Leveraging Large Language Models for Sequential Recommendation

Jesse Harte
Delivery Hero Research
Berlin, Germany
Delft University of Technology
Delft, The Netherlands

Wouter Zorgdrager
Delivery Hero Research
Berlin, Germany

Panos Louridas
Athens University of Economics & Business
Athens, Greece

Asterios Katsifodimos
Delft University of Technology
Delft, The Netherlands

Dietmar Jannach
University of Klagenfurt
Klagenfurt, Austria

Marios Fragkoulis
Delivery Hero Research
Berlin, Germany

ABSTRACT

Sequential recommendation problems have received increasing attention in research during the past few years, leading to the inception of a large variety of algorithmic approaches. In this work, we explore how large language models (LLMs), which are nowadays introducing disruptive effects in many AI-based applications, can be used to build or improve sequential recommendation approaches. Specifically, we devise and evaluate three approaches to leverage the power of LLMs in different ways. Our results from experiments on two datasets show that initializing the state-of-the-art sequential recommendation model BERT4Rec with embeddings obtained from an LLM improves NDCG by 15-20% compared to the vanilla BERT4Rec model. Furthermore, we find that a simple approach that leverages LLM embeddings for producing recommendations, can provide competitive performance by highlighting semantically related items. We publicly share the code and data of our experiments to ensure reproducibility.[1]

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Recommender Systems, Large Language Models, Sequential Recommendation, Evaluation

ACM Reference Format:
Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging Large Language Models for Sequential Recommendation. In *Seventeenth ACM Conference on Recommender Systems (RecSys '23), September 18–22, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3604915.3610639

[1] https://github.com/dh-r/LLM-Sequential-Recommendation

RecSys '23, September 18–22, 2023, Singapore, Singapore
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0241-9/23/09.
https://doi.org/10.1145/3604915.3610639

1 INTRODUCTION

Sequential recommendation problems have received increased interest recently [31, 38]. In contrast to the traditional, sequence-agnostic matrix-completion setup [33], the problem in sequential recommendation is to predict the next user interest or action, given a sequence of past user interactions. Practical applications of sequential recommendation include next-purchase prediction, next-track music recommendation, or next Point-of-Interest suggestions for tourism. Due to their high practical relevance, a multitude of algorithmic approaches have been proposed in the past few years [14, 21, 30, 35], including approaches that utilize side information about the items, such as an item's category [23, 41].

From a technical perspective, the sequential recommendation problem shares similarities with the next word prediction problem [7, 32]. Under this light, we can observe a parallel between research in Natural Language Processing (NLP) and sequential recommendation, where novel recommendation models are inspired by NLP models [6]. GRU4Rec [14] adopted the Gated Recurrent Unit (GRU) mechanism from [5], SASRec [21] used the transformer architecture from [37], and BERT4Rec [35] adopted BERT [7]. The influence of NLP research to sequential recommendation models extends naturally to Large Language Models (LLMs). LLMs, in particular ones based on Generative Pretrained Transformers [32], are exhibiting disruptive effects in various AI-based applications with their semantically rich and meaningful responses.

However, limited research exists so far on leveraging the inherent semantic information of LLMs, which the abovementioned approaches lack, for sequential recommendation problems. A number of recent works in fact started to explore the potential of relying on LLMs for recommendation tasks; see [27, 40] for recent surveys. Here, we extend this line of research for sequential recommendation problems, providing the following contributions and insights.

- We devise three orthogonal methods of leveraging LLMs for sequential recommendation. In our first approach (LLMSeqSim), we retrieve a semantically-rich embedding from an existing LLM (from OpenAI) for each item in a session. We then compute an aggregate session embedding to recommend catalog products with a similar embedding. In the second approach (LLMSeqPrompt), we fine-tune an LLM with dataset-specific information in the form of prompt-completion pairs and ask the model to produce next item recommendations for test prompts. Finally, our third approach (LLM2BERT4Rec) consists of initializing existing sequential models with item embeddings obtained from an LLM.

- Experiments on two datasets, including a real-world dataset from Delivery Hero, reveal that initializing a sequential model with LLM embeddings is particularly effective: applying it to the state-of-the-art model BERT4Rec improves accuracy in terms of NDCG by 15-20%, making it the best-performing model in our experiments.
- Finally, we find that in certain applications simply using LLM embeddings to find suitable items for a given session (LLMSeqSim) can lead to state-of-the-art performance.

2 BACKGROUND & RELATED WORK

The recent developments in LLMs have taken the world by surprise. Models like OpenAI GPT [4], Google BERT [7], and Facebook LLaMA [36], which employ deep transformer architectures, demonstrate how innovations in NLP can reshape mainstream online activities, such as search, shopping, and customer care. Inevitably, research in recommender systems is significantly impacted by the developments in the area of LLMs as well. According to recent surveys [27, 40], LLMs are mainly utilized for recommendation problems in two ways: by providing embeddings that can be used to initialize existing recommendation models [29, 39, 43], and by producing recommendations leveraging their inherent knowledge encoding [2, 13, 22]. LLMs as recommendation models can provide recommendations given *a)* only a task specification (zero-shot), *b)* a few examples given inline to the prompt of a task (few-shot), or *c)* after fine-tuning the model's weights for a task given a set of training examples [4]. This incremental training process deviates from typical recommendation models, which have to be trained from zero on domain data. In fact, LLMs show early indications of adaptability to different recommendation domains with modest fine-tuning [15, 16]. Finally, LLMs have been applied in various recommendation tasks, such as rating prediction [25], item generation [26], and reranking [17] across domains [29, 39].

In this work we explore the potential of using LLMs for sequential recommendation problems [20]. In short, in sequential recommendation problems, we consider as input a sequence of user interactions $S^u = (S^u_1, S^u_2, ..., S^u_n)$, for user u, where n is the length of the sequence and S^u_i are individual items. The aim is to predict the next interaction of the given sequence. Besides the recent sequential recommendation models mentioned in the introduction [14, 21, 35], in earlier works, the sequential recommendation problem has been modelled as a Markov Chain [9] or a Markov Decision Process [34]. Neighborhood-based approaches, such as SKNN [19], have also been proposed.

Early research work regarding LLMs for sequential recommendation problems has shown mixed results [8, 11, 17, 28, 44]. The very recent VQ-Rec model [15] employs a transformer architecture and applies a novel representation scheme to embeddings retrieved from BERT in order to adapt to new domains. VQ-Rec outperforms a number of sequential recommendation models across datasets of different domains, and it has been shown that SASRec with LLM embeddings is better than the original SASRec method for half of the datasets representing different domains. Finally, in an upcoming work [42], SASRec with LLM embeddings is shown to improve over SASRec. The recent approaches presented in [15] and [42] differ from our work in particular in terms of the goals they pursue. VQ-Rec [15] targets cross-domain recommendations with a novel item representation scheme, while [42] evaluates whether recommendation models leveraging different modalities perform better than existing recommendation models that rely on item identifiers.

The work presented in this paper complements these recent lines of research and proposes and evaluates three alternative ways of leveraging LLMs for sequential recommendation. Differently from earlier approaches, our work shows that initializing an existing sequential model with LLM-based embeddings is highly effective and helps to outperform existing state-of-the-art models. In addition, we find that retrieving relevant items solely based on LLM embedding similarity can lead to compelling recommendations depending on the dataset.

3 THREE LLM-BASED APPROACHES FOR SEQUENTIAL RECOMMENDATIONS

In this section, we describe the three technical approaches sketched in Section 1.

3.1 LLMSeqSim: Recommending Semantically Related Items via LLM Embeddings

With this first approach, our goal is to explore if recommendations can benefit from a holistic notion of similarity provided by LLMs. To achieve this, we leverage *LLM embeddings* to produce recommendations in three steps. First, we query the `text-embedding-ada-002`[2] OpenAI embedding model with the names of the products in the item catalog and retrieve their embeddings. Second, we compute a session embedding for each session in our test set by combining the embeddings of the individual products in the session. Here, we try different combination strategies: *a)* the average of the product embeddings, *b)* a weighted average using linear and exponential decay functions depending on the position of the item in the session, and *c)* only the embedding of the last product.[3] Third, we compare the session embedding to the embeddings of the items in the product catalog using cosine, Euclidean, and dot product similarity.[4] Finally, we recommend the top-k products from the catalog with the highest embedding similarity to the session embedding.

3.2 LLMSeqPrompt: Prompt-based Recommendations by a Fine-Tuned LLM

In this approach, we inject domain knowledge to the collective information that a base LLM incorporates, with the goal of increasing the quality of the recommendations by an LLM that is given information about an ongoing session in the form of a prompt. To this end, we fine-tune an OpenAI `ada` model on training samples consisting of a prompt (the input) and a completion (the intended output). In our case, the prompt is a session, which contains a list of product names except for the last product, and the completion is the name of the last product in the same session, see Figure 1.

[2] https://platform.openai.com/docs/guides/embeddings/second-generation-models

[3] We also tried to create an aggregated session embedding by concatenating the plain product names and then querying the Open AI embeddings API. This however led to worse results.

[4] The choice of the similarity measure did not significantly impact the results.

To optimize performance, we fine-tune the model until the validation loss converges. After training, we provide the prompts of the sessions in the test set to the fine-tuned model to obtain recommendations. We note that we make no strong assumption regarding the order of the returned recommendations. Therefore, we use the tendency of the model to provide duplicate recommendations as a proxy of its confidence and rank the recommendations by frequency of appearance. Then, to create a full slate of unique recommendations, we retrieve the embedding of each duplicate product using the OpenAI embeddings API and take the catalog's product that is closest in terms of embedding similarity using the dot product measure. Finally, we note that the fine-tuned LLM, being a generative model, may also return hallucinated products, which we map to catalog products using the same method as for duplicate products.

3.3 LLM2BERT4Rec: Recommending with an LLM-enhanced Sequential Model

In our third approach, our goal is to leverage the semantically-rich item representations provided by an LLM to enhance an existing sequential recommendation model. Specifically, in our work we focus on BERT4Rec [35], a state-of-the-art transformer-based model, which employs the transformer architecture [37] of BERT [7].

BERT's transformer architecture consists of an embedding layer, a stack of encoder layers, and a projection head. Furthermore, BERT features a masked language model training protocol, which involves masking items at random positions and letting the model predict their true identity. Initially, the embedding layer embeds an input sequence of (potentially masked) item IDs into a sequence of embeddings using both the item ID and the item position. Then the transformer encoder layers process the embedding sequence using a multi-head attention module and a feed-forward network shared across all positions. Finally, the projection head projects the embeddings at each masked position to a probability distribution in order to obtain the true identity of the masked item. The projection head reuses the item embeddings of the embedding layer to reduce the model's size and to avoid overfitting.

To allow BERT4Rec to leverage the rich information encoded in LLMs, we initialize BERT4Rec's item embeddings using the LLM embeddings described in Section 3.1. In order to align the embedding dimension of the LLM embeddings (1536) with the configured dimension of BERT4Rec's embedding layer (e.g., 64), we employ Principal Components Analysis (PCA) to get 64 principal components of the LLM embeddings, which we then use to initialize the item embeddings of BERT4Rec's embedding layer. Finally, we train the enhanced model the same way as our baseline BERT4Recmodel.

4 EXPERIMENTAL EVALUATION

In this section, we describe our experimental setup (Section 4.1) and the results of our empirical evaluation (Section 4.2).

4.1 Experimental setup

Datasets and Data Splitting. We use the public Amazon Beauty [12] dataset and a novel, real-world e-commerce dataset from Delivery Hero[5] for our experiments. The Beauty dataset contains product reviews and ratings from Amazon. In line with prior research [1], we pre-processed the dataset to include at least five interactions per user and item (p-core = 5). The Delivery Hero dataset contains anonymous QCommerce sessions for dark store and local shop orders. To better simulate a real-world setting, we did not preprocess this dataset, except that we removed sessions with only one interaction from the test set. QCommerce is a segment of e-Commerce focusing on fast delivery times on the last mile. Dataset statistics are given in Table 1. To create a train and test set in a sound way, we first split a dataset containing sessions temporally such that all test sessions succeed train sessions in time. Then in the test set, we adopt the leave-one-out approach followed by [21, 35] where all but the last interaction of each session represent the prompt, while the last interaction serves as the ground truth.

Metrics. We use the standard ranking accuracy metrics NDCG, MRR, and HitRate at the usual cut-off lengths of 10 and 20. Furthermore, we consider the following *beyond-accuracy* metrics to obtain a more comprehensive picture of the performance of the different algorithms: catalog coverage, serendipity, and novelty. *Catalog coverage* represents the fraction of catalog items that appeared in at least one top-n recommendation list of the users in the test set [18]. *Serendipity* measures the average number of correct recommendations for each user that are not recommended by a popularity baseline [10]. *Novelty* computes the negative log of the relative item popularity, or self-information [45].

Models. We include both session-based algorithms of different families, GRU4Rec [14], and SKNN [19], as well as two state-of-the-art sequential models, BERT4Rec [35] and SASRec [21]. We tested all variants of the SKNN nearest-neighbor method proposed in [30] and report the results in the online material. In addition, we include the three LLM-based approaches proposed in Section 3. Finally, we include a popularity-based baseline (MostPopular) in the experiments.

Hyperparameter Tuning. We systematically tuned all models (except the LLMSeqSim and the LLMSeqPrompt) on three validation folds with the Tree Parzen Estimator (TPE) sampler [3], and used the average NDCG@20 across the folds as the optimization goal. For LLMSeqPrompt, we applied manual hyperparameter search. The examined hyperparameter ranges and optimal values for each dataset are reported in the online material.

4.2 Results and Discussion

Table 2 and Table 3 show the results obtained for the Amazon Beauty and the Delivery Hero dataset on the hidden test set, respectively. We report the best results of 5 runs. The table is sorted according to NDCG@20.

Accuracy Results. The highest values in terms of NDCG@20 are obtained by **LLM2BERT4Rec** for both datasets. In both cases, the gains obtained by using LLM-based item embeddings are substantial, demonstrating the benefits of relying on semantically-rich embeddings in this sequential model. The NDCG value increased by more than 20% for Beauty and over 15% on the Delivery Hero

[5] https://www.deliveryhero.com

```
{"prompt":     "1. Burt's Bees Rosewater Toner 8oz\n
                2. Philosophy Lip Shine, Mimosa, 0.5 Ounce\n
                3. LaLicious Sugar Souffle Body Scrub 16 fl oz.\n
                4. Marc Jacobs Daisy Eau So Fresh Eau de Toilette Spray-125ml/4.25 oz.\n \n\n###\n\n",
 "completion": " Bcbg Max Azria Eau De Parfum Spray for Women, 3.4 Ounce ###"}
```

Figure 1: Example prompt and completion for fine-tuning from the Beauty dataset

Dataset	# sessions	# items	# interactions	Avg. length	Density
Beauty 5-core	22,363	12,101	198,502	8.9	0.073%
Delivery Hero	258,710	38,246	1,474,658	5.7	0.015%

Table 1: Dataset statistics

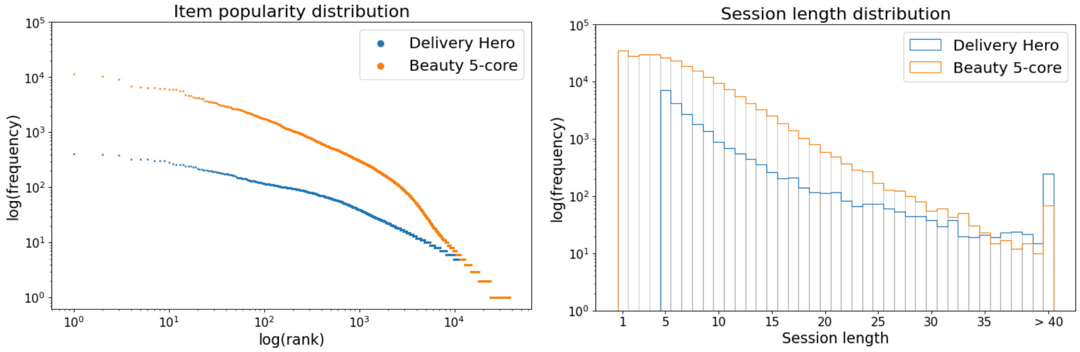

Figure 2: Distribution of items ranked by popularity (left) and histogram of session length (right) for the datasets

dataset.[6] To confirm that the semantics of the LLM embeddings is the driver of performance, we ran an experiment in which we permuted the item embeddings such that the embedding of each item is initialized to the principal components of the LLM embedding of another product from the catalogue. The experiment maintains the statistical properties of the embeddings, but deprives the item embeddings of the semantics of the LLM embeddings. The resulting model exhibited worse performance than the baseline BERT4Rec model with randomly-initialized item embeddings clearly showing that the performance improvement cannot be credited to the statistical properties of the embeddings.

The relative performance of **LLMSeqSim**, again considering NDCG values, varies across the two datasets. On the Beauty dataset, the model is highly competitive, with NDCG@20 values only being slightly lower than LLM2BERT4Rec. At shorter list lengths, i.e., at NDCG@10, the LLMSeqSim model even leads to the best performance for this dataset. Notably, the embedding combination strategy that led to the best results considered only the last item of the session (see Section 3.1). For the Delivery Hero dataset, in contrast, the picture is very different, and LLMSeqSim leads to quite poor performance, only outperforming the popularity-based baseline. We hypothesize that this phenomenon is a result of the quite different characteristics of the two datasets. For example, in Figure 2, we observe that many items in the real-world Delivery Hero dataset occur very infrequently. This may limit the capacity of LLMSeqSim to find similar items, given also the substantially broader item catalog in the Delivery Hero dataset. Furthermore, a manual inspection of a sample of test prompts, recommendations, and ground truths of the two datasets indicates that users in the Beauty dataset frequently rate items of a certain brand. Since brand names are part of the product names that are input to the LLM, recommending similar items may turn out to be particularly effective.

Looking at the other accuracy metrics (**Hit Rate** and **MRR**), we find that these are generally highly correlated with the NDCG results. A notable exception are the MRR values of the LLMSeqSim model and the V_SKNN approach on the Beauty dataset. While these two approaches lead to slightly inferior results at NDCG@20 and in particular also for HR@20, they are superior in terms of MRR. This means that these methods place the hidden target item higher up in the recommendation list in case the target item is included in the top 20. Similar observations regarding the good performance of some methods in terms of MRR on specific datasets were previously reported also in [30].

Interestingly, as also reported in [24, 30], **nearest-neighbor** approaches can be quite competitive depending on the dataset. On Beauty, V_SKNN outperforms all of the more sophisticated neural models (BERT4Rec, GRU4Rec, SASRec) in all accuracy metrics except Hit Rate@20. On the Delivery Hero dataset, in contrast, the neural models perform better in all accuracy metrics except MRR and NDCG@10. Further inspection (see online material) showed that

[6] We also examined the value of LLM embeddings for the SASRec model, where we observed marked increases in the NDCG, but not to the extent that it outperformed LLM2BERT4Rec. We report these additional results in the online material.

Model	Top@10						Top@20					
	nDCG	HR	MRR	CatCov	Seren	Novel	nDCG	HR	MRR	CatCov	Seren	Novel
LLM2BERT4Rec	0.041	**0.076**	0.030	0.180	**0.072**	11.688	**0.051**	**0.118**	0.033	0.260	**0.110**	11.888
LLMSeqSim	**0.044**	0.063	**0.038**	**0.763**	0.063	**13.819**	0.048	0.079	**0.039**	**0.889**	0.079	**13.858**
V_SKNN	0.041	0.071	0.033	0.673	0.069	12.241	0.047	0.095	0.034	0.889	0.091	12.492
BERT4Rec	0.034	0.067	0.024	0.231	0.064	12.293	0.043	0.103	0.027	0.312	0.098	12.423
GRU4Rec	0.027	0.051	0.020	0.145	0.047	11.409	0.035	0.082	0.022	0.214	0.074	11.597
SASRec	0.026	0.051	0.019	0.121	0.048	11.485	0.033	0.080	0.021	0.182	0.073	11.678
LLMSeqPrompt	0.025	0.045	0.019	0.500	0.044	13.001	0.030	0.064	0.020	0.688	0.063	13.361
MostPopular	0.005	0.010	0.003	0.001	0.001	9.187	0.006	0.018	0.003	0.002	0.001	9.408

Table 2: Evaluation results for the Amazon Beauty dataset

Model	Top@10						Top@20					
	nDCG	HR	MRR	CatCov	Seren	Novel	nDCG	HR	MRR	CatCov	Seren	Novel
LLM2BERT4Rec	**0.102**	**0.179**	**0.078**	0.245	**0.151**	10.864	**0.120**	**0.252**	**0.083**	0.311	**0.198**	11.050
BERT4Rec	0.088	0.157	0.067	0.325	0.128	10.821	0.104	0.221	0.071	0.429	0.165	11.032
GRU4Rec	0.085	0.153	0.064	0.127	0.124	10.570	0.101	0.218	0.068	0.172	0.161	10.823
SASRec	0.084	0.149	0.065	0.170	0.120	10.674	0.100	0.212	0.069	0.229	0.156	10.913
V_SKNN	0.087	0.148	0.068	0.381	0.120	10.444	0.100	0.200	0.072	0.452	0.146	10.602
LLMSeqPrompt	0.063	0.116	0.047	0.400	0.107	12.048	0.070	0.144	0.049	0.611	0.123	13.788
LLMSeqSim	0.039	0.069	0.029	**0.633**	0.069	**16.315**	0.046	0.096	0.031	**0.763**	0.093	**16.536**
MostPopular	0.024	0.049	0.017	0.000	0.000	7.518	0.032	0.079	0.019	0.001	0.000	7.836

Table 3: Evaluation results for the Delivery Hero dataset

SKNN's performance drops as the length of sessions increases, while the performance of the other models remains stable.

The performance of the **LLMSeqPrompt** model again depends on the dataset. On the Beauty dataset, it leads to accuracy values that are often only slightly lower than SASRec, which is typically considered a strong state-of-the-art baseline. On the Delivery Hero dataset, in contrast, the drop in performance compared to the other models is substantial. Still, LLMSeqPrompt leads to accuracy values that are markedly higher than the popularity baseline. Given its versatility, ease of configuration and promising performance, LLMSeqPrompt merits further research.

Beyond-Accuracy Results. We make the following observations for **coverage**, **serendipity** and **novelty**. The LLMSeqSim model consistently leads to the best coverage and novelty. This is not too surprising, given the nature of the approach, which is solely based on embeddings similarities. Unlike other methods that use collaborative signals, i.e., past user-item interactions, the general popularity of an item in terms of the amount of observed past interactions does not play role in LLMSeqSim, neither directly nor implicitly. Thus, the model has no tendency to concentrate the recommendations on a certain subset of (popular) items. We recall that the used novelty measure is based on the popularity of the items in the recommendations. The serendipity results are largely aligned with the accuracy measures across the datasets. This generally confirms the value of personalizing the recommendations to individual user preferences, compared to recommending mostly popular items to everyone. We iterate that our serendipity measure counts the fraction of correctly recommended items that would not be recommended by a popularity-based approach.

5 CONCLUSIONS

In this work, we devised and evaluated three approaches that leverage LLMs for sequential recommendation problems. A systematic empirical evaluation revealed that BERT4Rec initialized with LLM embeddings achieves the best performance for two datasets, and that the LLM-based initialization leads to a substantial improvement in accuracy. In our future work, we plan to investigate if our findings generalize to different domains, using alternative datasets with diverse characteristics. Furthermore, we will explore if using other LLMs, e.g., ones with different architectures and training corpora, will lead to similar performance gains, including a hybrid of LLM2BERT4Rec with LLMSeqSim towards combining their accuracy and beyond-accuracy performance. Finally, it is open so far if passing other types of information besides product names, e.g., category information, to an LLM can help to further improve the performance of the models.

REFERENCES

[1] Vito Walter Anelli, Alejandro Bellogín, Tommaso Di Noia, Dietmar Jannach, and Claudio Pomo. 2022. Top-N Recommendation Algorithms: A Quest for the State-of-the-Art. In *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization* (Barcelona, Spain) *(UMAP '22)*. Association for Computing Machinery, 121–131.

[2] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. arXiv:2305.00447 [cs.IR]

[3] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing*

Systems, Vol. 24. Curran Associates, Inc.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 1877–1901.

[5] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 1724–1734.

[6] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, and Even Oldridge. 2021. Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation. In *Proceedings of the 15th ACM Conference on Recommender Systems (RecSys '21)*. Association for Computing Machinery, 143–153.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 4171–4186.

[8] Hao Ding, Anoop Deoras, Yuyang (Bernie) Wang, and Hao Wang. 2022. Zero shot recommender systems. In *ICLR 2022 Workshop on Deep Generative Models for Highly Structured Data*.

[9] Florent Garcin, Christos Dimitrakakis, and Boi Faltings. 2013. Personalized News Recommendation with Context Trees. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. Association for Computing Machinery, 105–112.

[10] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. Association for Computing Machinery, 257–260.

[11] 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 *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys '22)*. Association for Computing Machinery, 299–315.

[12] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. International World Wide Web Conferences Steering Committee, 507–517.

[13] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. TabLLM: Few-shot Classification of Tabular Data with Large Language Models. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 206)*. PMLR, 5549–5581.

[14] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR*.

[15] Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. 2023. Learning Vector-Quantized Item Representation for Transferable Sequential Recommenders. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*. Association for Computing Machinery, 1162–1171.

[16] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards Universal Sequence Representation Learning for Recommender Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. Association for Computing Machinery, 585–593.

[17] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2023. Large Language Models are Zero-Shot Rankers for Recommender Systems. arXiv:2305.08845 [cs.IR]

[18] Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac. 2015. What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction* 25, 5 (2015), 427–491.

[19] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks Meet the Neighborhood for Session-Based Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. Association for Computing Machinery, 306–310.

[20] Dietmar Jannach, Bamshad Mobasher, and Shlomo Berkovsky. 2020. Research directions in session-based and sequential recommendation. *User Modeling and User-Adapted Interaction* 30, 4 (2020), 609–616.

[21] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *Proceedings of the* 18^{th} *IEEE International Conference on Data Mining (ICDM 2018)*. 197–206.

[22] Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed Chi, and Derek Zhiyuan Cheng. 2023. Do LLMs Understand User Preferences? Evaluating LLMs On User Rating Prediction. arXiv:2305.06474 [cs.IR]

[23] Siqi Lai, Erli Meng, Fan Zhang, Chenliang Li, Bin Wang, and Aixin Sun. 2022. An Attribute-Driven Mirror Graph Network for Session-based Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1674–1683.

[24] Sara Latifi, Dietmar Jannach, and Andrés Ferraro. 2022. Sequential Recommendation: A Study on Transformers, Nearest Neighbors and Sampled Metrics. *Information Sciences* 609 (2022), 660 – 678.

[25] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. In *KDD '23: The 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[26] Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni. 2023. GPT4Rec: A Generative Framework for Personalized Recommendation and User Interests Interpretation. arXiv:2304.03879 [cs.IR]

[27] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. 2023. How Can Recommender Systems Benefit from Large Language Models: A Survey. arXiv:2306.05817 [cs.IR]

[28] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is ChatGPT a Good Recommender? A Preliminary Study. arXiv:2304.10149 [cs.IR]

[29] Yiding Liu, Weixue Lu, Suqi Cheng, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained Language Model for Web-scale Retrieval in Baidu Search. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 3365–3375.

[30] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of Session-based Recommendation Algorithms. *User-Modeling and User-Adapted Interaction* 28, 4–5 (2018), 331–390.

[31] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *Comput. Surveys* 51, 4 (2018).

[32] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

[33] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*. 175–186.

[34] Guy Shani, David Heckerman, and Ronen I. Brafman. 2005. An MDP-Based Recommender System. *Journal of Machine Learning Research* 6, 43 (2005), 1265–1295.

[35] 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 (CIKM '19)*. 1441–1450.

[36] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc.

[38] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. 2021. A Survey on Session-Based Recommender Systems. *ACM Comput. Surv.* 54, 7 (2021).

[39] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering News Recommendation with Pre-Trained Language Models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. Association for Computing Machinery, 1652–1656.

[40] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. 2023. A Survey on Large Language Models for Recommendation. arXiv:2305.19860 [cs.IR]

[41] Hao Xu, Bo Yang, Xiangkun Liu, Wenqi Fan, and Qing Li. 2022. Category-aware Multi-relation Heterogeneous Graph Neural Networks for Session-based Recommendation. *Knowledge-Based Systems* 251 (2022).

[42] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to Go Next for Recommender Systems? ID- vs. Modality-based Recommender Models Revisited. In *SIGIR (To appear)*.

[43] Qi Zhang, Jingjie Li, Qinglin Jia, Chuyuan Wang, Jieming Zhu, Zhaowei Wang, and Xiuqiang He. 2021. UNBERT: User-News Matching BERT for News Recommendation. In *Proceedings of the Thirtieth International Joint Conference on*

Artificial Intelligence, IJCAI-21. International Joint Conferences on Artificial Intelligence Organization, 3356–3362.

[44] Yuhui Zhang, Hao Ding, Zeren Shui, Yifei Ma, James Zou, Anoop Deoras, and Hao Wang. 2021. Language models as recommender systems: Evaluations and limitations. In *NeurIPS 2021 Workshop on I (Still) Can't Believe It's Not Better*.

[45] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. 2010. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences* 107, 10 (2010), 4511–4515.