Transformers with multi-modal features and post-fusion context for e-commerce session-based recommendation

Gabriel de Souza P. Moreira
gmoreira@nvidia.com
NVIDIA
São Paulo, Brazil

Sara Rabhi
srabhi@nvidia.com
NVIDIA
Ontario, Canada

Ronay Ak
ronaya@nvidia.com
NVIDIA
Florida, United States

Md Yasin Kabir
mkabir@nvidia.com
NVIDIA
Missouri, United States

Even Oldridge
eoldridge@nvidia.com
NVIDIA
British Columbia, Canada

ABSTRACT

Session-based recommendation is an important task for e-commerce services, where a large number of users browse anonymously or may have very distinct interests for different sessions. In this paper we present one of the winning solutions for the Recommendation task of the SIGIR 2021 Workshop on E-commerce Data Challenge. Our solution was inspired by NLP techniques and consists of an ensemble of two Transformer architectures – Transformer-XL and XLNet – trained with autoregressive and autoencoding approaches. To leverage most of the rich dataset made available for the competition, we describe how we prepared multi-model features by combining tabular events with textual and image vectors. We also present a model prediction analysis to better understand the effectiveness of our architectures for the session-based recommendation.

ACM Reference Format:
Gabriel de Souza P. Moreira, Sara Rabhi, Ronay Ak, Md Yasin Kabir, and Even Oldridge. 2021. Transformers with multi-modal features and post-fusion context for e-commerce session-based recommendation. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR eCom'21).* ACM, New York, NY, USA, 7 pages.

1 INTRODUCTION

In many recommendation domains such as e-commerce, news, streaming video and music services, users might be untrackable, their histories can be short, and users can have rapidly changing tastes[34]. Providing recommendations based purely on the interactions that happen in the current session is an extremely important and challenging problem. Many methods have been proposed that leverage the sequence of interactions that occur during a session, including session-based k-NN (k-Nearest Neighbours) algorithms like V-SkNN [22] and neural approaches like GRU4Rec. [15].

Within the NLP domain, approaches based on Transformers architectures [36] have demonstrated significant advantages over sequential and CNN based approaches. Transformers apply self-attention to the input sequence, allowing the model to focus on the most important inputs, while also providing improved access to inputs that occur further back in the sequence. These improvements are important when comparing the effectiveness of Transformers relative to RNN-based models that have limited access to the history of the sequence.

In this paper we present our solution as one of the winners for the Recommendation task of the SIGIR 2021 Workshop on E-commerce Data Challenge [33]. It consists of an ensemble of two different Transformer architectures – Transformer-XL and XLNet – trained with autoregressive and autoencoding approaches inspired by Natural Language Processing (NLP). We leveraged the rich information provided by the dataset, and explored different ways to combine tabular data of user interactions events (e.g., clicks, add-to-card, remove-from-cart, purchases, search queries) with unstructured data (product description and images) in a multi-modal approach.

In §2 we introduce the related work, in §3 describe the competition and dataset, and in subsequent sections we describe our solution for the competition, including data preprocessing and feature engineering in §4, model architectures, training and evaluation in §5, and finally our ensembling approach and results in §6.

2 RELATED WORK

In this section, we describe previous work on sequential and session-based recommendation tasks and also how Transformers architectures, originally proposed for NLP, have been adapted for recommendation tasks.

2.1 Sequential and Session-based Recommendation

Sequential recommendation approaches have been proposed to capture sequential and temporal dependencies between user interactions, including algorithms like Markov models [12, 14, 20] and neural architectures like Recurrent Neural Networks (RNNs) [16], Convolutional Neural Networks (CNNs) [35, 37] and attention-based networks [38]. More recently Transformer architectures [5, 19, 21, 31, 39, 43] have been applied for CTR prediction and next-item recommendation leveraging the sequence of past user interactions. In many real-world applications, however, such longer-term information is often not available, because users are not logged in or because they are first-time users. To address the issue, session-based recommender systems [24] have been proposed to model the sequence of interactions within the current user session, with no

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGIR eCom'21, July 15, 2021, Virtual Event, Montreal, Canada
© 2021 Copyright held by the owner/author(s).

access to past user interactions. In this context, a session is a short sequence of user interactions, typically bounded by user inactivity.

Algorithms based on Session k-NN [24] and neural networks like RNNs [8, 11, 16], CNNs [42], and memory networks [25] have been explored for session-based recommendation.

2.2 Transformers for Session-based recommendation

More recently, Transformers and the self-attention mechanism were also explored for session-based recommendation [6, 10, 23, 28, 32, 40, 44]. Most of those works use a causal Language Model (LM) approach for training (auto-regressive), i.e., using only interactions before the target item as input. Only [6] use a training scheme similar to BERT [4, 9] – masked LM (autoencoding) – which randomly masks items in the sequence for prediction, allowing during training the usage of future interactions on the right of the masked items.

In a recent competition, the NVIDIA RAPIDS.AI team won the ACM WSDM 2021 WebTour Workshop Challenge, organized by Booking.com[13]. The task was predicting the last booked city in a multi-destination trip and we treated it as a session-based recommendation problem. Our solution[30] for that competition used an ensemble of a Transformer (XLNET), a GRU-based and an MLP-based architectures.

Our solution for this SIGIR 2021 Workshop on E-commerce Data Challenge was purely based on Transformers, challenging once more the status-quo of recent RecSys competitions that were won by non-deep learning solutions [18]. We employed causal LM and masked LM training approaches with Transformer-XL and XLNet, respectively, which are described next.

Transformer-XL [7] is a *causal language model* that solves context fragmentation by introducing a segment recurrence mechanism where the hidden states of previous blocks are cached and used to extend the context of the new upcoming segment.

XLNet [41] takes advantage of both causal and masked language modeling, while addressing their limitations, by defining a new training objective called *permutation Language Modeling*. In this work, we trained XLNet with masked LM, as we empirically found that it provided higher accuracy compared with its original permutation LM, maybe because of the shorter session length.

For the competition, as the majority of sessions are short we feed the user session sequences truncated to the last 30 interactions to the Transformer models and do not use the segment recurrence mechanism defined in Transformer-XL and XLNet.

3 THE CHALLENGE

The SIGIR 2021 Workshop on E-commerce Data Challenge [33] was organized by a partnership between academic researchers and Coveo company. The competition presented two tasks: (1) session-based recommendation and (2) cart-abandonment.

We competed for both tasks and open-source the code and documentation of our solutions on GitHub[1]. For space reasons, this paper describes only our solution for task (1), but a detailed report on our approach for task (2) can be found in that GitHub repository.

[1]https://github.com/NVIDIA-Merlin/competitions/tree/main/SIGIR_eCommerce_Challenge_2021

For the recommendation task (1), the models were evaluated for their ability to predict the *immediate next product* interacted by the user in a session (*Mean Reciprocal Rank - MRR*) and to predict *all subsequent interacted products* in the session, up to a maximum of 20 after the current event (*F1 score*).

3.1 Competition Dataset

This competition provided a very rich dataset in terms of the diversity of data that could be relevant for personalized recommendation in e-commerce. It contains more than 37 million events distributed in almost 5 million sessions and is composed of three tables: (1) browsing events, (2) search events and (3) sku content.

The browsing table (1) contains logs of user events on product pages (view, detail, add-to-card, remove-from-cart and purchases) and also views of non-product pages (page views) like FAQ and promotions. The search table (2) contains search events associated with sessions and includes a query vector generated from the text of the search terms, a list of the products presented to the user in the search results and the products that were clicked by the user. Finally, the SKU content table (3) provides product metadata information, like the quantized price, the product category and vectors representing the text and image of the product.

In particular, as can be seen in Appendix A, we could observe a high average intra-session similarity of interacted product description and image vectors. Under the assumption that users browse on similar items in their session, it is clear that those pre-trained vectors are really able to capture the semantics of the products textual and image data.

4 PRE-PROCESSING AND FEATURE ENGINEERING

In this section we describe our approach for preprocessing and feature engineering, which was implemented using RAPIDS cuDF, a library for GPU-accelerated dataframe transformations and NVTabular, a library for tabular data preprocessing specialized for recommender systems.

4.1 Data Augmentation

For the recommendation task models are evaluated by their ability to predict the next products the user would interact in the continuation of their sessions, and only product events were expected to be predicted. The percentage of page view events on non-product URLs (70%) was higher than product browsing events (28%) and search click events (2%) for the train set. We augmented the sequence of product events with page views and encoded page views URLs together with product SKUs in a single categorical feature, as if page view URLs were *virtual* products. This was especially useful for sessions where only page view events were available. In addition we also included in the sequences the search clicks events which were available in the search table. Using this augmentation approach allowed the models to capture more fine-grained sequential patterns and improved their recommendation accuracy.

4.2 Feature Engineering

In e-commerce datasets users may interact many times with the same product in different ways, e.g., by clicking, checking product

details, adding and removing from the cart, or purchasing. In this dataset 13% of interaction are repeated within sessions.

Recommendations were evaluated by the ability to predict the next item the user will interact with regardless of event type. To address this we kept only the first event with a product and summarized the level of interest of the user in such product by means of other features: number of interactions in the same product within the session, and flags about whether the user has checked product details or whether the product was added to cart within the session.

We encoded the following categorical features as contiguous ids, to be efficiently represented by embedding tables: event type, product category and sub-category, bucketed price, and item ids encoding both product SKUs and page views URLs (*virtual* items).

We also created a feature by dividing the bucketed product price by its average for the product category and sub-category, so that patterns about browsing on more expensive or cheaper products within the category can be detected. Finally, we created temporal features to encode the product recency – elapsed time since product was first seen, log(days) –, and also encoded hour of the day and day of the week by encoded with cycling transformation, i.e., using *sine* and *cosine* functions. All these numerical features were normalized using standardization, except the cycling features which are already uniformly distributed between -1 and 1. Finally all interaction features were grouped by sessions, so that each training example represented one session.

4.3 Frequency Capping of Unpopular Products

We analyzed the distribution of the item's frequency and observed that it follows the typical long-tail we find in e-commerce interactions datasets [2], where a small number of popular products concentrate most of the interactions. As an example, from the 57,483 available product SKUs, the top-1% and top-5% most popular products account for respectively 34% and 67% of the interactions. And 50% (median) of the items receive at most 5 interactions.

As it is hard to learn meaningful item embeddings for the unpopular items, we created a variation of the preprocessed dataset which encodes all product SKUs with less than 5 interactions into the same item id. It is important to note this approach was done after the feature engineering, so that the other features of infrequent items (e.g. category, price) remain the same. This *frequency-capped dataset* was used to train one of the ensembled models.

4.4 Data splitting for Cross-Validation

The train set covered 3 months of user interactions data and the test set covered the subsequent month.

We observed from data analysis that the distribution of session length for the test set was close to the half of session length for the train set. Based on that, we reserved for validation the last 3 weeks of the train set. In addition, we split validation sessions into two halves: the first half for inference and the second half for metrics evaluation.

The sessions of our train, validation and test sets were then split into 5-folds, to allow for better cross-validation and also for more diverse ensembling for allowing training models with different chunks (Out-Of-Fold) of data.

[2]For example, the Gini index of the item frequency distributions for the Coveo dataset is 0.8849 and for the YOOCHOOSE dataset[2] is 0.8987

5 MODEL ARCHITECTURES, TRAINING AND EVALUATION

5.1 Multi-modal features processing

The tabular features were composed by categorical embedded features and numerical features represented as continuous values. We apply *layer normalization* [1] to all features individually, as we empirically observed that without it the model accuracy decreased when combining numerical features with categorical features, despite the fact that numerical features had already been standardized during preprocessing.

For the pre-trained vectors provided in the dataset based on text (search query and product description vectors) and image (product image vector) we found out that it was better to apply L2-normalization rather than using the original vectors, either with layer normalization. L2-normalization makes the feature scale similar, but also preserves the similarity relationships between similar products, aligned with [27].

5.2 Model architecture

Our base neural network architecture is presented in Figure 1. From bottom-up, all interaction features are normalized, concatenated and combined by a Fully Connected (FC) layer to produce an *interaction embedding*. The sequence of *interaction embedding* is fed to a Transformer architecture (Transformer-XL or XLNET), which outputs a vector for each position in the sequence. Those outputs are then projected by a FC layer to prediction vectors $h \in R^d$.

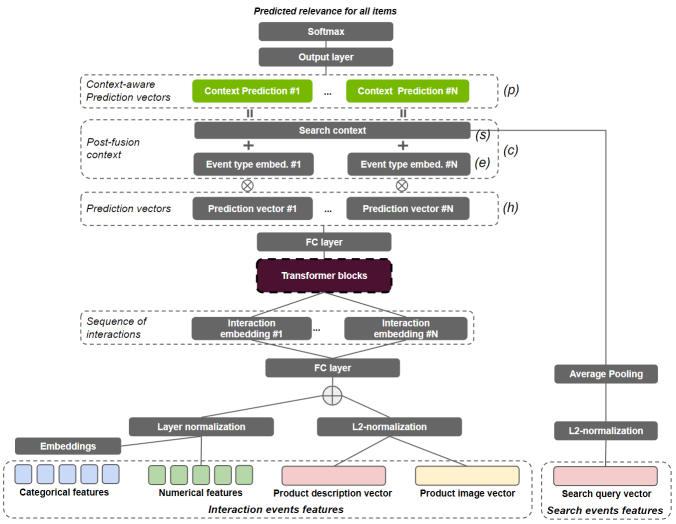

Figure 1: Our base Transformer architecture

To provide contextual information about the item to be predicted, we leveraged the *Latent Cross* [3] technique. We create *context-aware prediction vectors* p by combining the prediction vectors h with a *post-fusion* contextual vector c using element-wise multiplication $p = h \odot (1 + c)$. As the contextual vector mean is close to zero, summing 1 will center the multiplicative element distribution around 1 and can act as a mask over h. The contextual vector is

computed by $c = e + s + f$. All vectors h, c, e, s, f have the same dimension d to allow for element-wise operations. The *search context* s is calculated as the average of search query vectors (if search queries happened in the session or a zeroed vector otherwise).

The f vector represents an embedding of a boolean feature that indicates whether the item to be predicted is a infrequent item or not. This vector is only used when training on the dataset variant with item id frequency-capping (§4.3), to provide context on whether the item to be predicted is an infrequent item, so that the network can make the trivial prediction of the infrequent item id (i.e. 1) in such cases. During inference we set that boolean feature to *frequent item*, so that the network ignores the infrequent item id.

Finally, the event type embeddings e provides context about what is the event type associated to the item to be predicted (whether it is a product, search or page view event). During inference, we always set the event type to *product event*, as we are only interested in predicting items that correspond to product SKUs, and not *virtual* products that are actually page views URLs. With this approach, we increased the percentage of product SKUs among the top-100 recommended items during evaluation from 35% to 90% [3].

In the output layer, we use the *tying embeddings* technique originally proposed for NLP [17, 29], in which we share the weights of the item id embedding table with the output layer, followed by a *softmax* layer to predict the relevance scores over all items. That is possible because the context-aware prediction vectors p and item id embeddings have dimension d. This technique was originally applied for RecSys in [15] and demonstrated to be especially effective in the NVIDIA.AI team solution for Booking.com Challenge [30].

We treat the recommendation as a multi-class classification problem and use cross-entropy loss.

5.3 Training and evaluation approach

In real machine learning projects, we cannot use future data which is not available at inference time. But for machine learning competitions all available data is generally used, including test set data.

We observed that 6.7% of items present of the test set were not seen before in the train set. Thus, we decided to include the public part (first half) of test sessions in the training, so that we could learn embeddings for recently released products.

We tried three approaches to train on test set: (1) concatenating train, validation and test sets and shuffling, (2) same as (1), but sorting data by time, and (3) pre-training with train and validation set and fine-tuning only the item embeddings using the test set. The latter approach performed the best (2% better than the other approaches) and is illustrated in Figure 2.

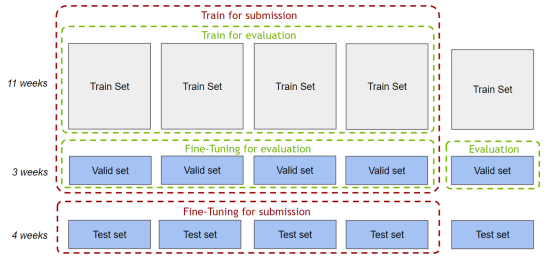

Figure 2: Illustration of training and evaluation strategy

[3]The rest of page view URLs that are eventually among the top recommended ones are removed in a post-filtering step.

For fine-tuning we freeze all the weights of the network except the item id embeddings, whose weights are shared with the output layer. For our evaluation setup, we perform an analogous approach, pre-training models with just training data and fine-tuning with validation data. Only the first half of validation sessions are used for fine-tuning, so that they are compatible with test sessions length.

Models are trained using a 5-fold strategy (§4.4), meaning that for each validation fold a model is trained using Out-Of-Fold (OOF) sessions, corresponding to about 80% data.

The full pipeline (pre-training, fine-tuning, evaluation and prediction over the test set) runtime for each model had an average 265 min with 53 min std., in an instance with one V100 GPU with 32 GB of memory and 8 CPUs. In particular, the models throughput during inference for next-click prediction is of about 800 sessions / sec. with this single GPU hardware.

6 ENSEMBLING AND RESULTS

We have used in our ensemble four variations of the base neural architecture presented in §5.2. All neural architectures use tabular features and product description vectors, but vary as follows:

- **XLNET-IM** - XLNET with image vectors;
- **XLNET-S** - XLNET with search context (post-fusion);
- **XLNET-IM-FC** - XLNET with image vectors and item frequency capping (described in §4.3); and
- **TransfoXL-IM** - Transformer-XL with image vectors.

Each of those four architectures were trained using three different hyperparameter configurations that performed well in our CV scores after hyperparameter tuning. The hyperparameters used for our models, the corresponding command lines and source code for reproducibility are available in the solution GitHub repository.

The *full ensemble* was composed of the test set predictions provided by 4 architectures x 3 hyperparameter configurations x 5 folds which results in 60 models. For each of those models, we saved the top-100 recommended items for each session and used weighted sum to produce the final recommendation lists.

In fact, some of the predictions for our *final ensemble* finished on top of the deadline hour and we had a last-minute memory issue when ensembling those large prediction files. Thus our *final ensemble* submission used only models from 2 folds (24 models), which scored 1st in the Leaderboard (LB) for F1 (0.0744) and 2nd in the MRR (0.2771) LB, very close to 1st place. As soon as the LB opened again the next day for probing, we submitted our *full ensemble* which scored 0.0747 for F1 and 0.2783 for MRR, which would have placed 1st for that metric too.

We present in Table 1 the LB scores for each architecture / its three hyperparameters configurations (suffixed by 1,2, and 3), after ensembling their 5-folds predictions. We also present the *final ensemble* and *full ensemble* LB results. It can be observed that the top-4 single models were XLNET-IM-2, XLNET-IM-3, XLNET-IM-FC-2, XLNET-IM-FC-3, which all include the product image vectors as input features. Embedding the original item ids (XLNET-IM-2, XLNET-IM-3) worked better than applying frequency capping (XLNET-IM-FC-2, XLNET-IM-FC-3). It can also be observed that the Transformer-XL, trained with causal LM approach, performed worse than XLNet which was trained with masked LM. Finally, it

can be seen that *full ensemble* improved the best single model LB score by 1.4% for MRR and 0.9% for F1.

Table 1: LB scores for individual architectures (5-folds ensembles) and for the full ensemble. Best single models for each metric are underlined.

Model	MRR	F1
XLNET-IM-1	0.2638	0.0712
XLNET-IM-2	0.2746	0.0734
XLNET-IM-3	0.2741	0.0741
XLNET-S-1	0.2634	0.0712
XLNET-S-2	0.2669	0.0713
XLNET-S-3	0.2671	0.0719
XLNET-IM-FC-1	0.2640	0.0714
XLNET-IM-FC-2	0.2726	0.0732
XLNET-IM-FC-3	0.2693	0.0724
TransfoXL-IM-1	0.2317	0.0614
TransfoXL-IM-2	0.2421	0.0632
TransfoXL-IM-3	0.2406	0.0629
Final Ensemble (24 models)	**0.2771**	**0.0744**
Full Ensemble (60 models)	**0.2784**	**0.0748**

We present an analysis of the *full ensemble* predictions to better understand the recommendation accuracy with respect to different characteristics of sessions and items in Appendix B.

7 CONCLUSION

In this paper, we present one of the winning solutions for the session-based recommendation task of SIGIR 2021 Workshop on E-commerce Data Challenge. Our solution leveraged an ensemble of Transformers models, which effectively learned sequential patterns on users browsing to predict the next interacted items. The proposed architecture leveraged multi-modal information from tabular, textual and image data for more accurate recommendations. Finally, in our prediction analysis (Appendix B) we observed how the item popularity-bias affects the recommendation accuracy.

A SIMILARITY ANALYSIS OF PRODUCT DESCRIPTION AND IMAGE VECTORS

We performed an analysis of how similar where products interacted by the users within their browsing sessions, in terms of the average similarity of their textual descriptions and product images vectors.

It can be seen in Figure 3 that the average intra-session similarity, i.e., for each pair of products within session, is generally high for both description vectors and image vectors. Under the assumption that interactions within sessions occur on similar items, it is clear that such pre-trained vectors were able to capture the semantics of textual and image data associated to the product, making them informative features to be used for next-click prediction.

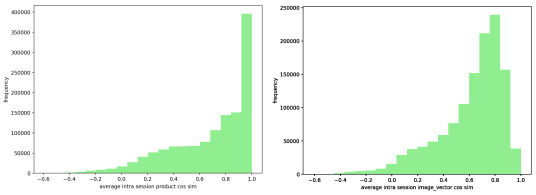

Figure 3: Distribution of avg. intra-session similarity of product description vectors (left) and product image vectors (right)

In Figure 4, it is interesting to observe how the avg. intra-session similarity based on both product descriptions and images decreases for longer session. That might indicate that the longer the sessions the less specific or targeted is the user browsing, as the interacted products are more diverse. This fact might be related to the lower recommendation accuracy we observe in longer sessions, which we discuss in Appendix B.

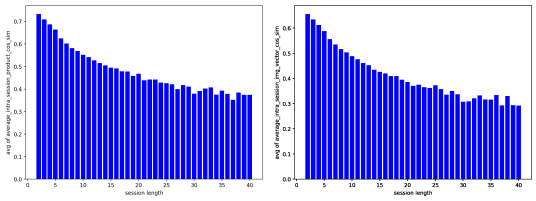

Figure 4: Avg. intra-session similarity of product description vectors (left) and product image vectors (right) per session length

B PREDICTION ANALYSIS

We present some predictions analyses that help to understand better models accuracy with respect to different characteristics of sessions and items to be predicted. So we used predictions over the validation set from the full ensemble, where the first half of each session is used for inference and the second half for computing metrics.

In Figure 5, we can observe that the F1 score is in general higher for longer sessions, which makes sense, as there are increased chances of the recommended items be present in the second half of the sessions. On the other hand, it is possible to observe that MRR, which measures the ability to predict the immediate next item, decreases for longer sessions. This might seem counterintuitive, as for longer sessions there is more contextual information available for recommendation. We can observe that popularity-bias (dark gray line) plays a role here, as first user interactions target more popular items. It might be the case that first session clicks come from popular products highlighted in the e-commerce home page, whereas later interactions might be more related to individual user interests. This phenomena has also been observed in session-based recommendation analyses for the news domain [26] (Figure 6.3).

We also believe that this finding can be somewhat related to Figure 4, where we see that in longer sessions users browsing might be less specific (lower similarity among interacted items), making it harder to predict accurately the next user move.

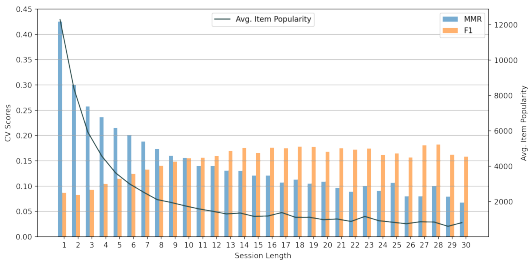

Figure 5: Accuracy (MRR and F1) x Session Length

In Figure 6, we plot the frequency of the top-50 most popular products (counting only the immediate next-item of the session) in the validation set and compare with the frequency of such items as the top prediction for a session. We can notice in general a pattern of

recommending popular items more often. The Pearson correlation between the actual and prediction item frequency is 0.9306 for the top-50 popular items and 0.8561 for all items.

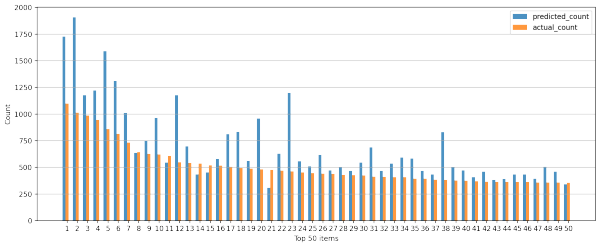

Figure 6: Frequency of popular items x frequency of top-1 predictions for such items

We specifically analyzed the MRR metric with respect to the popularity of the immediate next-item. It can be seen in Figure 7 that the more popular the product is, the higher its prediction accuracy. That might come from the fact that item embeddings of popular items have more chances to be trained and also because predicting very unpopular items is generally a wrong bet. Particularly, we observe a much lower accuracy (MRR=0.0009) for items whose frequency is between 1 and 4, which was one of the reasons we have included a variant of our pre-processing with item frequency capping (§4.3). We also observe a much higher MRR when next-items are products with more than 50 interactions. Finally, "New Items" refer to products which were not seen either during pre-training or fine-tuning, as they were present only in the hidden part (second half) of validation sessions. Predictions for such unseen items got MRR=0, suggesting that alternative approaches like content-based filtering or recommending newly launched products could be considered to deal initially with the *item-cold start problem*.

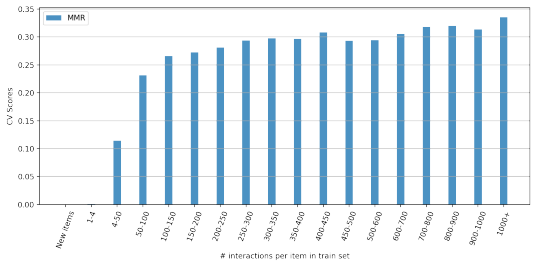

Figure 7: Accuracy (MRR) x next-item product popularity

Finally, in Figure 8 we analyze the MRR according to the product price (bucketed) of the immediate next-item, where 0 is the bucket for products without metadata information (price, and description/image vectors). We observe a pattern that for lower-priced products (buckets 1 to 4) the MRR is lower. Again, it might be an effect from popularity-bias (dark gray line), as more expensive products are more frequent in user interactions.

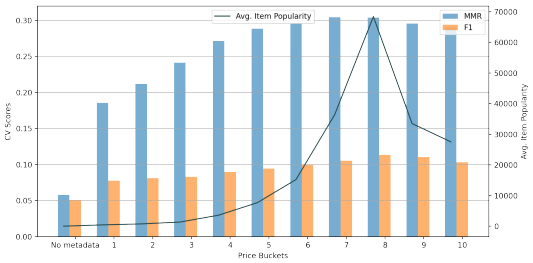

Figure 8: Accuracy (MRR) x next-item product price

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[2] David Ben-Shimon, Alexander Tsikinovsky, Michael Friedmann, Bracha Shapira, Lior Rokach, and Johannes Hoerle. 2015. Recsys challenge 2015 and the yoochoose dataset. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 357–358.

[3] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. 2018. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 46–54.

[4] Federico Bianchi, Bingqing Yu, and Jacopo Tagliabue. 2021. Bert goes shopping: Comparing distributional models for product representations. In *Proceedings of the Forth ECNLP at ACL-IJCNLP*.

[5] Wanyu Chen, Pengjie Ren, Fei Cai, Fei Sun, and Maarten de Rijke. 2020. Improving end-to-end sequential recommendations with intent-aware diversification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 175–184.

[6] Xusong Chen, Dong Liu, Chenyi Lei, Rui Li, Zheng-Jun Zha, and Zhiwei Xiong. 2019. BERT4SessRec: Content-Based Video Relevance Prediction with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 27th ACM International Conference on Multimedia*. 2597–2601.

[7] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

[8] Gabriel de Souza Pereira Moreira, Felipe Ferreira, and Adilson Marques da Cunha. 2018. News session-based recommendations using deep neural networks. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*. 15–23.

[9] 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, Volume 1*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186.

[10] Jun Fang. 2021. Session-based Recommendation with Self-Attention Networks. *arXiv preprint arXiv:2102.01922* (2021).

[11] P Moreira Gabriel De Souza, Dietmar Jannach, and Adilson Marques Da Cunha. 2019. Contextual hybrid session-based news recommendation with recurrent neural networks. *IEEE Access* 7 (2019), 169185–169203.

[12] Florent Garcin, Christos Dimitrakakis, and Boi Faltings. 2013. Personalized news recommendation with context trees. In *Proceedings of the 7th ACM Conference on Recommender Systems*. 105–112.

[13] Dmitri Goldenberg, Kostia Kofman, Pavel Levin, Sarai Mizrachi, Maayan Kafry, and Guy Nadav. 2021. Booking. com WSDM WebTour 2021 Challenge, In Proceedings of the Workshop on Web Tourism (WebTour) co-located with the 14th ACM International WSDM Conference (WSDM 2021). *CEUR Workshop Proceedings* Vol-2855.

[14] Qi He, Daxin Jiang, Zhen Liao, Steven CH Hoi, Kuiyu Chang, Ee-Peng Lim, and Hang Li. 2009. Web query recommendation via sequential query prediction. In *2009 IEEE 25th international conference on data engineering*. IEEE, 1443–1454.

[15] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 843–852.

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

[17] Hakan Inan, Khashayar Khosravi, and Richard Socher. 2016. Tying word vectors and word classifiers: A loss framework for language modeling. *arXiv preprint arXiv:1611.01462* (2016).

[18] Dietmar Jannach, Gabriel de Souza P. Moreira, and Even Oldridge. 2020. Why Are Deep Learning Models Not Consistently Winning Recommender Systems Competitions Yet? A Position Paper. In *Proceedings of the Recommender Systems Challenge 2020* (Virtual Event, Brazil) *(RecSysChallenge '20)*. Association for Computing Machinery, New York, NY, USA, 44–49.

[19] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.

[20] Komal Kapoor, Karthik Subbian, Jaideep Srivastava, and Paul Schrater. 2015. Just in time recommendations: Modeling the dynamics of boredom in activity streams.

In *Proceedings of the eighth ACM international conference on web search and data mining*. 233–242.

[21] Jing Lin, Weike Pan, and Zhong Ming. 2020. FISSA: fusing item similarity models with self-attention networks for sequential recommendation. In *Fourteenth ACM Conference on Recommender Systems*. 130–139.

[22] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction* 28, 4-5 (2018), 331–390.

[23] Anjing Luo, Pengpeng Zhao, Yanchi Liu, Fuzhen Zhuang, Deqing Wang, Jiajie Xu, Junhua Fang, and Victor S Sheng. [n.d.]. Collaborative Self-Attention Network for Session-based Recommendation. ([n. d.]).

[24] Ludewig Malte, Noemi Mauro, Latifi Sara, Jannach Dietmar, et al. 2020. Empirical analysis of session-based recommendation algorithms. (2020).

[25] Fei Mi and Boi Faltings. 2020. Memory Augmented Neural Model for Incremental Session-based Recommendation. *arXiv preprint arXiv:2005.01573* (2020).

[26] Gabriel de Souza Pereira Moreira. 2019. CHAMELEON: A Deep Learning Meta-Architecture for News Recommender Systems [Phd. Thesis]. *arXiv preprint arXiv:2001.04831* (2019).

[27] Gabriel de Souza P Moreira, Dietmar Jannach, and Adilson Marques da Cunha. 2019. On the importance of news content representation in hybrid neural session-based recommender systems, In Proceedings of the 7th International Workshop on News Recommendation and Analytics (INRA 2019). *CEUR Workshop Proceedings* Vol-2554.

[28] Zhiqiang Pan, Fei Cai, Yanxiang Ling, and Maarten de Rijke. 2020. Rethinking Item Importance in Session-based Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1837–1840.

[29] Ofir Press and Lior Wolf. 2017. Using the Output Embedding to Improve Language Models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, Valencia, Spain, 157–163. https://www.aclweb.org/anthology/E17-2025

[30] Benedikt Schifferer, Chris Deotte, Jean-Francois Puget, Gabriel de Souza Pereira Moreira, Gilberto Titericz, Jiwei Liu, and Ronay Ak. 2021. Using Deep Learning to Win the Booking.com WSDM WebTour21 Challenge on Sequential Recommendations, In Proceedings of the Workshop on Web Tourism (WebTour) co-located with the 14th ACM International WSDM Conference (WSDM 2021). *CEUR Workshop Proceedings* Vol-2855.

[31] 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.

[32] Shiming Sun, Yuanhe Tang, Zemei Dai, and Fu Zhou. 2019. Self-attention network for session-based recommendation with streaming data input. *IEEE Access* 7 (2019), 110499–110509.

[33] Jacopo Tagliabue, Ciro Greco, Jean-Francis Roy, Bingqing Yu, Patrick John Chia, Federico Bianchi, and Giovanni Cassani. 2021. SIGIR 2021 E-Commerce Workshop Data Challenge. *arXiv preprint arXiv:2104.09423* (2021).

[34] Jacopo Tagliabue and Bingqing Yu. 2021. Shopping in the Multiverse: A Counterfactual Approach to In-Session Attribution. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR eCom'20)*.

[35] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 565–573.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[37] Jingyi Wang, Qiang Liu, Zhaocheng Liu, and Shu Wu. 2019. Towards accurate and interpretable sequential prediction: A cnn & attention-based feature extractor. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1703–1712.

[38] Jianling Wang, Raphael Louca, Diane Hu, Caitlin Cellier, James Caverlee, and Liangjie Hong. 2020. Time to Shop for Valentine's Day: Shopping Occasions and Sequential Recommendation in E-commerce. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 645–653.

[39] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. 2020. SSE-PT: Sequential recommendation via personalized transformer. In *Fourteenth ACM Conference on Recommender Systems*. 328–337.

[40] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation.. In *IJCAI*, Vol. 19. 3940–3946.

[41] Zhilin Yang, Zihang Dai, Yiming Yang Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*.

[42] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 582–590.

[43] Shuai Zhang, Yi Tay, Lina Yao, Aixin Sun, and Jake An. 2019. Next item recommendation with self-attentive metric learning. In *Thirty-Third AAAI Conference on Artificial Intelligence*, Vol. 9.

[44] Yuanxing Zhang, Pengyu Zhao, Yushuo Guan, Lin Chen, Kaigui Bian, Lingyang Song, Bin Cui, and Xiaoming Li. 2020. Preference-aware mask for session-based recommendation with bidirectional transformer. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3412–3416.