End-to-End User and Item Embeddings: Specializing Retrieval Representations for Ranking

Simon Rauch
Amazon Music
Berlin, Germany
simonml@amazon.de

Vito Bellini
Amazon Music
Berlin, Germany
vitob@amazon.de

Anton Thielmann
Amazon Music
Berlin, Germany
thielmaf@amazon.de

Adrian Gruszczynski
Amazon Music
Berlin, Germany
adrigru@amazon.de

Giuseppe Di Benedetto
Amazon Music
Berlin, Germany
bgiusep@amazon.de

Abstract

Industrial recommender systems typically operate in two stages: retrieving a candidate set from a large catalog, then ranking those candidates using contextual information. The ranking stage relies on features that summarize a user's prior interactions with the system. These features are often carefully hand-crafted, and designing and maintaining them is time-consuming and computationally expensive. In this paper we propose leveraging the user and item embeddings already produced in the retrieval stage as input features to the downstream ranking model, and fine-tuning them for the ranking objective. We motivate this direction with A/B test results from a large-scale music streaming platform, where retrieval-stage user embeddings improve ranking performance even without fine-tuning. Our results suggest that representations learned for retrieval can reduce the reliance on expensive hand-crafted ranking features, and that fine-tuning them offers a promising path to further gains.

Keywords

recommender systems, music recommendation, learning to rank, sequential recommendation, user representations, transfer learning, off-policy evaluation, industrial recommender systems

ACM Reference Format:
Simon Rauch, Vito Bellini, Anton Thielmann, Adrian Gruszczynski, and Giuseppe Di Benedetto. 2026. End-to-End User and Item Embeddings: Specializing Retrieval Representations for Ranking. In *20th ACM Conference on Recommender Systems (RecSys '26), September 27-October 02, 2026, Minneapolis, MN, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3773078.3841256

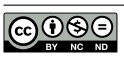
1 Introduction

Industrial recommender systems are typically organized in two stages [3]: a retrieval stage that selects a small set of candidate items from a large catalog, and a ranking stage that orders those candidates using richer contextual signals. The ranking model usually consumes features that summarize each user's prior interactions with the system, such as user-item affinity scores and counts of past interactions (e.g., like, skip, and follow signals in music streaming platforms). These features are effective, but they are largely hand-crafted [19]: designing them requires substantial engineering effort, and keeping them fresh is computationally expensive, since they must be recomputed and stored as user behavior evolves. At the same time, the retrieval stage is increasingly powered by sequential models that already learn item embeddings and a user embedding summarizing a user's interaction history [7, 13, 15]. This raises the natural question whether the representation learned for retrieval can be reused to inform ranking, reducing the reliance on expensive hand-crafted features. Reuse is appealing because user and item embeddings from the sequential recommender are already computed in the pipeline, with user representations that can be refreshed at request time, offering a more responsive alternative to hand-engineered features. We explore this direction by taking the user and item embeddings produced by a sequential retrieval model and supplying them as input features to a production ranking model. We report A/B test results from a large-scale music streaming platform showing that leveraging embeddings improves ranking performance even without fine-tuning them for the ranking task. Motivated by this result, we argue that the more promising direction is to fine-tune the retrieval embeddings end-to-end for the ranking objective, so that a single learned representation can progressively replace hand-crafted affinity and count features.

Our contributions are: **(i)** we propose reusing retrieval-stage user and item embeddings as ranking features, and fine-tuning them for the ranking task as a path toward retiring expensive hand-crafted features; **(ii)** we provide online A/B evidence that even non-fine-tuned user embeddings improve a production ranking model; and **(iii)** we outline the open questions this raises for end-to-end training across retrieval and ranking.

2 Method

We consider a standard two-stage recommender. The retrieval stage selects a candidate set from the catalog, and the ranking stage scores each candidate using user, item, and contextual features. As a baseline, we take a deep neural ranking model [2, 8, 20] whose inputs include hand-crafted features that summarize the user's history, such as user-item affinity scores and count-based features that can depend on the number of past interactions of a given type. These are the features we aim to complement and, ultimately, replace.



RecSys '26, Minneapolis, MN, USA

ACM ISBN 979-8-4007-2284-4/2026/09
https://doi.org/10.1145/3773078.3841256

As a proxy for the embedding retrieval model we consider gSASRec [9, 14], a self-attentive sequential model which, given a user's chronologically ordered interaction sequence, encodes the user's history into an embedding that can be used to score items for retrieval. We treat this embedding, namely the model's summary of the user's state after their most recent interaction, as a learned user representation. This embedding is computed at request time, taking into account the user's latest interactions with the system, without a separate feature-computation pipeline. We extract this embedding and supply it as an additional input to the ranking model, concatenated with the existing features. In the configuration we evaluate online, gSASRec is trained for its retrieval objective and its parameters are frozen when its output is consumed by the ranker, so no ranking-specific signal flows back into the embedding. This lets us isolate the value of the representation itself, independent of any adaptation to the ranking task. The configuration is deliberately conservative, since the embeddings are optimized for retrieval rather than ranking; we therefore propose fine-tuning the retrieval embeddings for the downstream ranking objective by propagating the ranking loss into the embeddings, or into a lightweight adapter over them, so that a single learned representation can take over the role of the hand-crafted affinity and count features.

3 Related Work

The idea of learning a user representation once and reusing it across tasks is well established. DUPN [12] learns universal user embeddings shared across multiple e-commerce tasks; PeterRec [18] pretrains a representation on interaction sequences and adapts it to downstream objectives via parameter-efficient adapters; Spotify's generalized user representations [5] centralize representation learning so downstream models consume a shared embedding. In industrial ranking, TransAct [17] instead trains a dedicated real-time sequence module end-to-end on the ranking objective, and SRP4CTR [6] transfers sequential pretraining into a CTR model through a cross-attention bridge. Our proposal shares the multi-task ambition of the first group but differs in three ways. First, our shared backbone is a sequential retrieval model (gSASRec) which computes embeddings on the fly at request time, staying responsive to the user's most recent interactions, rather than being precomputed in batch [5, 12]. Second, we show the representation transfers as a plain, frozen input feature with no task-specific adaptation, in contrast to adapter- or bridge-based transfer [6, 18] and to in-ranker end-to-end training [17]. Third, our objective is to replace expensive hand-crafted features per task; we report online A/B evidence on ranking as a first instance, and envision fine-tuning the item and user embeddings learned at the retrieval stage into a family of task-specialized embeddings (e.g., ranking) as the longer-term direction.

4 Online Experiment

We ran an A/B test on a ranking use case in a music streaming platform. The only difference between control and treatment was the presence of the responsive user embedding. Successful playbacks (a play of at least 30s) rose by 0.50% and listening hours by 0.16% (both $p < 0.005$), indicating better ranking quality. Stratifying requests by listening-history length, the embedding lifts per-request reward in every cohort, so the gain is broad-based rather than segment-specific. Lifts are smallest for cold-start users and largest at medium history, consistent with the embedding adding sequential signal that accrues with playbacks. Notably, cold-start users with no history still gain (+0.38%): despite an empty sequence, gSASRec learns a useful prior for the no-history state rather than a constant.

5 Ablation Study

The online gain is additive on top of the engineered features; we next ask offline how much of them the embedding can replace. We target the user-item features also derived from listening history — genre, artist, and recency affinity — in a 2×2 design crossing the embedding with removal of this group, retraining a ranker per cell and scoring it on one week of held-out traffic via position-bias-corrected off-policy estimation [1, 4, 10, 11, 16]. Removing the features costs the no-embedding ranker 4.1% of reward but the embedding ranker only 0.8% (Table 1; significant, disjoint intervals). The embedding thus subsumes most, though not all, of these features' signal. Full retirement is not yet possible, but the embedding recovers most of their contribution despite seeing only listening history — not the explicit follows and likes the features also use — which makes it a strong source of user signal for ranking.

6 Discussion

Our result suggests that a sequential model trained for retrieval already captures user signal that a ranking model can exploit, even with no adaptation. This opens a broader direction: rather than maintaining separate hand-crafted feature pipelines per task, a single sequential model could serve as a shared backbone, fine-tuned into task-specialized embeddings for ranking and other downstream tasks. Several questions remain open. It is unclear how much of the hand-crafted feature set such an embedding can ultimately replace; our preliminary offline study, in which we drop a subset of these features in favor of the user embedding, is encouraging but partial. Further testing will reveal whether adding item embeddings can capture higher-order information beyond the existing features. The choice between full fine-tuning and lightweight adapters trades off accuracy against the cost of maintaining many task-specific variants, and on-the-fly generation must remain within the ranker's latency budget. Finally, sharing a backbone across tasks raises the risk of negative transfer, which a multi-task formulation would need to manage. We see these as the central challenges in turning the present result into a practical, maintainable alternative to hand-crafted features.

References

[1] Alexander Buchholz, Giuseppe Di Benedetto, Jan Malte Lichtenberg, Yannik Stein, Vito Bellini, Ben London, and Thorsten Joachims. 2024. Off-Policy Evaluation for Learning-to-Rank via Interpolating the Item-Position Model and the Position-Based Model. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*. Association for Computing Machinery, New York, NY, USA.

[2] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[3] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. Association for Computing Machinery, New York, NY, USA.

Table 1: Estimated policy value (position-bias-corrected expected successful playbacks) from the position-based estimator, with 95% CIs. Engineered user-item features refer to genre affinity, artist affinity, and recency affinity (the user's affinity to recently released songs). Removing these from the ranker leads to a lower drop in offline performance when the user embedding is present (−0.8% vs −4.1% in reward). This suggests that the dense user embedding enables the ranker to learn relevant user-item features that overlap with our engineered features.

Condition	Policy value	95% CI
Ranker with dense user embedding		
all features	1.101	[1.100, 1.103]
w/o engineered user-item features	1.092	[1.085, 1.099]
Control ranker (no user embedding)		
all features	1.083	[1.081, 1.085]
w/o engineered user-item features	1.039	[1.030, 1.048]

[4] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601* (2011).

[5] Ghazal Fazelnia, Sanket Gupta, Claire Keum, Mark Koh, Ian Anderson, and Mounia Lalmas. 2024. Generalized User Representations for Transfer Learning. *CoRR* abs/2403.00584 (2024). doi:10.48550/arXiv.2403.00584

[6] Ruidong Han, Qianzhong Li, He Jiang, Rui Li, Yurou Zhao, Xiang Li, and Wei Lin. 2024. Enhancing CTR Prediction through Sequential Recommendation Pre-training: Introducing the SRP4CTR Framework. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*. 3777–3781. doi:10.1145/3627673.3679914

[7] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[8] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining (WSDM '17)*. Association for Computing Machinery, New York, NY, USA.

[9] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. In *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM '18)*. IEEE.

[10] Shuai Li, Yasin Abbasi-Yadkori, Branislav Kveton, Shan Muthukrishnan, Vishwa Vinay, and Zheng Wen. 2018. Offline evaluation of ranking policies with click models. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1685–1694.

[11] Ben London, Alexander Buchholz, Giuseppe Di Benedetto, Jan Malte Lichtenberg, Yannik Stein, and Thorsten Joachims. 2023. Self-Normalized Off-Policy Estimators for Ranking. In *RecSys 2023 Workshop on Causality, Counterfactuals & Sequential Decision-Making (CONSEQUENCES)*.

[12] Yabo Ni, Dan Ou, Shichen Liu, Xiang Li, Wenwu Ou, Anxiang Zeng, and Luo Si. 2018. Perceive Your Users in Depth: Learning Universal User Representations from Multiple E-commerce Tasks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 596–605. doi:10.1145/3219819.3219828

[13] Nikil Pancha, Andrew Zhai, Jure Leskovec, and Charles Rosenberg. 2022. PinnerFormer: Sequence Modeling for User Representation at Pinterest. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. Association for Computing Machinery, New York, NY, USA.

[14] Aleksandr V. Petrov and Craig Macdonald. 2023. gSASRec: Reducing Overconfidence in Sequential Recommendation Trained with Negative Sampling. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23)*. Association for Computing Machinery, New York, NY, USA.

[15] 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 *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[16] Adith Swaminathan and Thorsten Joachims. 2015. The self-normalized estimator for counterfactual learning. *advances in neural information processing systems* 28 (2015).

[17] Xue Xia, Pong Eksombatchai, Nikil Pancha, Dhruvil Deven Badani, Po-Wei Wang, Neng Gu, Saurabh Vishwas Joshi, Nazanin Farahpour, Zhiyuan Zhang, and Andrew Zhai. 2023. TransAct: Transformer-based Realtime User Action Model for Recommendation at Pinterest. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*. Association for Computing Machinery, New York, NY, USA.

[18] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-Efficient Transfer from Sequential Behaviors for User Modeling and Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. Association for Computing Machinery, New York, NY, USA.

[19] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, Yinghai Lu, and Yu Shi. 2024. Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. In *Proceedings of the 41st International Conference on Machine Learning (ICML '24, Vol. 235)*. PMLR.

[20] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.