Hierarchical Multi-Task Learning Framework for Session-based Recommendations

Sejoon Oh[1,*], Walid Shalaby[2], Amir Afsharinejad[2] and Xiquan Cui[2]

[1]Georgia Institute of Technology, Atlanta, Georgia,USA
[2]The Home Depot, Atlanta, Georgia, USA

Abstract

While session-based recommender systems (SBRSs) have shown superior recommendation performance, multi-task learning (MTL) has been adopted by SBRSs to enhance their prediction accuracy and generalizability further. Hierarchical MTL (H-MTL) sets a hierarchical structure between prediction tasks and feeds outputs from auxiliary tasks to main tasks. This hierarchy leads to richer input features for main tasks and higher interpretability of predictions, compared to existing MTL frameworks. However, the H-MTL framework has not been investigated in SBRSs yet. In this paper, we propose HierSRec which incorporates the H-MTL architecture into SBRSs. HierSRec encodes a given session with a metadata-aware Transformer and performs next-category prediction (i.e., auxiliary task) with the session encoding. Next, HierSRec conducts next-item prediction (i.e., main task) with the category prediction result and session encoding. For scalable inference, HierSRec creates a compact set of candidate items (e.g., 4% of total items) per test example using the category prediction. Experiments show that HierSRec outperforms existing SBRSs as per next-item prediction accuracy on two session-based recommendation datasets. The accuracy of HierSRec measured with the carefully-curated candidate items aligns with the accuracy of HierSRec calculated with all items, which validates the usefulness of our candidate generation scheme via H-MTL.

Keywords

Session-based Recommendation, Hierarchical Multi-task Learning

1. Introduction

Problem Description and Motivation. Multi-task learning (MTL) [1, 2, 3, 4, 5] has been employed to enhance the accuracy of existing recommender systems. MTL prevents the overfitting of a model via sharing parameters between multiple prediction tasks [6]. Hierarchical MTL (H-MTL) [7, 8, 9, 10] further improves the MTL by exploiting predictions from other tasks as another task's input in a hierarchical order. For example, the main task (e.g., next-item prediction) can use outputs from auxiliary tasks (e.g., next-category prediction) as input features to its prediction model. Those additional features can serve as rich external knowledge and enhance the performance of the main task.

ORSUM@ACM RecSys 2023: 6th Workshop on Online Recommender Systems and User Modeling, jointly with the 17th ACM Conference on Recommender Systems, September 19th, 2023, Singapore

*This work is based on the SO's internship project at The Home Depot.

soh337@gatech.edu (S. Oh); walid_shalaby@homedepot.com (W. Shalaby); amir_afsharinejad@homedepot.com (A. Afsharinejad); xiquan_cui@homedepot.com (X. Cui)

Table 1
Comparison of our proposed recommendation framework HIERSREC to existing session-based recommender systems.

	HIERSREC	NARM [11]	STAMP [12]	CSRM [13]	TAGNN [27]	COTREC [28]	M2TREC [29]
Session-based Recommendation	✓	✓	✓	✓	✓	✓	✓
Employing Item Metadata	✓						✓
Multi-task Learning	✓					✓	✓
Hierarchical Learning	✓			✓			
Candidate Generation for Test	✓						

While such an H-MTL framework gives an implicit data augmentation effect and higher generalization capability to a machine learning model [8, 10], its application to session-based recommender systems (SBRSs) [11, 12, 13, 14, 15, 16, 17, 18, 19] has not been investigated yet. SBRSs have gained attention as they capture the latest and evolving interests of a user in a session, where a session consists of a sequence of user-item interactions occurring within a short period [20]. The H-MTL architecture will be beneficial and important to SBRSs as per not only prediction accuracy [21] but also interpretability [22]. Let us assume a user is searching for a product to add to the current session in an e-commerce platform. With the H-MTL, we not only enhance the recommendation quality to users by leveraging prior knowledge obtained from auxiliary tasks but also provide meaningful and reasonable explanations of the recommendations to users by interpreting outputs from auxiliary tasks (e.g., top-K predicted categories or interaction types).

Challenges. Devising the H-MTL framework optimized for SBRSs is challenging for three major reasons. First, we need to define appropriate auxiliary tasks (e.g., category or interaction type predictions) related to the main task; in addition, we ought to set up proper hierarchical relationships between prediction tasks (e.g., a bottom-up approach from next-category prediction to next-item prediction). Second, rich and accurate session representations are required to ensure the high accuracy of the prediction tasks. The session features should contain item metadata information (e.g., categories) so that they can be used for auxiliary prediction tasks. Finally, it is computationally prohibitive to test the performance of H-MTL for SBRSs with millions of items available on online platforms (e.g., Amazon website). While existing methods [23, 24, 25] use randomly-sampled candidate items for the test, the sampled metrics can be inconsistent with the original performance measured with all items [26]. Thus, how can we generate high-quality candidate items to accurately evaluate the performance of H-MTL for SBRSs?

Proposed Method. To address the above challenges, we propose a novel recommendation model called HIERSREC which incorporates the H-MTL framework to SBRSs. HIERSREC is trained with multiple objectives of predicting the next item (main task) and next category (auxiliary task) in a session. We choose next-category predictions as our auxiliary task since category labels are mostly available in recommendation datasets (e.g., Amazon product review [30], Diginetica [31]). If categories labels are partially available or completely unavailable, we can cluster items to obtain implicit category information [32] or predict other metadata in a dataset such as interaction types (e.g., purchase, click, etc.) as auxiliary tasks. The first step of HIERSREC is generating a session representation using a metadata-aware Transformer [33] encoder. The Transformer encoder uses item IDs, item categories, and additional item metadata such as titles and descriptions to produce a precise and rich summary of the session. After that, HierSRec predicts the next category using the session encoding vector and transforms those category prediction results into embeddings. Next, we predict the next item in a session using the session representation and category prediction embeddings. Finally, the next-item and next-category prediction losses are combined together to train HierSRec. After training, we first perform the category prediction for all test instances, and we can generate high-quality candidate items for each test example by aggregating items belonging to top-K predicted categories.

Experiments. Thorough experiments on two large-scale E-commerce datasets show that HierSRec has superior next-item prediction performance to existing SBRSs by leveraging the hierarchical prediction framework. HierSRec shows at least 6.7% performance improvements as per three accuracy metrics compared to baselines. Moreover, HierSRec achieves comparable accuracy to that of HierSRec tested with full items by deliberately selecting a few items (e.g., 4% of total items) as ranking candidates using the category prediction. Ablation studies of HierSRec verify the effectiveness of each component of HierSRec.

Contributions. The main contributions of our paper are summarized as follows.

- To the best of our knowledge, this is the first work to leverage the H-MTL framework for SBRSs.
- We propose HierSRec that accurately predicts the next item in a session via employing the output of next-category prediction. HierSRec offers a compact set of candidate items of each test example for scalable ranking.
- Experiments on two recommendation datasets show that HierSRec outperforms existing SBRSs as per next-item prediction accuracy. We also confirm the effectiveness of our candidate generation method.

2. Related Work

Multi-Task Learning (MTL) & Hierarchical MTL. Recent research has shown that the generalizability and prediction performance of recommendation models can gain substantial improvements by using multi-task learning (MTL) [4, 5, 34]. In particular, MTL shares the knowledge learned from other related tasks with the main one, which has been shown to not only enhance the overall performance of the model, but also decrease the chance of overfitting and improve the quality of learned representations [6]. Hierarchical learning can improve the generalizability and interpretability of MTL by using the predictions of related tasks for another task. This architecture is called Hierarchical MTL (H-MTL). H-MTL has been utilized in Natural Language Processing [7, 8, 9, 35, 36, 37, 38] and Computer Vision [10, 37, 39, 40] domains to boost the performance of a model by sharing knowledge from lower-level tasks for more complex ones. In the context of recommender systems, Chen et al. [22] use H-MTL to improve the prediction accuracy and also provide a linguistic explanation of why a user likes/dislikes an item. Lim et al. [21] also utilize H-MTL to predict the Point-of-Interest (POI) a user will visit next.

Session-based Recommendation. Neural networks have served as the key component of the state-of-the-art SBRSs. Recurrent neural networks (RNNs) have been used to capture item

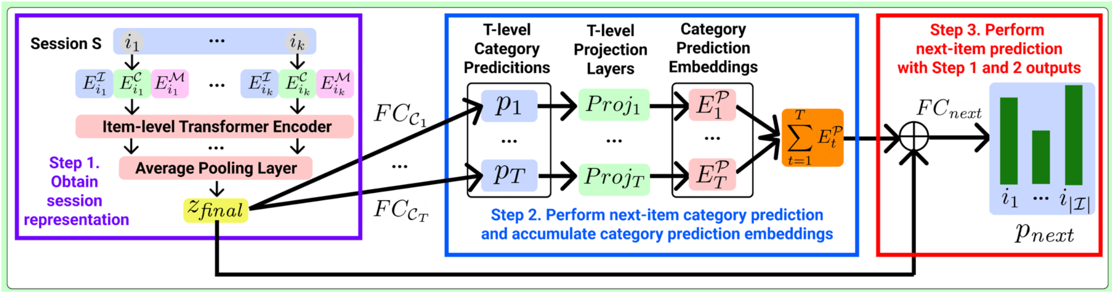


Figure 1: *Overview of* HIERSREC. Given a session and its observed items, HIERSREC first generates a session representation via a Transformer encoder, and it performs next-category and next-item predictions in a hierarchical manner using the session representation.

dependencies within sessions [11, 12, 13, 18, 19]. However, RNNs are limited in capturing longer dependencies across items. Thus, graph neural network-based approaches [15, 27, 28, 41, 42] and attention-based methods [11, 12, 13, 18, 19] have been proposed to incorporate such dependencies precisely into SBRSs. Transformer [33]-based approaches provide superior performance in predictions [43, 44, 45, 46]. MTL has been utilized to improve the accuracy of next-item prediction in SBRSs [47, 1, 2, 3, 29]. User intent prediction [48, 32, 49] is a well-known example of applying MTL to SBRSs. However, existing SBRSs are either designed only for next-item prediction or incompatible with the H-MTL architecture.

3. Proposed Approach: HIERSREC

Overview. As shown in Figure 1, our proposed recommendation model HIERSREC employs a Transformer [33] architecture to encode items in a session accurately and utilizes the obtained session representation for the ***hierarchical MTL*** [7, 8, 9, 10, 39]. We define tasks of HIERSREC as the next item prediction (main) and multi-level category predictions (auxiliary), where the category information is available on many recommendation datasets (e.g., Amazon product review [30], Diginetica [31]). Notice that our framework can be easily extended to other types of auxiliary tasks such as predicting user actions (e.g., click, add-to-cart, or purchase). Finally, we offer a candidate item generation scheme that uses the output from auxiliary tasks for scalable inference or evaluation.

Session Encoding with Metadata-aware Transformer. We use the Transformer encoder [33] to create an accurate representation of a user's interest within a session. Formally, given a session $\mathcal{S}$ with a sequence of k observed items $\{i_1, \ldots, i_k\}$, the first step is transforming the item sequence to the item representation sequence $\{E_{i_1}, \ldots, E_{i_k}\}$ using the item ID, the item category information, and the other item metadata such as titles and descriptions. Given an observed item $i_{pos}, \forall pos, 1 \leq pos \leq k$ in the current session $\mathcal{S}$, its embedding is constructed as follows.

$$E_{i_{pos}} = TransEnc(concat(E^{\mathcal{I}}_{i_{pos}}, E^{\mathcal{C}}_{i_{pos}}, E^{\mathcal{M}}_{i_{pos}})), \tag{1}$$

where $TransEnc$ and $concat$ indicate the Transformer encoder and embedding concatenation operation, and $E^{\mathcal{I}}$, $E^{\mathcal{C}}$, and $E^{\mathcal{M}}$ denote item ID embeddings, item category embeddings, and item metadata embeddings, respectively. Different encoders such as STAMP [12] can be used as $TransEnc$ in HierSRec, but our metadata-aware Trasnformer shows the best empirical performance. Given a set of items $\mathcal{I}$ and item ID embedding dimension $d_{\mathcal{I}}$, item ID embeddings $E^{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}| \times d_{\mathcal{I}}}$ map each item ID to the $d_{\mathcal{I}}$-dimensional feature. Assuming there are T-level item categories (e.g., 3-level categories for a *"dinner plate"* item are $\mathcal{C}_1$ - Kitchen; $\mathcal{C}_2$ - Tableware & Bar; $\mathcal{C}_3$ - Dinnerware), we create $d_{\mathcal{C}}$-dimensional embeddings for all category levels (i.e., $E^{\mathcal{C}_1}, \ldots, E^{\mathcal{C}_T}$) and sum up all category embeddings of an item to encode its category information, i.e., $E^{\mathcal{C}}_{i_{pos}} = E^{\mathcal{C}_1}_{i_{pos}} + \ldots + E^{\mathcal{C}_T}_{i_{pos}}$. Finally, item metadata embeddings $E^{\mathcal{M}} \in \mathbb{R}^{|\mathcal{I}| \times d_{\mathcal{M}}}$ are concatenations of trainable or pre-trained embeddings (e.g., from BERT [50] for texts and ResNet [51] for images) for each metadata of an item. For instance, the metadata embedding of an item i_{pos} can be derived as follows: $E^{\mathcal{M}}_{i_{pos}} = concat(E^{\text{title}}_{i_{pos}}, E^{\text{description}}_{i_{pos}}, E^{\text{image}}_{i_{pos}})$.

Finally, the item representation sequence $\{E_{i_1}, \ldots, E_{i_k}\}$ generated by the Transformer is fed to the average pooling layer to generate an accurate session representation z_{final} of a given session $\mathcal{S}$ (i.e., $z_{\mathit{final}} \in \mathbb{R}^{d_{\mathcal{I}}+d_{\mathcal{C}}+d_{\mathcal{M}}} = pooling(E_{i_1}, \ldots, E_{i_k})$). We choose the average pooling since it empirically shows the best next-item prediction performance compared to the trainable pooling or max pooling.

Hierarchical Multi-Task Learning (H-MTL). Given the final session representation z_{final} of a session $\mathcal{S} = \{i_1, \ldots, i_k\}$, a simple yet effective way to predict the next item i_{k+1} in $\mathcal{S}$ is employing a fully-connected layer to transform z_{final} to a next-item prediction score vector. However, this approach can easily make a model overfit the training data compared to multi-task learning (MTL) with sharing representations [6, 52, 53, 54].

While the MTL method can avoid the overfitting problem, it can be further enhanced by introducing a *"hierarchy"* or order between the prediction tasks, which is called hierarchical MTL (H-MTL) [7, 8, 9, 10, 39]. H-MTL models fully exploit the outputs from other tasks as *"additional knowledge"* via performing predictions of multiple tasks in a specific order or hierarchically (i.e., tree-structure), which offers the implicit data augmentation effect and the generalization capability to the main model.

To adapt H-MTL to SBRSs, we first predict categories of the next item (e.g., i_{k+1}) in a session $\mathcal{S} = \{i_1, \ldots, i_k\}$ and employ the category prediction results to enhance the item prediction. Specifically, we obtain T-level category prediction score vectors (i.e., $\{p_1, \ldots, p_T\}$) of the next item in a session using a session representation z_{final}, as shown below.

$$p_t = FC_{\mathcal{C}_t}(z_{\mathit{final}}) \in \mathbb{R}^{|\mathcal{C}_t|}, \forall t, 1 \leq t \leq T, \tag{2}$$

where FC indicates a fully-connected layer. Next, we transform category prediction vectors $\{p_1, \ldots, p_T\}$ to category prediction embeddings $\{E^{\mathcal{P}}_1, \ldots, E^{\mathcal{P}}_T\}$ via projection layers: $Proj_t \in \mathbb{R}^{|\mathcal{C}_t| \times (d_{\mathcal{I}}+d_{\mathcal{C}}+d_{\mathcal{M}})}, \forall t, 1 \leq t \leq T$, as shown below.

$$E^{\mathcal{P}}_t = Proj_t(p_t) \in \mathbb{R}^{(d_{\mathcal{I}}+d_{\mathcal{C}}+d_{\mathcal{M}})}, \forall t, 1 \leq t \leq T. \tag{3}$$

Finally, we sum the session representation z_{final} and all category prediction embeddings $\{E^{\mathcal{P}}_1, \ldots, E^{\mathcal{P}}_T\}$ and feed it to the fully-connected layer to generate the next-item prediction vector $p_{next} \in \mathbb{R}^{|\mathcal{I}|}$, as shown below.

$$p_{next} = FC_{next}(z_{\mathit{final}} + \lambda_C \sum_{t=1}^{T} E^{\mathcal{P}}_t), \tag{4}$$

where λ_C (a hyperparameter) controls the impact of category predictions on the next-item prediction. Category prediction results $\{p_1, \ldots, p_T\}$ can be used as implicit explanations of the next-item prediction p_{next} as the category prediction embeddings are used as a part of input features to the next-item predictor.

Loss Function and Optimization. Since HierSRec is a multi-task learning method, loss functions of multi-level next-category prediction and next-item prediction are combined and jointly optimized together. We use the Cross-Entropy loss for both category and item predictions. Assuming the ground-truth T-level categories of a next-item i_{k+1} we want to predict are $c_1, \ldots, c_T$, then a loss function of a level-t category prediction task is given as follows.

$$\mathcal{L}_{\mathcal{C}_t}(p_t) = -\sum_{i=1}^{|\mathcal{C}_t|} y_t(i) \log \left(Softmax(p_t)_i \right), \tag{5}$$

where y_t is a one-hot vector whose c_t^{th} value is 1, p_t is a level-t category prediction score vector, and $Softmax(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$. Similarly, the next-item prediction loss is given as follows.

$$\mathcal{L}_{next}(p_{next}) = -\sum_{i=1}^{|\mathcal{I}|} y(i) \log \left(Softmax(p_{next})_i \right), \tag{6}$$

where y is a one-hot vector whose i_{k+1}^{th} value is 1, and p_{next} is a next-item prediction score vector. The combined loss function for our proposed hierarchical multi-task learning is given as follows.

$$\mathcal{L}_{final} = \mathcal{L}_{next} + \lambda \sum_{t=1}^{T} \mathcal{L}_{\mathcal{C}_t}, \tag{7}$$

where λ is the importance weight of category prediction tasks. We tested several weighting strategies for λ and λ_C such as trainable weights or randomized weights per epoch, but **setting** $\lambda = \lambda_C = 1.0$ **shows the best prediction performance empirically**. We optimize the above loss function (7) with Adam [55] optimizer for all training data.

Candidate Item Generation for Scalable Evaluation. During the test (or inference) stage, we use the next-item prediction vector $p_{next} \in \mathbb{R}^{|\mathcal{I}|}$ generated from the trained model. Computing p_{next} can be computationally expensive on large-scale recommendation datasets (e.g., e-commerce domain) with millions of items. Thus, for practicality, existing algorithms [23, 24, 25] uses a small set of candidate items instead of all items during the inference. However, those methods use randomly-sampled candidates, and the accuracy measured with such candidates can be significantly different from the accuracy calculated with full items [26]. Thus, we propose a more accurate candidate generation method that leverages the category prediction result. First, given a test session with observed items, we conduct the category prediction and obtain the score vectors $\{p_1, \ldots, p_T\}$. We sample top-K (K: hyperparameter) categories $\{c_1^t, \ldots, c_K^t\}$ from each level-t category prediction p_t and construct a candidate item set $\mathcal{I}' \subset \mathcal{I}$ by the following.

$$\mathcal{I}' = \{i \mid i \in c_k^1, \forall k, 1 \le k \le K\} \cup \cdots \cup \{i \mid i \in c_k^T, \forall k, 1 \le k \le K\}$$

We empirically verify that our candidate selection policy can achieve nearly equivalent accuracy to that of an original policy that uses all items during the inference (refer to Figure 2 later).

Table 2
Summary of datasets and sessions used for experiments. M: million, K: thousand. N/A: hidden due to the company's policy.

Name	Training sessions	Test sessions	Items	Interactions	Session Length	Category	Item Metadata
The Home Depot (THD) (E-commerce)	2.8M	85K	259K	N/A	N/A	N/A	Title, Description, Category, etc.
Diginetica (E-commerce)	191K	16K	117K	880K	4.60	Level-1: 1.2K	Title, Category

Table 3
Performance of HierSRec in terms of predicting the next item in a session on the Diginetica and THD datasets, compared to baseline SBRSs (**Bold** indicates the best model, while the second-best model is underlined). HierSRec shows the best prediction performance among all methods across two datasets, with statistical significance (P-values from one-tailed t-test are ≤ 0.05).

Dataset	Diginetica			The Home Depot		
Method	MRR@20	HITS@20	Recall@20	MRR@20	HITS@20	Recall@20
NARM [11]	0.0751	<u>0.2993</u>	<u>0.4621</u>	0.1036	<u>0.2792</u>	<u>0.4163</u>
STAMP [12]	0.0717	0.2545	0.4161	<u>0.1086</u>	0.2718	0.3996
CSRM [13]	0.0730	0.2804	0.4363	0.0968	0.2608	0.3919
TAGNN [27]	0.0785	0.2643	0.4153	0.0865	0.2188	0.3507
COTREC [28]	<u>0.0787</u>	0.2837	0.4306	0.0625	0.2271	0.3518
HierSRec	**0.0990**	**0.3347**	**0.5031**	**0.1245**	**0.3019**	**0.4443**
% Improvement	25.8%	11.8%	8.9%	14.6%	8.1%	6.7%

What If Category Information Is Unavailable? While item category information is available in most recommendation datasets, it might be partially available or completely unavailable in a few cases. To address it, we can find implicit categories of items by utilizing graph neural networks and clustering [32]. For instance, we apply an off-the-shelf node embedding algorithm on a session-item bipartite graph to obtain item representations. Applying a clustering method (e.g., K-means) on obtained item embeddings will generate clusters of items, which will approximate the category information. Another solution is employing other metadata (e.g., interaction type) for auxiliary tasks of the H-MTL.

4. Experimental Evaluations of HierSRec

Datasets. Table 2 lists the statistics of the datasets. The Home Depot (THD) is an E-commerce dataset obtained from a large online retailer THD. The dataset is composed of Add-to-Cart (ATC) events within millions of online sessions. The dataset has rich product metadata including 7 attributes: product title, 3-level categories, brand, manufacturer, color, department name, and class name. We exclude certain information from the THD dataset according to the company's policy. For the THD dataset, we filter users and items with less than 10 interactions. Diginetica[1]

[1]https://competitions.codalab.org/competitions/11161

is a **public E-commerce dataset** that was a part of CIKM Cup 2016 challenge. We did pre-process of the Diginetica similar to [15].
Baselines. We use the following state-of-the-art session-based recommenders:[2] (1) **NARM** [11]: An attention-based model that employs a hybrid encoder to reflect a user's global and local interests with an attention mechanism, (2) **STAMP** [12]: An attention/memory-based model that incorporates a user's short-term and long-term interests via short-term attention and long-term memory modules, respectively, (3) **CSRM** [13]: a session-based recommendation model that contextualizes the current and neighborhood sessions with inner and outer memory encoders, respectively, (4) **TAGNN** [27]: a graph neural network (GNN)-based session-based recommender that utilizes a target-aware attention module for predictions, and (5) **COTREC** [28]: a state-of-the-art GNN-based recommendation model that combines self-supervised learning with graph co-training. We exclude several models including nearest-neighbor algorithms if they show similar or worse performance compared to our existing baselines.
Hyperparameters. Hyperparameters of HierSRec and baseline methods are found by **extensive grid search** using a validation set (randomly sampled 10% from training). Specifically, $d_{\mathcal{I}} = d_{\mathcal{C}} = d_{\mathcal{M}} = 128$, $\lambda = \lambda_C = 1.0$, and batch size and learning rates are set to 1024 and 0.0001, respectively. We use all items as candidates during the test by default. HierSRec also uses 2 layers of Transformer Encoder with 8 attention heads.
Reproducibility. While the code of HierSRec and the THD dataset cannot be released due to the company policy, we release the public dataset (Diginetica) and baseline implementations used in the paper.
Next-item Prediction Accuracy of HierSRec. To verify the effectiveness of HierSRec, we measure the next-item prediction accuracy of HierSRec and baselines on diverse datasets, with respect to three accuracy metrics: Mean Reciprocal Rank@20 (MRR) [56], HITS@20, and Recall@20. HITS@20 counts only ground-truth next-items in top-20 lists, while Recall@20 counts all future items (including the next-item) in a session in top-20 lists.

Table 3 shows the next-item prediction accuracy of HierSRec and baselines on the Diginetica and THD datasets. HierSRec shows the best performance as per all metrics among all methods across all datasets, with statistical significance (P-values from one-tailed t-test are ≤ 0.05). Relative performance improvements of HierSRec compared to the best baseline are $6.7\% - 25.8\%$. The high performance of HierSRec is due to **the hierarchical learning architecture, not additional item metadata** (compare the first and last row in Table 4b). In other words, the key reason for these performance improvements is incorporating prior knowledge from the next-category prediction into the next-item prediction, so that we can filter out items associated with irrelevant categories easily while predicting the next item.
Next-category Prediction Accuracy of HierSRec. We test how accurately HierSRec can predict the next category in a session on the Diginetica dataset. As shown in Table 4a, HierSRec outperforms the heuristic and shows almost the same accuracy as an MTL variant of HierSRec without hierarchical learning. It is expected since the category prediction does not take any additional feature from the next-item prediction task due to its lower hierarchy, and enhancing the next-category prediction accuracy is not the main goal of HierSRec.
Ablation Study of HierSRec. We conduct the ablation study of HierSRec to show how

[2]We used open-source implementations of baseline algorithms (https://github.com/rn5l/session-rec).

Table 4
Category prediction result (left) and ablation study (right) of HierSRec on the Diginetica dataset. Both results substantiate the usefulness of the proposed H-MTL framework used in HierSRec for session-based recommendations.

a Category prediction result of HierSRec.

Models / Metrics	MRR @20	HITS @20	Recall @20
Heuristic (recommends historical categories)	0.7861	0.8793	0.8908
Multi-task learning (no hierarchical predictions)	**0.8803**	**0.9338**	**0.9445**
HierSRec (proposed)	<u>0.8747</u>	<u>0.9291</u>	<u>0.9396</u>

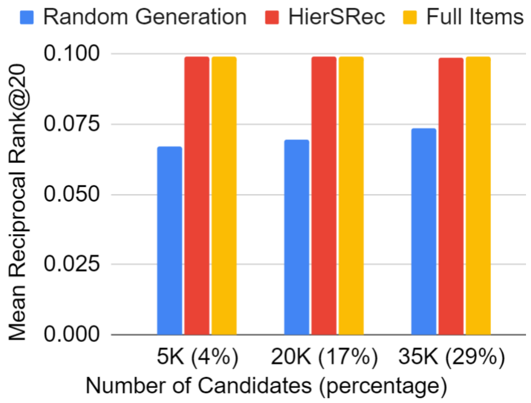
b Ablation study of HierSRec.

Models / Metrics	MRR @20	HITS @20	Recall @20
Single-task learning (only next-item predictions)	0.0752	0.2524	0.4037
Multi-task learning (no hierarchical predictions)	<u>0.0842</u>	<u>0.2991</u>	<u>0.4689</u>
HierSRec (proposed)	**0.0990**	**0.3347**	**0.5031**

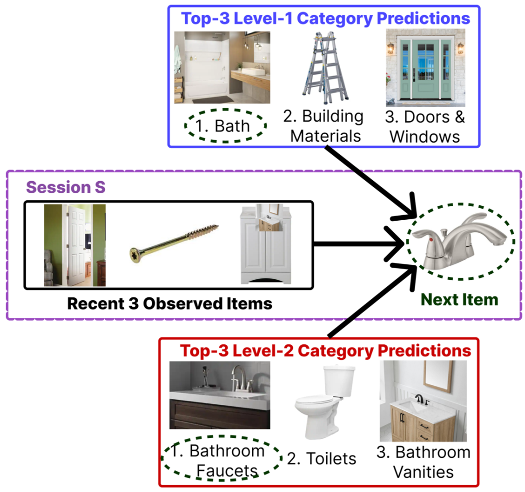
effective the hierarchical learning of HierSRec is for session-based recommendations. We create two variants of HierSRec, where the first one only performs next-item predictions **only with metadata-aware Transformer (no MTL)**, and the second one employs normal MTL architecture without hierarchical predictions. Table 4b shows the ablation study result of HierSRec on the THD dataset. As we can notice, HierSRec exhibits the highest accuracy compared to the two variants, with at least 7.3% relative performance improvements and statistical significance. This result shows that our proposed hierarchical MTL architecture induces a higher generalization capability of a model and an implicit data augmentation effect.

Verification of Candidate Items Generated by HierSRec. We confirm the quality of candidate items generated by HierSRec by comparing the accuracy measured with our candidates, random candidates, and full items. Figure 2a shows the MRR@20 metric of HierSRec calculated with three different candidate generation policies on the Diginetica dataset. Using the category prediction knowledge, HierSRec can create a small candidate set (e.g., 4% of total items) consisting of key items that are highly related to the ground-truth next item in a session. However, the random candidate policy exhibits poor performance as it cannot selectively choose important items as well as the ground-truth next item.

Case Study: Hierarchical Predictions of HierSRec. Figure 2b is a case study result on the



a Next-item prediction accuracy of HierSRec as per the number of candidates used for test.



b Case-study result of HierSRec on the The Home Depot dataset.

Figure 2: *Candidate generation verification (left) and case-study (right) results of HierSRec.* The left figure shows that accuracy measured with candidates generated by HierSRec is close to accuracy measured with full items in a dataset. The right figure indicates that the output of auxiliary tasks (e.g., Bath and Bathroom Faucets categories) can be crucial input features to the next-item prediction.

THD dataset of how HierSRec utilizes category predictions to improve the next-item predictions. Top-K category prediction results of HierSRec include diverse and evolving preferences of a user in a session (e.g., Door $\longrightarrow$ Bathroom) and identify the most important categories (e.g., Bath and Bathroom Faucets) for next-item predictions. Providing these meaningful knowledge from category predictions will make next-item recommendations personalized and accurate.

5. Conclusion

In this paper, we proposed a novel session-based recommendation model HierSRec that employs a metadata-aware Transformer encoder and a hierarchical multi-task learning framework to obtain higher model generalizability. Future works of HierSRec include that (1) more complex hierarchical structures (e.g., tree-shape) between various tasks (e.g., next-item, next-action, and next-category predictions) can be explored, and (2) extending HierSRec to predict next items accurately in sessions with cold-start items or only a few items by employing their metadata.

References

[1] N. Qiu, B. Gao, F. Huang, H. Tu, W. Luo, Incorporating global context into multi-task learning for session-based recommendation, in: International Conference on Knowledge Science, Engineering and Management, Springer, 2021, pp. 627–638.

[2] C. Huang, J. Chen, L. Xia, Y. Xu, P. Dai, Y. Chen, L. Bo, J. Zhao, J. X. Huang, Graph-enhanced

multi-task learning of multi-level transition dynamics for session-based recommendation, in: AAAI Conference on Artificial Intelligence (AAAI), 2021.

[3] W. Meng, D. Yang, Y. Xiao, Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1091–1100.

[4] C. Gao, X. He, D. Gan, X. Chen, F. Feng, Y. Li, T.-S. Chua, D. Jin, Neural multi-task recommendation from multi-behavior data, in: 2019 IEEE 35th international conference on data engineering (ICDE), IEEE, 2019, pp. 1554–1557.

[5] G. Hadash, O. S. Shalom, R. Osadchy, Rank and rate: multi-task learning for recommender systems, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 451–454.

[6] S. Ruder, An overview of multi-task learning in deep neural networks, arXiv preprint arXiv:1706.05098 (2017).

[7] V. Sanh, T. Wolf, S. Ruder, A hierarchical multi-task approach for learning embeddings from semantic tasks, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 6949–6956.

[8] X. Wang, G. Xu, Z. Zhang, L. Jin, X. Sun, End-to-end aspect-based sentiment analysis with hierarchical multi-task learning, Neurocomputing 455 (2021) 178–188.

[9] B. Tian, Y. Zhang, J. Wang, C. Xing, Hierarchical inter-attention network for document classification with multi-task learning., in: IJCAI, 2019, pp. 3569–3575.

[10] H. Park, H. Bharadhwaj, B. Y. Lim, Hierarchical multi-task learning for healthy drink classification, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[11] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, J. Ma, Neural attentive session-based recommendation, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 1419–1428.

[12] Q. Liu, Y. Zeng, R. Mokhosi, H. Zhang, Stamp: short-term attention/memory priority model for session-based recommendation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1831–1839.

[13] M. Wang, P. Ren, L. Mei, Z. Chen, J. Ma, M. de Rijke, A collaborative session-based recommendation approach with parallel memory modules, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 345–354.

[14] C. Hansen, C. Hansen, L. Maystre, R. Mehrotra, B. Brost, F. Tomasi, M. Lalmas, Contextual and sequential user embeddings for large-scale music recommendation, in: ACM RecSys, 2020, pp. 53–62.

[15] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, T. Tan, Session-based recommendation with graph neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 346–353.

[16] B. Hidasi, A. Karatzoglou, Recurrent neural networks with top-k gains for session-based recommendations, in: Proceedings of the 27th ACM international conference on information and knowledge management, 2018, pp. 843–852.

[17] J. You, Y. Wang, A. Pal, P. Eksombatchai, C. Rosenburg, J. Leskovec, Hierarchical temporal

convolutional networks for dynamic recommender systems, in: The world wide web conference, 2019, pp. 2236–2246.
[18] Z. Pan, F. Cai, Y. Ling, M. de Rijke, An intent-guided collaborative machine for session-based recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1833–1836.
[19] P. Ren, Z. Chen, J. Li, Z. Ren, J. Ma, M. De Rijke, Repeatnet: A repeat aware neural recommendation machine for session-based recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 4806–4813.
[20] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. Orgun, D. Lian, A survey on session-based recommender systems, arXiv preprint arXiv:1902.04864 (2019).
[21] N. Lim, B. Hooi, S.-K. Ng, Y. L. Goh, R. Weng, R. Tan, Hierarchical multi-task graph recurrent network for next poi recommendation, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022.
[22] Z. Chen, X. Wang, X. Xie, T. Wu, G. Bu, Y. Wang, E. Chen, Co-attentive multi-task learning for explainable recommendation., in: IJCAI, 2019, pp. 2137–2143.
[23] J. Li, Y. Wang, J. McAuley, Time interval aware self-attention for sequential recommendation, in: Proceedings of the 13th international conference on web search and data mining, 2020, pp. 322–330.
[24] W.-C. Kang, J. McAuley, Self-attentive sequential recommendation, in: 2018 IEEE international conference on data mining (ICDM), IEEE, 2018, pp. 197–206.
[25] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, P. Jiang, Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, in: Proceedings of the 28th ACM international conference on information and knowledge management, 2019, pp. 1441–1450.
[26] W. Krichene, S. Rendle, On sampled metrics for item recommendation, in: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, 2020, pp. 1748–1757.
[27] F. Yu, Y. Zhu, Q. Liu, S. Wu, L. Wang, T. Tan, Tagnn: Target attentive graph neural networks for session-based recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1921–1924.
[28] X. Xia, H. Yin, J. Yu, Y. Shao, L. Cui, Self-Supervised Graph Co-Training for Session-based Recommendation, in: 30th ACM International Conference on Information and Knowledge Management (CIKM 2021), ACM, 2021.
[29] W. Shalaby, S. Oh, A. Afsharinejad, S. Kumar, X. Cui, M2trec: Metadata-aware multi-task transformer for large-scale and cold-start free session-based recommendations, in: Proceedings of the 16th ACM Conference on Recommender Systems, 2022, pp. 573–578.
[30] J. Ni, J. Li, J. McAuley, Justifying recommendations using distantly-labeled reviews and fine-grained aspects, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 188–197.
[31] Diginetica dataset for cikm cup 2016 challenge, https://competitions.codalab.org/competitions/11161, 2016.
[32] S. Oh, A. Bhardwaj, J. Han, S. Kim, R. A. Rossi, S. Kumar, Implicit session contexts for next-item recommendations, in: Proceedings of the 31st ACM International Conference

on Information & Knowledge Management, 2022, pp. 4364–4368.
[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
[34] R. Caruana, Multitask learning, Machine Learning 28 (1997). doi:10.1023/A:1007379606734.
[35] Y. Farag, H. Yannakoudakis, Multi-task learning for coherence modeling, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics (ACL), 2019, pp. 629–639.
[36] M. Song, Y. Zhao, Enhance rnnlms with hierarchical multi-task learning for asr, in: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, pp. 6102–6106. doi:10.1109/ICASSP43922.2022.9747525.
[37] D.-K. Nguyen, T. Okatani, Multi-task learning of hierarchical vision-language representation, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 10484–10493.
[38] W. Song, Z. Song, L. Liu, R. Fu, Hierarchical multi-task learning for organization evaluation of argumentative student essays, in: C. Bessiere (Ed.), Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, International Joint Conferences on Artificial Intelligence Organization, 2020, pp. 3875–3881. URL: https://doi.org/10.24963/ijcai.2020/536. doi:10.24963/ijcai.2020/536.
[39] J. Fan, T. Zhao, Z. Kuang, Y. Zheng, J. Zhang, J. Yu, J. Peng, Hd-mtl: Hierarchical deep multi-task learning for large-scale visual recognition, IEEE transactions on image processing 26 (2017) 1923–1938.
[40] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, Y. Yu, Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 2740–2748.
[41] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, X. Zhou, Graph contextualized self-attention network for session-based recommendation., in: IJCAI, volume 19, 2019, pp. 3940–3946.
[42] Z. Wang, W. Wei, G. Cong, X.-L. Li, X.-L. Mao, M. Qiu, Global context enhanced graph neural networks for session-based recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 169–178.
[43] G. de Souza Pereira Moreira, S. Rabhi, J. M. Lee, R. Ak, E. Oldridge, Transformers4rec: Bridging the gap between nlp and sequential/session-based recommendation, in: Fifteenth ACM Conference on Recommender Systems, 2021, pp. 143–153.
[44] G. de Souza Pereira Moreira, S. Rabhi, R. Ak, B. Schifferer, End-to-end session-based recommendation on gpu, in: Fifteenth ACM Conference on Recommender Systems, 2021, pp. 831–833.
[45] X. Chen, D. Liu, C. Lei, R. Li, Z.-J. Zha, Z. Xiong, Bert4sessrec: Content-based video relevance prediction with bidirectional encoder representations from transformer, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 2597–2601.
[46] G. d. S. P. Moreira, S. Rabhi, R. Ak, M. Y. Kabir, E. Oldridge, Transformers with multi-modal

features and post-fusion context for e-commerce session-based recommendation, arXiv preprint arXiv:2107.05124 (2021).
[47] M. Tavakol, U. Brefeld, Factored mdps for detecting topics of user sessions, in: Proceedings of the 8th ACM Conference on Recommender Systems, 2014, pp. 33–40.
[48] J. Guo, Y. Yang, X. Song, Y. Zhang, Y. Wang, J. Bai, Y. Zhang, Learning multi-granularity consecutive user intent unit for session-based recommendation, in: Proceedings of the fifteenth ACM International conference on web search and data mining, 2022, pp. 343–352.
[49] Z. Liu, H. Chen, F. Sun, X. Xie, J. Gao, B. Ding, Y. Shen, Intent preference decoupling for user representation on online recommender system, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 2575–2582.
[50] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, Association for Computational Linguistics, 2019, pp. 4171–4186.
[51] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 770–778.
[52] S. Chowdhuri, T. Pankaj, K. Zipser, Multinet: Multi-modal multi-task learning for autonomous driving, in: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2019, pp. 1496–1504.
[53] X. Wang, C. Zhang, Z. Zhang, Boosted multi-task learning for face verification with applications to web image and video search, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 142–149.
[54] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Computing Surveys (CSUR) 52 (2019) 1–38.
[55] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations (ICLR), 2015.
[56] E. M. Voorhees, et al., The trec-8 question answering track report., in: Text Retrieval Conference, volume 99, 1999, pp. 77–82.