Large Language Model Personalization in Recommendation

Eric Anderson and **Prahitha Movva** and **Darsh Gondalia** and **Michael Yang**
College of Information and Computer Sciences
University of Massachusetts Amherst

Paul Bennett and **Praveen Ravichandran** and **Ghazal Fazelnia**
Spotify Research

Abstract

Large Language Models are effective at the task of sequential recommendation by leveraging text to provide users with more accurate recommendations. Traditionally, LLMs are trained on a broad range of users with varying characteristics and behaviors. We experiment with increasing recommendation performance through model personalization by pre-training, then fine-tuning a language model at the user-level. Further, we attempt to increase efficiency by grouping users into cohorts and examining the performance trade off between these techniques.

1 Introduction

1.1 Task Description

We investigate the spectrum of personalization in terms of sequential recommendation performance. We start by measuring the performance of a backbone model with no personalization, then adapt the backbone model to the recommendation task. We continue to increase levels of personalization by fine-tuning the backbone model on user-item interactions histories at both the cohort-level and user-level. The following research questions are of interest:

- **RQ1:** Does personalization in recommendation work better than not personalizing?
- **RQ2:** Does grouping users into cohorts by historical item similarity work better than random cohorts?
- **RQ3:** Does cohort personalization improve over the baseline?
- **RQ4:** Does per-user personalization improve over the baseline?
- **RQ5:** Does per-user personalization improve over cohort personalization?
- **RQ6:** Does per-user personalization improve as the frequency of user observations increase?

1.2 Motivation and Limitations of Existing Work

Early recommendation systems such as Matrix Factorization (He et al., 2017) and Markov Chain (Rendle et al., 2010) use basic user and item summary statistics with short histories to suggest related items to users. Later, attention based models such as SASRec (Kang and McAuley, 2018) and BERT4Rec (Geng et al., 2023) move from these limited representations to sequences. These sequences are lists of item IDs that the user has interacted with that are then embedded into a high dimensional vector space. More recent LLMs such as TASTE (Liu et al., 2023) leverage even more sequence information, allowing models to use not just item IDs, but other sources of text such as item titles, brands, and product categories.

Examining existing sequential recommendation algorithms, it is apparent the task of personalization is not fully explored. The datasets the models are trained on span thousands of users (e.g. Amazon Reviews (McAuley et al., 2015)). We investigate different levels of pesonalization using these datasets. At one end of the spectrum, models have no user specific weights and no information about items. These models are trained to only observe sequences of item IDs. TASTE takes a step forward by fine-tuning a single transformer model and utilizes text matching clues in the prompt for recommendation. The model improves recommendation performance, but training still takes place across a wide variety of users - it does not directly cater to individuals. In the extreme case, each user gets their own model, trained from scratch on only their data. However, due to large data requirements, this remains impractical in a real world setting.

Our approach investigates the personalization spectrum by training an LLM called TASTE, at the user-level and cohort-level, we further contrast personalized and random cohorts as well as personalization for highly active users. Through this we observe and analyze changes in recommendation performance and efficiency compared to existing methods that do not consider such techniques.

1.3 Approach

We use TASTE as our base model architecture for personalization. Then, using collections of user-level data, we personalize this model at the user level through fine-tuning with LoRA (Raffel et al., 2023). Further, we investigate how to improve efficiency by clustering users into cohorts based on historical item similarities and user features. We look into the concept of personalization at different levels of granularity to understand the potential benefits and trade-offs of personalization in sequential recommendation. Random cohorts serve as a baseline to compare against the personalized cohort approach, and the effectiveness of using historical item similarity for cohort creation. We use Recall@10, Recall@20, NDCG@10 and NDCG@20 for evaluation metrics.

2 Related Work

Similar works on the sequential recommendation task lay the foundation for our investigation of LLM based personalization methods. GRU4Rec (Hidasi et al., 2016), SASRec (Kang and McAuley, 2018), and BERT4Rec (Geng et al., 2023) process sequences of item IDs without any text data, representing the evolution of sequential recommendation over time. GRU4Rec utilizes a recurrent neural network architecture to observe sequences of item IDs. SASRec makes use of self-attention modules to selectively attend to sequence data. BERT4Rec takes a different approach than SASRec to attention based recommendation by bi-directionally attending to sequences of item IDs during training.

DIF-SR (Xie et al., 2022) is the state of the art model prior to TASTE. This model emphasizes side information by encoding item IDs and text attributes separately before fusing the representations for prediction.

As previously described, TASTE (Liu et al., 2023) is an LLM based on T5 (Raffel et al., 2023). It injects text attributes like titles for items, location names and addresses for stores as well as item IDs into user-item history sequences. It concatenates a user's text based item interaction history as a prompt, then takes the decoder's final state representation of this prompt and uses this vector representation in a similarity search to look up the next item to recommend in an item database.

3 Method

Our primary contribution defines a process of pre-training and fine-tuning TASTE at a few levels of data granularity in order to capture increasing levels of personalization.

3.1 Data Preparation

We start by chronologically splitting historical user-item interaction histories into two parts. The first 80% of each users history we denote as Historical Data and the last 20% of each users history we call Future-Test. Next, we subdivide the Historical Data into two components: the first 75% we call Historical Pre-train (representing the first 60% of total user history) and the next 25% (representing the next 20% of total user history) is Historical Continue Train. From each of Historical Pre-train and Historical Continue train, a per-user leave-one-out validation set is created.

3.1.1 Cohort and User Levels

We cover the spectrum of LLM personalization, by partitioning Historical Continue Train Data into two levels of granularity: Cohort-level and User-level. Cohorts are groups of similar users where similarity is defined by various methods such as user-interaction history or product rating distributions. The user-level naturally divides the Historical Continue Train Data on a per-user basis.

We take two general approaches to cohort users. Our first approach is to create cohorts based on text similarity across user-item interaction histories, we reference this approach as 'Text Embed' in the Results section. With this method we create two clustering models with KMeans: an item clustering model and a user clustering model to create user cohorts. First, items within the Historical Continue Train data set are processed using Sentence Transformers to embed item text representations into a high dimensional vector representation. We apply KMeans over item vectors to generate groups of similar items which we call item-clusters. Next, we count how many times each user within Historical Continue Train Data interacts with an item in each item-cluster. We use these item-cluster

interaction frequencies as new features for each user and apply KMeans to create cohorts of similar users. In essence, users in the same cohort tend to interact with the same item clusters and therefore have similar text sequence data.

Another approach for cohorting is through creating several user-level features like interaction count, average rating, item interaction count, frequently bought category and frequently bought brand. Similar to the text embedding approach, we only use the Historical Continue Train data set to create these features. These features are derived from the user-item interaction data and are used to characterize the behaviour and preferences of users within each cohort. In this approach, the actions are first clustered using KMeans and then users are assigned to cohorts in which they have the maximum number of actions. By clustering the actions first and then assigning users based on their predominant action type, this approach aims to group users exhibiting similar patterns of interaction with items.

3.2 Model Definitions

The following models enumerate the personalization spectrum from very low to extreme levels of personalization.

1. Model 1 (Very Low) is a model with no personalization. This model is T5-base with the TASTE architecture and is not trained on any of the previously described datasets.

2. Model 2 (Low) represents a model with low levels of personalization. This model uses pre-training on all weights, to domain-adapt Model 1 to our data sets and the recommendation task. Historical Pre-train is used as the training set, Historical Pre-train's validation set for validation and all of Future-Test as the test set. A learning rate of 1e-4 is used.

3. Model 3 (Low-Full) represents a domain-adapted and fine-tuned model with low levels of personalization by fine-tuning across all data of Historical Continue Train. It is our baseline model we need to outperform in order to demonstrate performance gains from personalization. In this instance, Model 3 expands upon Model 2 by freezing Model 2 weights, then adding LoRA weights to all linear layers with rank 32, LoRA alpha 64, and 0.05 LoRA dropout. Additionally, the learning rate is lowered by 10x to 1e-5. This learning rate is also used in the remaining models.

4. Model 4 (Moderate Personalization Cohorts) represents moderate levels of personalization. Here, LoRA weights are attached to a frozen Model 2 and a separate model is trained on each partition of users for a given cohort and clustering strategy. Though we experiment with various cohort creation strategies, in our best performing approach, one special cohort is created for users with less than 10 user-item interactions in the Continue Train set. We call this group the Low Activity Cohort. Remaining users are divided with other clustering strategies. In order to evaluate this model, the Future-Test set is split across users in their corresponding cohorts and performance metrics are aggregated using a weighted average of the number of items in each cohort's Future-Test set.

5. Model 5 (Extreme Users) represents extreme levels of personalization. Each user gets their own model by using a frozen Model 2 as a backbone and then adding LoRA weights with rank 8. These weights are fine-tuned, for each user, on their personal Historical Continue Train data. Individual user models are then evaluated based on that user's data within Future-Test.

6. Model 6 (Moderate Personalization Random Cohorts) is used in contrast to Model 4. Here, LoRA weights are attached to a frozen Model 2 and a separate model is trained on randomly partitioned users from the Historical Continue Train data. The number of users in each random cohort matches the number of users from each clustered cohort defined in its Model 4 counterpart for comparison purposes. The same partition with less than 10 user-item interactions defined in Model 4 is used here. The same process is followed for applying the test set and evaluation as Model 4.

7. Model 7 (Extreme-Frequent Users) is a different application of the user-level models from Model 5. Now, only frequent users with activity levels above a certain threshold (e.g. the top 20% most active users in the dataset) have their own personalized model applied to them at test time. For other less active users, Model

3 is applied instead. Again, users are evaluated based on their individual data within Future-Test.

4 Experiments

4.1 Datasets

Our main dataset is the 2018 version of Amazon Toys. It is then filtered down such that only users with at least 15 interactions per user and 5 interactions per item are left. Preprocessing code for the datasets is publicly available through the DIF-SR (Liu et al., 2023) open-source project. See Table 1 for statistics and Section 3.1 for processing steps.

Users	Items	Total Reviews
7,892	14,512	204,177
Pre-train	**Continue Train**	**Future-Test**
116,347	28,131	36,023

Table 1: Amazon Toys and Games 2018 (Filtered)

Users	Items	Total Reviews
6,523	16,927	240,959
Pre-train	**Continue Train**	**Future-Test**
139,427	37,720	44,243

Table 2: Yelp 2018 (Filtered)

Another publicly available sequential recommendation dataset we use in common with TASTE is Yelp. Overall, Yelp offers a far greater number of users and interactions. However, the full size of Yelp 2018 is too large to train our model in a reasonable amount of time, therefore we randomly sample 20% of users then apply the same user and item filtering as Amazon Toys. We use Yelp to verify the results of cohort and user level experiments made on Amazon Toys. See Table 2 for statistics and Section 3.1 for processing steps.

4.2 Baselines

We use the TASTE model architecture with an added set of LoRA weights trained on both Historical Pre-train and Historical Continue train as a baseline, which is represented by Model 3.

4.3 Results

See Table 3 for metrics and model comparisons on Amazon Toys and Table 4 for Yelp. Models 4 and 6 comprise multiple cohort models; thus, their metrics are a weighted average across each

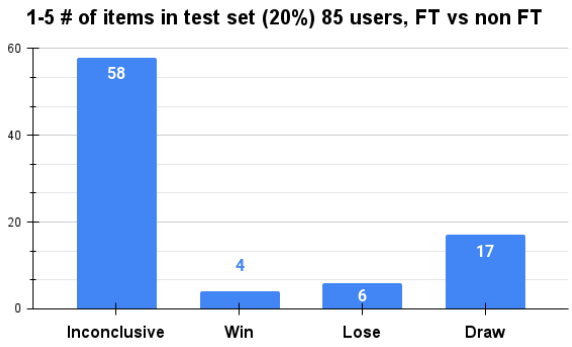

Figure 1: User level approach results for very low activity users on Amazon Toys

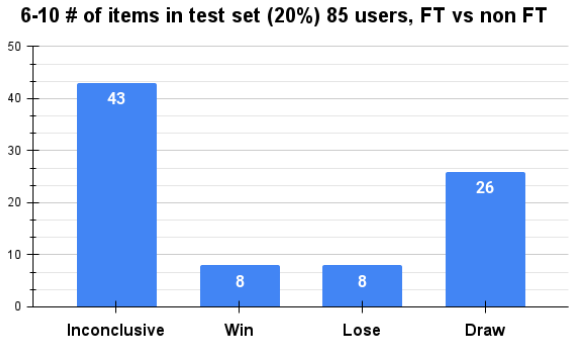

Figure 2: User level approach results for low activity users on Amazon Toys

cohort, weighted by the number of test samples in each cohort's Future-Test set. To compare Model 4 to a random control, Model 6 was created with the same number of cohorts, the same number of users per cohort, and an identical low-activity user cohort. The only difference is that, outside of the low-activity user cohort, the remaining users in Model 6 are randomly partitioned.

In Table 3, Model 4.1 uses the 'Text Embed' approach, which clusters users based on item similarity in their user-item interaction histories. Model 4.1 is comparable to Model 6.1. Model 4.2 uses the same low-activity cohort as Models 4.1 and 6.1 but utilizes engineered features to create cohorts. Model 4.2 is comparable to Model 6.2.

Upon examining Models 4.1, 4.2, 6.1, and 6.2, we conclude grouping users into cohorts based on historical item similarity and user feature similarity does not improve recommendation performance compared to randomly assigning cohorts using our current clustering strategies. Table 3 indicates that cohort personalization (Model 4) does not improve performance over the baseline (Model 3) on the

Model	Pre-train Data	Fine-tune Data	Recall@10	Recall@20	NDCG@10	NDCG@20
1. Very Low	N/A	N/A	0.0002	0.0012	0.0001	0.0004
2. Low	Pre-train	N/A	0.1149	0.1553	0.0671	0.0773
3. Low-Full	Pre-train	Continue Train	0.1317	0.1786	0.0752	0.0871
4.1 Text Embedded Cohorts	Pre-train	Text Embed Cohort	0.1296	0.1754	0.0749	0.0865
6.1 Random Cohorts	Pre-train	Random Cohort	0.1300	0.1758	0.0752	0.0868
4.2 User Feature Cohorts	Pre-train	Feature Engineered Cohorts	0.1296	0.1753	0.0748	0.0864
6.2 Random Cohorts	Pre-train	Random Cohort	0.1286	0.1745	0.0744	0.0861

Table 3: Evaluation Metrics on Amazon Toys. An underline is used to mark the highest score.

Model	Pre-train Data	Fine-tune Data	Recall@10	Recall@20	NDCG@10	NDCG@20
1. Very Low	N/A	N/A	0.0007	0.0015	0.0003	0.0005
2. Low	Pre-train	N/A	0.0334	0.0596	0.0159	0.0224
3. Low-Full	Pre-train	Continue Train	0.0414	0.0745	0.0195	0.0278
4. Text Embedded Cohorts	Pre-train	Text Embed Cohort	0.0409	0.0754	0.0194	0.0280
6. Random Cohorts	Pre-train	Random Cohort	0.0413	0.0756	0.0195	0.0281

Table 4: Evaluation Metrics on Yelp

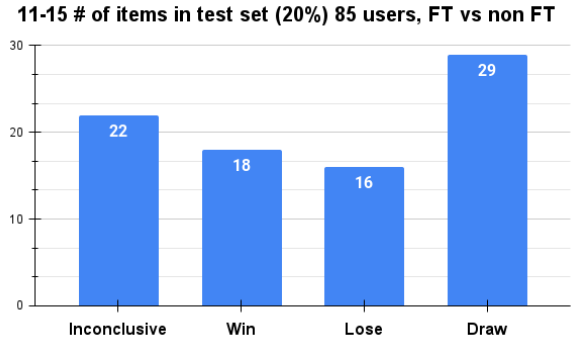

Figure 3: User level approach results for high activity users on Amazon Toys

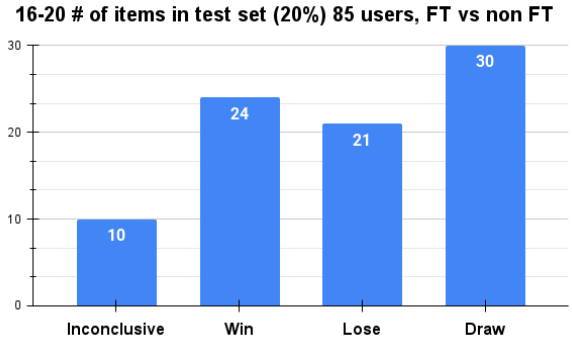

Figure 4: User level approach results for very high activity users on Amazon Toys

Toys dataset. However, in Table 4, Models 4 and 6 show improvement over Model 3 for Recall@20 and NDCG@20 metrics on the Yelp dataset, but the performance difference is negligible.

An interesting observation arises from Table 6 where we compare the performance of each modeling strategy as the number of cohorts increases. From this table, we conclude that adding a low activity cohort improves performance in both clustered and randomly partitioned user groups. Additionally, increasing the number of clusters does not significantly decrease performance (compare Model 4 performance as the number of cohorts increases). We believe LoRA can be used for distributing training across separate models focused on cohorts of different users without significantly decreasing performance across their weighted average, enabling shorter training times.

For Models 5 and 7, we feel using a weighted average for evaluating aggregate user-level performance is less expressive of the performance of this modeling approach. Instead, we divide user-level fine-tuning results into 4 categories: win, loss, draw, and inconclusive. We compare the performance of each user on their slice of the Future-Test set by performing inference on both Models 3 and 5. If the user-level modeling approach outperforms the baseline in at least 2 out of the 4 evaluation metrics, we consider that a win. If the user-level modeling approach under performs in at least 3 of the evaluation metrics in comparison with the baseline, we consider that a loss. If user-level model and the baseline end up in an absolute draw with all 4 evaluation metrics equal to each other but not 0, we consider those as draws. Finally, when both the baseline and the proposed approach result in zeros across all 4 evaluation metrics, we consider those inconclusive tests.

Refer to Figure 1 through 4 for the results. The relationship between the increasing activity levels of the users and the model tests results is very clear. Although in reality there are a lot more users, we

can see a general pattern that emerges over the progression of these plots with 85 users. We see that Model 5 as well as the baseline, when evaluating the very low and low activity users, fail to correctly recommend their future reviewed items. We can see an overwhelming majority of inconclusive/draws, showing that the baseline and Model 5 are unable to learn user preferences because of lack of historical data. As we move to Figure 3 and 4, we see definite performance differences between the models. These reveal that as the user's activity increases, the models are able to gauge the preferences of the users. With regard to wins and losses, the per-user model approach successfully understands the user preferences and gives slightly better or at par recommendations as the baseline.

4.4 Error Analysis

4.4.1 Cohort-Level Error Analysis

Consider the difference in performance between Models 3, 4, and 6 on Amazon Toys and Yelp (Table 3 and Table 4). On Yelp data, Models 4 and 6 outperform Model 3 on Recall@20, NDCG@20 then match or come close to matching Recall@10 and NDCG@10, unlike Amazon Toys. We suspect the performance differences between cohort level personalization models could be due to user activity that reflects the interests of individuals other than the target user. For instance, when purchasing toys on Amazon, a user might be a parent who buys toys for their child, but also for other family members or friend's children who may have wildly different preferences. On the other hand, users within the Yelp data set are more likely to be individuals visiting locations that reflect their own preferences.

We rely on KMeans to create clusters for both users and items. Here, we look at the relationship between cohort (or cluster) strength, as measured by silhouette scores and its impact on model performance. Figure 5 contains the silhouette plot for cohorts generated using an item similarity clustering approach we call 'Text Embed' after removing the Low Activity Cohort. The top three cohorts with the highest silhouette scores are: 0, 4, then 3. However, Table 5 shows the top three highest Recall@10 scores are cohorts: 4, 0, then 2, with scores of 0.2752, 0.2006, and 0.0962 respectively. While Cohorts 0 and 4 both have outsized silhouette scores and recommendation performance compared to the other cohorts, this relationship breaks down when we examine cohort 3. This cohort has the third highest silhouette score, but the worst recommendation performance metrics out of all cohorts with Recall@10 at 0.0826. Within cluster similarity does not appear to directly relate to cohort performance. We see this evidence re-appear in a similar experiment represented in Figure 7 and Table 8. In this experiment, we do not use a Low Activity Cohort, instead we cohort *all* users under the 'Text Embed' approach. We observe Cohort 2 to have the highest silhouette score followed by Cohort 0. However, Table 8 indicates Cohort 2 has the second highest Recall@10 score while Cohort 0 has the lowest Recall@10 across all cohorts. Further, these two cohorts have the lowest gains in Recall@10 with increases in performance of only 3.85% and 2.25% for Cohorts 0 and 2 respectively. Additionally, Cohort 1 has the largest gain in Recall@10 with a 14.85% increase, but the worst silhouette score. A positive impact on recommendation performance is revealed in Tables 5 and 8. They demonstrate an increase in Recall@10 percentage gains as the number of training samples increase across each cohort. We conclude the amount of training data bears a stronger relationship to recommendation performance than silhouette scores under our current methods.

4.4.2 User-Level Error Analysis

Considering the user-level modeling approach, the model is fine-tuned with user specific historic-continued data, and the results change depending on the users' activity levels. When the quantity of the users' activities is higher, the model is exposed to more data for those users. This results in the increased bias for high activity users. When considering low activity users, it might be the case that 100% of their history is less than 20% of a high activity user. This results in the per-user approach to fail for such users. Lower activity users tend to result in inconclusive test results. As stated above, the generalized model which we use as a baseline is also not immune to this bias and ends up failing to make relevant recommendations. As the user activity increases, we can see that the model is able to decide on some relevant recommendations with the increase in the number of wins, losses and draws. As we look deeper into the evaluation metrics for each user for baseline and per user personalization, we see that per user personalization does not have a fatal failure as in most of the per user personalized results are not too far off from the baseline. The results are so close that we can introduce a threshold

of difference where we consider a certain range of error in the evaluation metrics.

5 Conclusion

From enumerating the spectrum of personalization, we make the following conclusions about our research goals when constrained to our cohort-level and user-level training methodologies:

1. Personalization does not necessarily increase recommendation performance.

2. Personalization by creating cohorts of similar users is just as effective as creating non-personalized random cohorts.

3. Personalization at the user-level can increase performance for some users and increases in effectiveness as the length of their history increases.

Our study reveals a nuanced relationship between personalization and user-level performance, wherein personalized recommendations yield improved outcomes for specific users, particularly those with a longer history of user-item interaction. As users accumulate a richer historical context, personalized recommendations become increasingly tailored to their preferences and exhibit greater efficacy in meeting their needs.

6 Future Work

Constrained by time, we acknowledge several promising avenues for future exploration and analysis.

An enhancement to our phased training strategy could be made by utilizing only the first 20% of the Historical data set for pre-training. This allows for a larger portion of our data to be available for fine-tuning LoRA to better reflect our users and cohorts.

An area of potential advancement lies in refining text embeddings to better capture relationships between items to enhance item clusters. We see promise in removing item brands from text descriptions. This way, different items with the same brands could have closer vectors in high dimensional space.

There is a chance personalization is a short-term effect. In essence, cohort and user-level over-fitting may benefit in finding a relevant next 'few items', but not the next 'many items'. To test this effect, evaluate each modeling strategy with metrics such as Recall@5 and NDCG@5.

Finally, we feel a critical flaw in our approach is the lack of user demographic data. Most data used in our clustering methods are also explicitly seen by the LLM. However, an LLM with sufficient capacity could theoretically capture a more complex relationship across all users, overpowering the effect of personalization. We propose adding demographic data accounting for age, gender, occupation, and location information which could provide the strongest signals of user similarity to our KMeans / clustering models. By creating cohorts based on this information, we may also be able to implicitly add user similarity information to the LLM even if such demographic data is omitted in the prompt all while saving context length.

7 Contributions

- Eric: Identified data processing steps for creating user-item interaction sequences within the RecBole (Zhao et al., 2021) information retrieval toolkit by reading documentation and contacting TASTE paper authors. Helped teammates understand and modify TASTE model code. Validated data splitting approaches within RecBole by recreating TASTE data sets. Processed data for Toys and Yelp data sets at various filtering levels to ensure reasonable training times while providing sufficient data quantity for creating user cohorts. Explored approaches for creating text embedding cohorts. Wrote scripts for processing data sets at the cohort level, analyzing content and quality of cohorts, and generating random cohorts. Attached LoRA to TASTE model, searched over rank and LoRA alpha hyperparameters, and identified high performance settings. Trained, validated, tested, and conducted error analysis on several iterations of Models 1-4 and 6 for Toys and Yelp data sets. Lead weekly masters student and mentor discussions. Summarized key take-aways from industry mentorship meetings into actionable tasks for each team member. Outlined and wrote majority of project presentation slides, reports, and poster.

- Prahitha: Started off by working on the Amazon Beauty data set and pivoted to Toys after deciding on the current pipeline and data splitting strategy. Explored multiple ways to create cohorts and worked on the feature engineered cohorts for both Beauty and Toys. Explored ways to gather and incorporate additional item metadata like pricing information for Toys to

improve performance. Modified the data processing pipeline to accommodate MovieLens though we did not end up using it.

- Darsh: Initiated the secondary "user-level" model approach. Identified and modified data processing steps within the TASTE data processing to accommodate per user model development. After initial proof-of-concept for 10 user models, further trained 340 models for generalization and analysis of the results and accuracy. Accompanied LoRA implementation to the base TASTE model for testing and then furthering it with our user specific fine-tuning approach. Initiated the investigation of the causality between the number of user actions that the model is trained/tested on. Trained, validated, and tested Model 5, with modifications in accordance with user-level accommodated data processing. Assist teammates with technical issues related to migration of files, successful model pipeline runs and alternative approaches to roadblocks. Assisted in reviewing and editing the paper.

- Michael: Examined Amazon Toys dataset and initially worked with cohort-level tuning without LoRA. Worked on implementing baselines like SASRec for comparison but moved away from it after further details about the data/model pipeline were specified. Worked with Darsh on user-level fine tuning, implementing a pipeline for user-level model creation and evaluation based on activity levels. Examined user-level results and potential causes for them. Helped write and edit presentation slides/reports, and create the poster board.

8 Acknowledgements

We would like to thank our industry mentors, Praveen Ravichandran, Ghazal Fazelnia, and Paul Bennett of Spotify, for originating this project and for providing our team with a clear direction, resources, and support throughout our time together. We are grateful to Sheshera Mysore and Shib Dasgupta for aiding us in our efforts and interpreting results. We thank Hamed Zamani for critical feedback on our project through the semester. Finally, we extend our appreciation to Wenlong Zhao, Dhruvesh Patel, and Andrew McCallum for enabling this unique research opportunity.

References

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2023. Recommendation as language processing (rlp): A unified pretrain, personalized prompt predict paradigm (p5).

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation.

Zhenghao Liu, Sen Mei, Chenyan Xiong, Xiaohua Li, Shi Yu, Zhiyuan Liu, Yu Gu, and Ge Yu. 2023. Text matching improves sequential recommendation by reducing popularity biases.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *The Web Conference*.

Yueqi Xie, Peilin Zhou, and Sunghun Kim. 2022. Decoupled side information fusion for sequential recommendation.

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.

9 Appendix

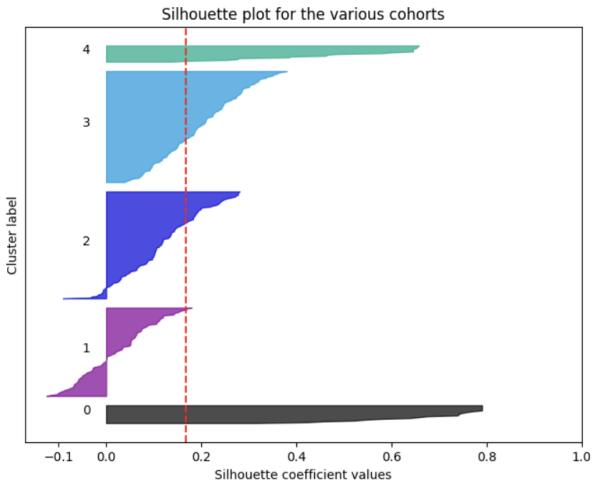

Figure 5: Amazon Toys: Silhouette Plot for 5 Text Embed Cohorts. Avg Silhouette Score: 0.16

Model	Recall@10	Recall@10 % Gain	Recall@20	NDCG@10	NDCG@20	Users	Training Samples
3. Low-Full	**0.1317**	14.6	**0.1786**	**0.0752**	**0.0871**	7892	28131
4. Low Activity Users	0.1358	14.6	0.1819	0.0791	0.0908	7486	21078
4. Text Embed 0	0.2006	-1.6	0.3072	0.1009	0.1279	22	297
4. Text Embed 1	0.1062	3.1	0.1513	0.0573	0.0686	105	1759
4. Text Embed 2	0.0962	2.6	0.1272	0.0547	0.0625	127	2357
4. Text Embed 3	0.0826	12.5	0.1275	0.0466	0.0579	132	2362
4. Text Embed 4	0.2752	1.3	0.3624	0.1572	0.1792	20	278
4 Weighted Average	0.1296	12.8	0.1754	0.0749	0.0865	7892	28131
6. Random Cohort 20	0.0923	7.6	0.1352	0.048	0.0586	20	446
6. Random Cohort 22	0.0755	0.0	0.1042	0.0493	0.0546	22	362
6. Random Cohort 105	0.1172	1.4	0.1632	0.0643	0.0759	105	1764
6. Random Cohort 127	0.0984	4.7	0.1427	0.0535	0.0647	127	2150
6. Random Cohort 132	0.1173	13.3	0.164	0.0675	0.0794	132	2331
6. Weighted Average	0.1300	12.2	0.1758	0.0752	0.0868	7892	28131

Table 5: Evaluation Metrics across each Amazon Toys Cohort. Model 4 Text Embed Cohort labels are an identifier for each cluster. Model 6 Random Cohorts are labeled with the number of users within the cohort. Recall@10 % Gain represents the percentage increase in performance when applying the test set from a cohort to Model 4, as compared to Model 2.

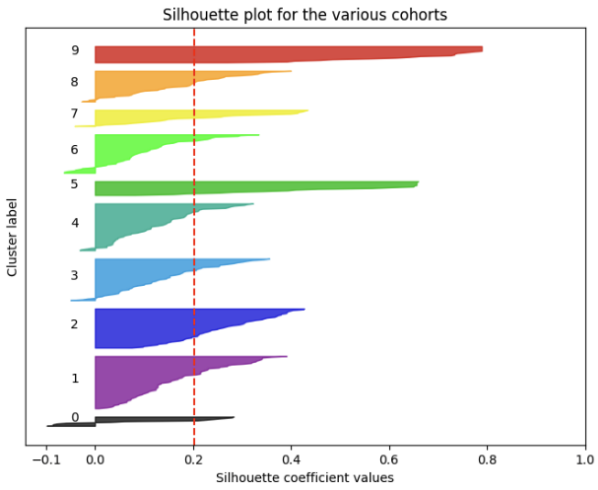

Figure 6: Amazon Toys: Silhouette Plot for 10 Text Embed Cohorts. Avg Silhouette Score: 0.20

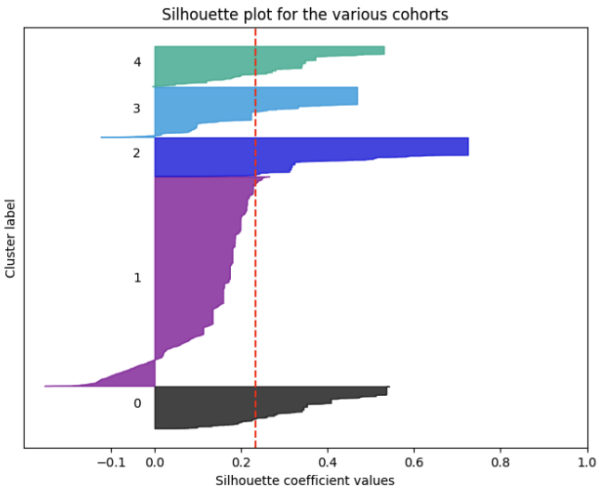

Figure 7: Amazon Toys: Silhouette Plot for 5 Text Embed Cohorts. No low activity Cohort. Avg Silhouette Score: 0.23

Model	# Cohorts	Recall@10	Recall@20	NDCG@10	NDCG@20	Recall@10 % Diff.
3. Low-Full	1	0.1317	0.1786	0.0752	0.0871	0.0
4. Low Activity and High Activity	2	0.1287	0.1745	0.0741	0.0857	-2.3
4. Text Embed Only	5	0.1266	0.1707	0.0730	0.0842	-3.9
6. Random Cohorts (Pair with above)	5	0.1248	0.1692	0.0722	0.0834	-5.2
4. Low Activity and 5 Text Embed Cohorts	6	0.1296	0.1754	0.0749	0.0865	-1.6
6. Random Cohorts (Pair with above)	6	0.1300	0.1758	0.0752	0.0868	-1.3
4. Low Activity and 10 Text Embed Cohorts	11	0.1294	0.1754	0.0748	0.0864	-1.7

Table 6: Distributed LoRA Performance. Recall@10 % Diff. is the percent change from Model 3's performance

Model 4 Cohort	Recall@10	Recall@20	NDCG@10	NDCG@20	Users	Training Samples
Low Activity Users	0.1358	0.1819	0.0791	0.0908	7486	21078
Text Embed 0	0.0773	0.1134	0.0518	0.0608	13	181
Text Embed 1	0.0943	0.1275	0.0553	0.0636	69	1405
Text Embed 2	0.067	0.1088	0.0351	0.0456	52	858
Text Embed 3	0.1237	0.1671	0.0638	0.0747	55	1028
Text Embed 4	0.0845	0.1343	0.0478	0.0604	62	1122
Text Embed 5	0.2826	0.3732	0.1616	0.1845	19	257
Text Embed 6	0.0931	0.1297	0.0536	0.0627	51	905
Text Embed 7	0.1111	0.1626	0.0629	0.0758	22	347
Text Embed 8	0.0893	0.1297	0.0492	0.0594	41	653
Text Embed 9	0.2006	0.3072	0.1009	0.1279	22	297
Weighted Average	0.1294	0.1754	0.0748	0.0864	7892	28131

Table 7: Evaluation Metrics for Low Activity Cohort and Text Embed Approach with 10 Cohorts.

Model 4 Cohort	Recall@10	Recall@10 % Gain	Recall@20	NDCG@10	NDCG@20	Users	Training Samples
Text Embed 0	0.1000	2.25	0.1386	0.0627	0.0724	872	2677
Text Embed 1	0.1106	14.85	0.1489	0.0666	0.0757	4345	17649
Text Embed 2	0.1755	3.85	0.2382	0.0934	0.1093	800	2437
Text Embed 3	0.2032	8.95	0.2772	0.1055	0.1243	1039	3153
Text Embed 4	0.1154	2.03	0.1475	0.0697	0.0778	836	2215
Weighted Average	0.1266	10.16	0.1707	0.0730	0.0842	7892	28131

Table 8: Evaluation Metrics for Text Embed Approach with 5 Cohorts. No Low Activity Cohort.