, Diego Antognini, and Boi Faltings. 2021. Multi-step critiquing user interface for recommender systems. In *Recsys '21*.
[6] Reinald Adrian Pugoy and Hung-Yu Kao. 2021. Unsupervised Extractive Summarization-Based Representations for Accurate and Explainable Collaborative Filtering. In *ACL-IJCNLP '21*.
[7] Naime Ranjbar Kermany, Jian Yang, Jia Wu, and Luiz Pizzato. 2022. Fair-srs: a fair session-based recommendation system. In *WSDM '22*.
[8] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP-IJCNLP '19*.
[9] Jan Šafařík, Vojtěch Vančura, and Pavel Kordík. 2022. RepSys: Framework for Interactive Evaluation of Recommender Systems. In *Recsys '22*.
[10] Ghazaleh Haratinezhad Torbati, Anna Tigunova, and Gerhard Weikum. 2023. Unveiling Challenging Cases in Text-based Recommender Systems. In *PERSPECTIVES Workshop at Recsys '23*.
[11] Ghazaleh Haratinezhad Torbati, Anna Tigunova, Andrew Yates, and Gerhard Weikum. 2023. Recommendations by Concise User Profiles from Review Text. *arXiv preprint arXiv:2311.01314* (2023).
[12] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *WSDM '17*.