ML-1M++: MovieLens-Compatible Additional Preferences for More Robust Offline Evaluation of Sequential Recommenders

Kazutoshi Umemoto
The University of Tokyo
Tokyo, Japan
umemoto@tkl.iis.u-tokyo.ac.jp

ABSTRACT

Sequential recommendation is the task of predicting the next interacted item of a target user, given his/her past interaction sequence. Conventionally, sequential recommenders are evaluated offline with the last item in each sequence as the sole correct (relevant) label for the testing example of the corresponding user. However, little is known about how this sparsity of preference data affects the robustness of the offline evaluation's outcomes. To help researchers address this, we collect additional preference data via crowdsourcing. Specifically, we propose an assessment interface tailored to the sequential recommendation task and ask crowd workers to assess the (potential) relevance of each candidate item in MovieLens 1M, a commonly used dataset. Toward establishing a more robust evaluation methodology, we release the collected preference data, which we call ML-1M++, as well as the code of the assessment interface.[1]

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

sequential recommendation, offline evaluation, preference assessment, crowdsourcing

ACM Reference Format:
Kazutoshi Umemoto. 2022. ML-1M++: MovieLens-Compatible Additional Preferences for More Robust Offline Evaluation of Sequential Recommenders. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557643

1 INTRODUCTION

Sequential recommendation aims at predicting the item with which a target user will interact next, given his/her past interaction sequence [6]. While early work used Markov chains to learn the transition of user preferences over time [24], the major approach has shifted to neural networks recently due to their great potential for sequential modeling and representation learning [12, 14, 30, 36]. Conventionally, this task adopts an offline evaluation that investigates whether the last item in each sequence can be predicted. As such, there is only one item known to be relevant to the user, which is the most severe case of the positive unlabeled problem [22].

How robust is the conventional evaluation that relies on the sparse preference data? Suppose that two sequential recommenders rank the known relevant item at the same position. The relevance of other items are unknown since they were not interacted with at the target time point. Even if one recommender ranks more *potentially relevant* (i.e., unlabeled but seemingly positive) items higher than the other, the conventional evaluation cannot tell the difference.

Motivated by this issue, we densify preference data with the potential relevance of other items toward establishing a more robust evaluation methodology. Recently, Lu et al. [21] showed the potential of external preference assessments collected in a laboratory study for evaluating general recommenders. Inspired by their work, we propose an assessment framework that uses crowdsourcing to improve scalability and is tailored to the sequential setting.

The main contributions of this work are as follows. **(1)** We propose a crowdsourcing-based framework to collect external preference assessments for evaluating sequential recommenders. We consider several design factors and identify their best combination in terms of the assessment quality measured automatically. **(2)** Selecting MovieLens 1M [8] as our target dataset, we collect large-scale assessments consisting of the potential relevance of candidate movies (as items) for each user. We release the collected data, which we call ML-1M++, as well as the code of the assessment interface and discuss future research directions enabled by these resources.

2 POOLING CANDIDATE ITEMS

We need to prepare candidate items that may be potentially relevant, for which crowd workers provide preference assessments. To this end, we apply the pooling method [26], which is widely used by the information retrieval community to obtain documents to be assessed, to the predictions of multiple sequential recommenders.

2.1 Dataset

A wide variety of datasets have been used for evaluating sequential recommenders [7, 12, 18, 19, 23, 28, 30, 36], including MovieLens [8], Last.fm [4, 27], Foursquare [37, 39], Gowalla [5], Amazon [10], and Steam [14]. We need to carefully select a target dataset so that we can collect reliable assessments from crowd workers. For this purpose, we consider two requirements that the target dataset should satisfy. **(1) Dataset domain.** Unlike professional assessors, crowd workers may not have specialized knowledge and/or skills for assessments. Thus, the domain of the target dataset should be familiar to the general public. **(2) Item details.** While a number of datasets

[1] Our resources are available here: https://umemotsu.github.io/ml1mpp-portal/

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557643

only include anonymized identifiers as item information, the lack of details about items makes the external assessments impossible. Thus, sufficient information about items should be available.

Taking the aforementioned requirements into consideration, we selected MovieLens 1M [8][2] as our target dataset. (1) This dataset contains user ratings on the movie domain, where many crowd workers, we believe, would have some familiarity and interest. (2) It includes the title and genres of each movie. We can retrieve further details from online movie databases such as IMDb[3] and TMDB[4].

We followed the common preprocessing practice [9, 14, 24, 28]. Specifically, we treated each rating event as implicit feedback from the user and constructed his/her interaction sequence by sorting the rated movies in ascending order of the event timestamps. We iteratively discarded rare users and items having less than five interactions. Finally, we excluded the last item from each user sequence so that our assessment interface can use it as contextual information (cf. *the next item* in Section 3). The preprocessed dataset contains 993 571 interactions between 6040 users and 3416 items.

2.2 Recommenders

The diversity of candidate items is as important as the potential relevance to collect external preference assessments that can be used for a fairer, less biased evaluation. Thus, we selected eight sequential recommenders representing *different approaches and architectures*: FPMC [24] and TransRec [9] from *non-neural* models, GRU4Rec [29] from *RNN* [25]-based models, NextItNet [38] from *CNN* [15]-based models, NARM [16] and STAMP [20] from models adopting the *attention* mechanism [2], and SASRec [14] and BERT4Rec [28] from *Transformer* [31]-based models.

Note that several of the selected recommenders were originally proposed for session-based recommendation [35], which is slightly different from sequential recommendation. We included them since both tasks have been addressed with similar approaches. For simplicity and a fair comparison, we did not select sequential recommenders incorporating side information (e.g., item features [11, 40], time features [17], and knowledge bases [12, 34]) in this work.

2.3 Training and Pooling Settings

We followed the conventional leave-one-out setting [14, 28, 36] to divide the dataset into three sets: the last item in the sequence for testing, the second last item for validation, and the remaining for training. We used RecBole (version 1.0.0) [41]'s implementation for our selected recommenders. During the training, the six neural recommenders were optimized with the cross-entropy loss while the two non-neural ones (i.e., FPMC and TransRec) were optimized in a pairwise manner, in accordance with the original papers [9, 24]. For the pairwise optimization, we sampled one negative item uniformly at random for each training example. We tuned the hyperparameters of each recommender via a grid search by using the validation set of the dataset. We stopped training each recommender when its validation performance measured by Reciprocal Rank [33] for the top-10 ranking (RR@10) did not improve for 10 epochs in a row.[5]

[2]https://grouplens.org/datasets/movielens/1m/
[3]https://imdb.com/
[4]https://www.themoviedb.org/
[5]For reproducibility, our released resources[1] include the information about the search space and the hyperparameter values chosen for each recommender.

After training all recommenders, we prepared candidate items by using the pooling method [26]. Specifically, we selected the top N_{pool} recommendation results of each recommender for the testing example of each user. In this work, we set the pooling depth to $N_{\text{pool}} = 3$, following Lu et al. [21]. As a result, we obtained on average 14.5 movies as candidate items for each user.

3 COLLECTING PREFERENCE ASSESSMENTS

Inspired by Lu et al. [21], we rely on external assessors to collect additional preferences for candidate items prepared in the previous section. As reviewed in Section 1, however, their approach cannot be directly applied to this work due to three reasons. **(1) Scalability.** They collected external assessments in a laboratory study, where participants (16 users and 19 assessors) spent two hours on average. Laboratory studies are not *scalable to existing datasets for offline evaluation*, which typically contain the interactions of thousands of users or more. **(2) Quality.** They collected preference data from not only assessors but also users to measure the quality of the assessments. As it is difficult to communicate with users appearing in offline evaluation datasets, we need a method that can *automatically measure the assessment quality*. (3) **Sequentiality.** Last but not least, their approach was intended for general (non-sequential) recommendation scenarios; they showed assessors users' historical interactions that were sampled by popularity and not ordered by interaction timestamps. We need to *design an interface tailored to sequential recommendation* with which assessors can estimate users' dynamic preferences that may evolve over time.

To overcome the scalability challenge, we adopt crowd workers as external assessors, which makes the quality challenge more important since crowdsourcing is less controllable than laboratory studies. With these challenges in mind, we design an assessment framework tailored to sequential recommendation.

3.1 Conditions

As users' preferences may change over time, understanding their latest preferences plays a key role in judging which item they are likely to interact with at the target time point. In this work, we consider two contextual factors that may help assessors go through the understanding process. **(1) Past items.** We present the N_{past} most recent items with which the target user interacted before the target time point. The hyperparameter N_{past} controls the tradeoff between the richness of historical context and the cognitive load of assessors. Balancing this tradeoff, we compare two different values in our study: $N_{\text{past}} \in \{10, 20\}$. **(2) Next item.** We also consider presenting the item with which the target user interacted right after the target time point. Our rationale behind this factor is as follows: estimating missing data at an intermediate position of a sequence would be easier than at the last position since candidates can be reduced from both sides in the former. Unlike the past items, increasing the number of the next items, I_{next}, by moving up the target time point decreases the amount of training data. Thus, we focus only on the last item in each sequence and compare its presence ($I_{\text{next}} = 1$) or absence ($I_{\text{next}} = 0$) in our study.

In summary, each of the two factors has two levels, resulting in a total of four conditions. We treat the conditions as within-subjects variables. A detailed procedure is described in Section 3.4.

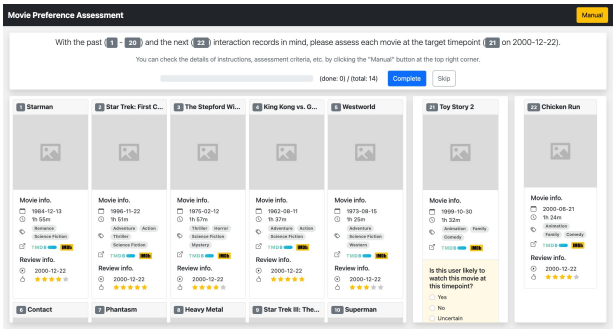

Figure 1: Main page of our assessment interface.

3.2 Interface

Figure 1 shows the main page of our assessment interface (translated for ease of explanation). The header area presents the instruction and progress of the assessment task. The main area below the header consists of three parts: the left part presents the past items ordered by their interaction timestamps; the right presents the next item (or nothing when $I_{\text{next}} = 0$); and the middle presents candidate items to be assessed. The candidate items are shuffled before the presentation to eliminate confounding due to order effects.

Each item is presented in a card style consisting of several sections. The first section summarizes the metadata of the movie (e.g., title, release date, and genres). We also include links to the corresponding pages on IMDb[3] and TMDB[4], if any, so that assessors unfamiliar with the movie can check its details. The past and next items have an additional section summarizing interactions between each item and the target user. Specifically, we include the timestamp and score of the user's rating, in the hope that such additional information may help assessors estimate the user's preference.

Candidate items in the middle part has another section asking two questions about preference assessments. **(1) Potential relevance.** This is the primary question asking *whether the target user is likely to watch this candidate movie at the target time point.* Annotators are instructed to take the past items (and the next item if shown) into account when answering this question. There are three choices for this question: "Yes" (referred to as `Relevant` hereinafter), "No" (`Irrelevant`), and "Uncertain" (`Neutral`). **(2) Knowledge.** This is the secondary question asking *whether the assessor has watched this candidate movie.* Response alternatives are "Yes," "No, but I have heard of it," and "No, and I have never heard of it." While we focus only on the primary question in this work, our released resources[1] includes responses to the secondary one for future work.

3.3 Quality control

As described earlier, we need to automatically measure the assessment quality without communicating with users. To this end, we leverage the interaction history of each user. Specifically, we add the following two types of items to candidate items. **(1) Positive item.** This is the item with which the target user actually interacted at the target time point. As this is used as the sole correct label in the conventional offline evaluation, assessors should judge it as `Relevant`. **(2) Negative items.** These are N_{neg} items randomly sampled from all but past, positive, and next items. As the target user has not interacted with these items, assessors should judge them as `Irrelevant`. In this work, we empirically set $N_{\text{neg}} = 2$.

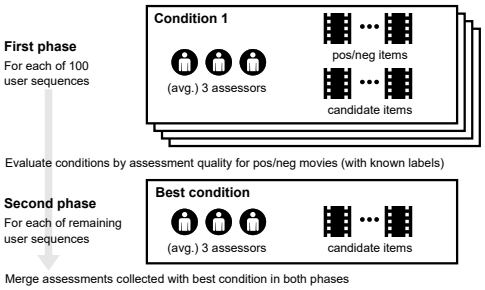

Figure 2: Two-phased assessment procedure.

Among the four conditions of our interface (Section 3.1), we can determine the best one by using the `Relevant` ratio for the positive item and the `Irrelevant` ratio for the negative items.

3.4 Procedure

As shown in Figure 2, our assessment framework follows a two-phased procedure: the first phase identifies the interface condition achieving the highest assessment quality while the second phase uses the best condition to collect large-scale preference data. In what follows, we first describe the procedure common to both phases and then detail the procedure of each phase.

Common procedure. For each ⟨user, condition⟩ pair, we constructed an assessment task, referred to as *a single task* hereinafter. We recruited assessors via Lancers[6], one of the most commonly used crowdsourcing platforms in Japan. The common workflow is as follows. (1) When assessors participated in single tasks for the first time, they were shown a consent document about the use of the collected data for research purposes. Only those that agreed to it proceeded to the main assessment page. (2) After first landing on this page, assessors were asked to read a guideline document containing instructions, assessment criteria, and interface usage. (3) Assessors judged each of candidate (and positive/negative) items by referencing to the contextual information depending on the condition. We allowed them to skip the current single task when necessary. (4) We paid JPY 45 (about USD 0.3) to each assessor on completion of a single task.

To enable assessors to participate in single tasks multiple times (for scalability), we ensured that each assessor was assigned to users with which he/she had not worked. On average, each single task was completed by three assessors. The mean completion time of a single task was around four minutes.

In the **first phase** of the procedure, we collected the assessments for 100 users in the dataset by using all of the four interface conditions. We then determined the best condition in terms of assessment quality for these users (Section 4.1). In the **second phase**, we collected the assessments for the remaining 5940 users in the dataset by using the best interface condition. As the target users did not overlap between the two phases, assessors were able to participate in single tasks in both phases. Finally, we merged the assessment data collected with the best condition in the first and second phases as our resource on additional preferences.

4 EXPERIMENTS ON ML-1M++ PREFERENCES

We conducted two experiments using the collected preferences.

[6] https://www.lancers.co.jp/

Table 1: Quality and agreement of assessments.

Condition		Quality		Agreement	
N_{past}	I_{next}	% Rel. / Pos.	% Irrel. / Neg.	% Agree.	Overlap
10	1	**0.557**	0.632	0.597	0.409
10	0	0.549	0.640	**0.612**	**0.415**
20	1	0.523	0.645	0.590	0.387
20	0	0.556	**0.668**	0.608	0.412

4.1 Comparison Among Conditions

First, we compared the four conditions to determine the best one.

Quality. We measured the quality of the assessments for each condition by using the metrics described in Section 3.3. Table 1 summarizes the mean ratios of the `Relevant` label for the positive item and `Irrelevant` label for the negative items.

The mean `Relevant` ratio for the positive item was larger than 0.5 for all conditions. While the conditions $(N_{\text{past}}, I_{\text{next}}) = (10, 1), (20, 0)$ achieved the highest and second highest scores, respectively, their difference was marginal. For this metric, we could not find any consistent trend with respect to individual factors. The mean `Irrelevant` ratio for the negative items was larger than 0.6, suggesting that judging false positives seemed to be easier than judging true positives. The best score 0.668 achieved by the condition $(20, 0)$ indicates that on average two of three assessors assigned to this condition for each single task detected the negative items correctly. For this metric, we can observe a consistent (and expected) effect on the N_{past} factor: showing more past items leaded to higher metric scores.

Agreement. The consistency among assessments is as important as the assessment quality for reliable data collection. We thus examined the degree of assessment consistency for each condition by using two agreement metrics widely adopted in the literature [1, 3, 21, 32]: **(1) Percentage agreement**, which counts the items receiving the same judgements from assessors and divides that number by the total number of the items assessed; **(2) Overlap**, which is the size of the intersection of the relevant item sets divided by the size of the union of the same sets. For simplicity, we binarized the relevance labels by treating both `Irrelevant` and `Neutral` as non-relevant and computed agreement scores between each assessor pair. Table 1 summarizes the result.

Overall, all conditions achieved similar degrees of agreement for each metric. The conditions $(N_{\text{past}}, I_{\text{next}}) = (10, 0), (20, 0)$ were always ranked in the top two, suggesting that contrary to our expectation, showing the next item (i.e., $I_{\text{next}} = 1$) had no positive effect on the agreement. Compared with the literature, the observed agreement scores are slightly lower. For example, the overlap scores measured for TREC-4 relevance assessments [32] were reported to be 0.426 (our best: 0.415). In the context of recommendation, Lu et al. [21] reported a percentage agreement score of 0.678 (our best: 0.612). Note that their assessments were collected via a laboratory study. Given the limited controllability of crowdsourcing, we believe our data has a satisfactory degree of assessment agreement.

Summary. Among the four, the condition $(N_{\text{past}}, I_{\text{next}}) = (20, 0)$ achieved the most promising and stable results: it was always ranked in the top two for both the assessments quality and agreement. Therefore, we regarded it as the best condition, with which we collected additional assessments for all users. On average, each assessor using this condition judged 8.1 candidate items as `Relevant`, 1.0 as `Neutral`, and 5.4 as `Irrelevant` for each target user.

Figure 3: Rankings based on different preference data.

4.2 Comparison Among Evaluation Outcomes

We next investigated how the outcome of offline evaluation could change by using either the original preferences (i.e., users' next items) or external ones (crowd workers' assessments). For the latter, we used simple aggregation methods to construct two ground truths: *lenient* and *strict* sets, which regard a candidate item as relevant if more than two and three assessors judge it as `Relevant`, respectively. As there was more than one relevant item in this setting, we adopted Normalized Discounted Cumulative Gain [13] with a cutoff of 10 (NDCG@10) as the evaluation measure.

Following the motivating example in Section 1, we compared the numbers of cases where a recommender pair obtained the same evaluation score for a testing example. Remarkably, the mean tie ratio for the original preferences was 70%, indicating that the conventional evaluation relying on the single ground truth cannot distinguish the difference of two rankings more than half the time. When using the external preferences instead, the ratio dramatically improved to 13% for the strict set and 3% for the lenient set.

We also compared the rankings of the recommenders on the basis of their mean performance. As shown in Figure 3, the overall trend looks similar (Kendall's τ was 0.93 for the $\langle$original, strict$\rangle$ pair and 0.79 for $\langle$original, lenient$\rangle$, with both $p < 0.01$). However, we can also observe several fluctuations at near positions. Similar observations by Lu et al. [21] suggest that our evaluation based on the external assessments may be a better proxy for users' true preferences, calling for further exploration in this direction.

5 CONCLUSIONS AND FUTURE DIRECTIONS

We have introduced ML-1M++, large-scale external assessments enriching the preferences of 6040 MovieLens users. Our contributions are as follows. **(1) Novelty.** To our knowledge, this is the first attempt to collect external preference assessments from crowd workers in the context of sequential recommendation. **(2) Availability.** We have released ML-1M++ as well as the code of our assessment interface, both of which can be used for research purposes.[1] **(3) Utility.** This paper details the process of obtaining the released resources, demonstrates the best assessment condition, and compares evaluation outcomes on the basis of the original and external preferences. **(4) Predicted Impact.** We believe external preference assessments play a pivotal role in establishing a more robust offline evaluation methodology for sequential recommendation. Interesting future directions enabled by our resources include developing new evaluation measures that leverage both the original and external preferences, better aggregating external assessments on the basis of the reliability and knowledge of assessors, exploring more accurate, scalable assessment frameworks, and collecting external assessments on other domains to study the generalizability.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number JP20K19935.

REFERENCES

[1] Omar Alonso and Stefano Mizzaro. 2012. Using crowdsourcing for TREC relevance assessment. *Information Processing & Management* 48, 6 (2012), 1053–1066. https://doi.org/10.1016/j.ipm.2012.01.004

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*. http://arxiv.org/abs/1409.0473

[3] Ben Carterette, Paul N. Bennett, David Maxwell Chickering, and Susan T. Dumais. 2008. Here or There: Preference Judgments for Relevance. In *ECIR*. 16–27.

[4] Oscar Celma. 2010. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space* (1st ed.). Springer. https://link.springer.com/book/10.1007/978-3-642-13287-2

[5] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and Mobility: User Movement in Location-Based Social Networks. In *KDD*. 1082–1090. https://doi.org/10.1145/2020408.2020579

[6] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep Learning for Sequential Recommendation: Algorithms, Influential Factors, and Evaluations. *ACM Transactions on Information Systems* 39, 1, Article 10 (2020). https://doi.org/10.1145/3426723

[7] Lei Guo, Hongzhi Yin, Qinyong Wang, Tong Chen, Alexander Zhou, and Nguyen Quoc Viet Hung. 2019. Streaming Session-Based Recommendation. In *KDD*. 1569–1577. https://doi.org/10.1145/3292500.3330839

[8] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4, Article 19 (2015). https://doi.org/10.1145/2827872

[9] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-Based Recommendation. In *RecSys*. 161–169. https://doi.org/10.1145/3109859.3109882

[10] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *WWW*. 507–517. https://doi.org/10.1145/2872427.2883037

[11] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel Recurrent Neural Network Architectures for Feature-Rich Session-Based Recommendations. In *RecSys*. 241–248. https://doi.org/10.1145/2959100.2959167

[12] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In *SIGIR*. 505–514. https://doi.org/10.1145/3209978.3210017

[13] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446. https://doi.org/10.1145/582415.582418

[14] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*. 197–206. https://doi.org/10.1109/ICDM.2018.00035

[15] LeCun Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324. https://doi.org/10.1109/5.726791

[16] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-Based Recommendation. In *CIKM*. 1419–1428. https://doi.org/10.1145/3132847.3132926

[17] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *WSDM*. 322–330. https://doi.org/10.1145/3336191.3371786

[18] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-Aware Sequential Location Recommendation. In *KDD*. 2009–2019. https://doi.org/10.1145/3394486.3403252

[19] Jing Lin, Weike Pan, and Zhong Ming. 2020. FISSA: Fusing Item Similarity Models with Self-Attention Networks for Sequential Recommendation. In *RecSys*. 130–139. https://doi.org/10.1145/3383313.3412247

[20] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-Based Recommendation. In *KDD*. 1831–1839. https://doi.org/10.1145/3219819.3219950

[21] Hongyu Lu, Weizhi Ma, Min Zhang, Maarten de Rijke, Yiqun Liu, and Shaoping Ma. 2021. Standing in Your Shoes: External Assessments for Personalized Recommender Systems. In *SIGIR*. 1523–1533. https://doi.org/10.1145/3404835.3462916

[22] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *ICDM*. 502–511. https://doi.org/10.1109/ICDM.2008.16

[23] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. RepeatNet: A Repeat Aware Neural Recommendation Machine for Session-Based Recommendation. In *AAAI*. 4806–4813. https://doi.org/10.1609/aaai.v33i01.33014806

[24] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-Basket Recommendation. In *WWW*. 811–820. https://doi.org/10.1145/1772690.1772773

[25] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. *Learning Internal Representations by Error Propagation*. MIT Press, Chapter 8, 318–362.

[26] Tetsuya Sakai, Douglas W. Oard, and Noriko Kando. 2020. *Evaluating Information Retrieval and Access Tasks: NTCIR's Legacy of Research Impact*. Springer. https://link.springer.com/book/10.1007/978-981-15-5554-1

[27] Markus Schedl. 2016. The LFM-1b Dataset for Music Retrieval and Recommendation. In *ICMR*. 103–110. https://doi.org/10.1145/2911996.2912004

[28] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *CIKM*. 1441–1450. https://doi.org/10.1145/3357384.3357895

[29] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-Based Recommendations. In *DLRS*. 17–22. https://doi.org/10.1145/2988450.2988452

[30] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM*. 565–573. https://doi.org/10.1145/3159652.3159656

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *NIPS*. 6000–6010.

[32] Ellen M. Voorhees. 2000. Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing & Management* 36, 5 (2000), 697–716. https://doi.org/10.1016/S0306-4573(00)00010-8

[33] Ellen M. Voorhees and Dawn M. Tice. 1999. The TREC-8 Question Answering Track Evaluation. In *TREC*. 83–105.

[34] Pengfei Wang, Yu Fan, Long Xia, Wayne Xin Zhao, Shaozhang Niu, and Jimmy Huang. 2020. KERL: A Knowledge-Guided Reinforcement Learning Model for Sequential Recommendation. In *SIGIR*. 209–218. https://doi.org/10.1145/3397271.3401134

[35] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. 2021. A Survey on Session-Based Recommender Systems. *Comput. Surveys* 54, 7, Article 154 (2021). https://doi.org/10.1145/3465401

[36] Zhenlei Wang, Jingsen Zhang, Hongteng Xu, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Counterfactual Data-Augmented Sequential Recommendation. In *SIGIR*. 347–356. https://doi.org/10.1145/3404835.3462855

[37] Dingqi Yang, Daqing Zhang, Vincent W. Zheng, and Zhiyong Yu. 2015. Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2015), 129–142. https://doi.org/10.1109/TSMC.2014.2327053

[38] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In *WSDM*. 582–590. https://doi.org/10.1145/3289600.3290975

[39] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-Aware Point-of-Interest Recommendation. In *SIGIR*. 363–372. https://doi.org/10.1145/2484028.2484030

[40] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-Level Deeper Self-Attention Network for Sequential Recommendation. In *IJCAI*. 4320–4326. https://doi.org/10.24963/ijcai.2019/600

[41] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *CIKM*. 4653–4664. https://doi.org/10.1145/3459637.3482016