Hypergraph Enhanced Knowledge Tree Prompt Learning for Next-Basket Recommendation

Zifeng Mai[1,2,3], Jianyang Zhai[1,4], Naiqing Li[1], Chang-Dong Wang[1,2,3]✉, Zhongjie Zeng[5], Ya Li[6], and Jiaquan Chen[5]

[1] School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, Guangdong, China
[2] Guangxi Key Laboratory of Digital Infrastructure, Guangxi Zhuang Autonomous Region Information Center, Nanning 530000, China
[3] Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, 510006, China
[4] Pengcheng Laboratory, Shenzhen, Guangdong, China
[5] Wens Foodstuff Group Co., Ltd., Yunfu, Guangdong, China.
[6] School of Electronics and Information, Guangdong Polytechnic Normal University, Guangzhou, Guangdong, China

maizf3@mail2.sysu.edu.cn, zhaijy01@pcl.ac.cn, adslnq@mail.sysu.edu.cn, changdongwang@hotmail.com, zengzhjgdut@gmail.com, liya2829@gpnu.edu.cn, 18998812918@189.cn

Abstract. Next-basket recommendation (NBR) aims to infer the next basket given a basket sequence. The existing NBR methods rely solely on item IDs and ignore the semantic information of items. What's more, these methods only consider binary item relationships which are often in higher order in the NBR scenario. In this paper, we propose HEKP, which addresses these challenges by pretrained language model (PLM) and hypergraph. Specifically, we use PLM to encode the basket sequence by masked user prompt (MUP). However, PLM-based recommendation will degrade when encountering Out-Of-Vocabulary (OOV) items. To tackle this issue, we construct another knowledge tree prompt (KTP) as the explanation of those OOV items. Additionally, we design a multi-item relation encoder to model the high order correlations among items by building a hypergraph based on item similarities. Lastly, we design a frequency based gating module to recommend the next basket. Extensive experiments are conducted on HEKP on three datasets, and the results validate its effectiveness against state-of-the-art methods.

Keywords: Next-Basket Recommendation · Pretrained Language Model · Knowledge Graph.

1 Introduction

With the explosive growth in e-commerce, recommendation has become a potential research field. One of the most popular topics is next-basket recommendation (NBR), which groups multiple items that a user interacted at the same time into

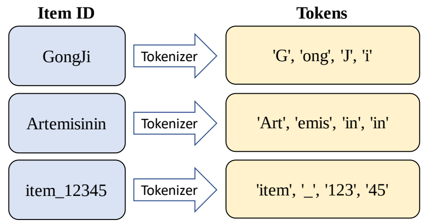


Fig. 1: Example of OOV item IDs. These IDs become irrelevant tokens after tokenization. We use T5-small [16] tokenizer in this example.

a basket and aims to infer the next basket given an interacted basket sequence. Compared with sequential recommendation (SR), NBR has a wider application in real world since SR assumes that user interactions must strictly follow a chronological order. Maintaining this assumption is difficult because users often interact with multiple items simultaneously to satisfy a certain need.

Many of the existing NBR methods [2, 14, 23] focus on modeling long- and short-term transition patterns in the basket sequences with carefully designed sequential models, aiming to learn an better representation for each item from the basket sequence to which it belongs. However, these methods rely solely on item IDs and ignore the semantic information of the items themselves. Recently, PLMs [3, 5, 12, 16] have shown remarkable performance in multiple natural language processing (NLP) tasks, especially in natural language understanding (NLU), exhibiting outstanding capability in semantic capturing and understanding. Therefore, PLMs are suitable for mining semantic-level correlations between items, which is difficult for the traditional methods. Following the prompt learning paradigm, many PLM-based recommendation (PLM-Rec) methods [4, 17, 21, 29, 30] transform users, items and their interactions into texts using manually designed prompts and make recommendations either generatively or discriminatively.

However, one of the biggest challenges in PLM-Rec is how to represent items in prompts. Since the PLM corpus cannot be infinite, it often struggles to understand the Out-Of-Vocabulary (OOV) items that do not exist in its corpus. There are mainly two ways to represent an item in existing methods. The first way is to use its original name directly, but many OOV items are either terminologies (such as) or non-English words (such as `GongJi`). The other way is to map each item to a unique integer as its item ID (such as `item_12345`). As Fig. 1 shows, neither of these two ways is intelligible to PLMs. For example, `Artemisinin` has nothing to do with the token `Art`. The token `123` is not semantically related to the following token `45`.

With the help of various sequential models or the PLM, we are able to estimate the preferences for users through the basket sequences. However, capturing the collaborative information among users, baskets and items is also significant for NBR, and this is exactly where graph-based NBR methods [13, 19, 20, 28] excel. These methods organize the user-item interactions as a graph and attempt to mine collaborative signals in the graph using GNNs. Despite of the remarkable improvements, they still face the following challenges:

- These methods only consider the binary relationship between items, ignoring that item dependencies often go beyond binary and are often triadic or even higher order in the NBR scenario since users tend to purchase multiple items at the same time. Therefore, hypergraphs are more appropriate to describe such high order dependencies.
- These methods propose to learn item embeddings directly on the user-item graph or the basket-item graph, which is noisy and unreliable. Taking the well-known story of beer and diapers as example, a man goes to a grocery and buys some diapers for his baby and some beer for himself. In this case, beer and diapers are connected by this man and thus they will interfere with each other during the message passing process.

In this paper, we address the above challenges by proposing **H**ypergraph **E**nhanced **K**nowledge Tree **P**rompt Learning for Next-Basket Recommendation (HEKP). For the OOV challenge, we design a knowledge tree prompt (KTP) by building a knowledge tree from KG. The idea of the KTP can be analogized to the learning process of natural human beings. When someone encounters an unfamiliar word while reading, it is better to look up the explanation of that word in the dictionary to understand it, rather than forcing himself to remember the word. KTP can be seen as a natural language explanation for the OOV items in the basket sequence. Besides, we also design a Maksed User Prompt (MUP) following [29] to reformulate the NBR task to the pretraining task of the PLM. For the challenge in graph-based NBR, we design a multi-item relation encoder. First, we use a Mixture-of-Expert (MoE) model to learn the pairwise similarity between items, which are constrained by two ranking loss of different levels. Then, an item hypergraph is built by based on the similarity. We employ an hypergraph convolutional module to model the high order correlations among multiple items. Additionally, we propose a simple yet effective frequency based gating (FBG) module to predict the next basket. Extensive experiments show that our method outperforms the state-of-the-art methods on three datasets. The contributions of this paper can be summarized as follows:

- We construct a new type of prompt KTP to provide explanation for the OOV items, encouraging the PLM to understand rather than remember.
- We design a multi-item relation encoder to mine the high order correlations among multiple items. Specifically, we employ convolution on a hypergraph which is built based on item similaries.
- We propose HEKP in this paper, which is a paradigm for PLM-based Next-Basket Recommendation. Extensive experiments are conducted on three datasets and the results demonstrate the effectiveness of our method.

2 Related Work

2.1 Next-Basket Recommendation

The existing NBR methods can be broadly divided into two classes. The first class is sequence-based methods. Early works [18] utilize Markov Chain (MC) to model the transaction between two nearby baskets. With the development of deep learning, RNN-based methods such as DREAM [27], CLEA [15] and Beacon [11] apply RNNs to capture the long-term dependency in the basket sequence. Similarly, some attention-based methods, such as IneNet [23] and Intention2basket [24], adaptively mine the user preferences to different items by the attention mechanism. These methods rely solely on item IDs and lack the ability to mine deeper semantic-level correlations between items.

The second class is graph-based methods. Using graph to model the user-item interactions has long been a general and popular method due to their intrinsic consistency. MMNR [2] aggregates item representations within baskets from multiple aspects and multiple views. HapCL [20] mines information from multiple views with polar contrastive learning. These methods only model point-to-point binary item relationships, while in NBR scenarios, item dependencies are often in higher order. To solve this problem, DigBot [13] and BRL [28] constructs hypergraphs by connecting items in a basket by a hyperedge.

Some recent work, such as M^2 [14], Sets2Sets [7] and TIFU-KNN [8] has also suggested that the user interactions in NBR show stronger repeat patterns than those in SR. Therefore, they explicitly model the frequency of each user interacting with different items to enhance the recommendation.

2.2 PLM for Recommendation

A large number of PLM-Rec methods have been proposed recently. For example, UniSRec [6] use BERT [3] as a sequential encoder to obtain universal sequence representation. TALLRec [1] proposes a lightweight tuning strategy to align various PLMs with recommendation. P5 [4] is a unified framework that integrates various recommendation tasks by reformulating them into NLP tasks. Following P5, KP4SR [29] further incorporates KG into prompt and block attention between irrelevant triplets. GenRec [21] auto-regressively generates each item in the next basket one by one to simulate the process of users satisfying their needs step by step. P5-ID [9] discusses multiple ways to assign each item with a unique ID. TIGER [17] quantities item content into semantic ID and trains a Transformer model from scratch. LC-Rec [30] designs a series of tuning tasks to enhance the integration of collaborative semantics in LLMs.

3 Methodology

3.1 Problem Formulation and Preliminary

Let $\mathcal{B} = \left\{b_1, \ldots, b_{|\mathcal{B}|}\right\}$ denotes the set of all baskets. The historical interaction sequence of a user is a list of basket arranged in chronological order, denoted by $\mathcal{S}^u = \left[b_1^u, \ldots, b_{|\mathcal{S}^u|}^u\right]$, where $b_j^u \in \mathcal{B}$. The NBR task is to infer the items in the next basket $b_{|\mathcal{S}^u|+1}^u$ that user u may interact with given the basket sequence $\mathcal{S}^u$.

Here, we briefly introduce hypergraph. A hypergraph $\mathcal{HG} = (\mathcal{V}, \mathcal{E}_{hyper})$ consists of a vertex set $\mathcal{V}$ and a hyperedge set $\mathcal{E}_{hyper}$. Each hyperedge $e \in \mathcal{E}_{hyper}$ is

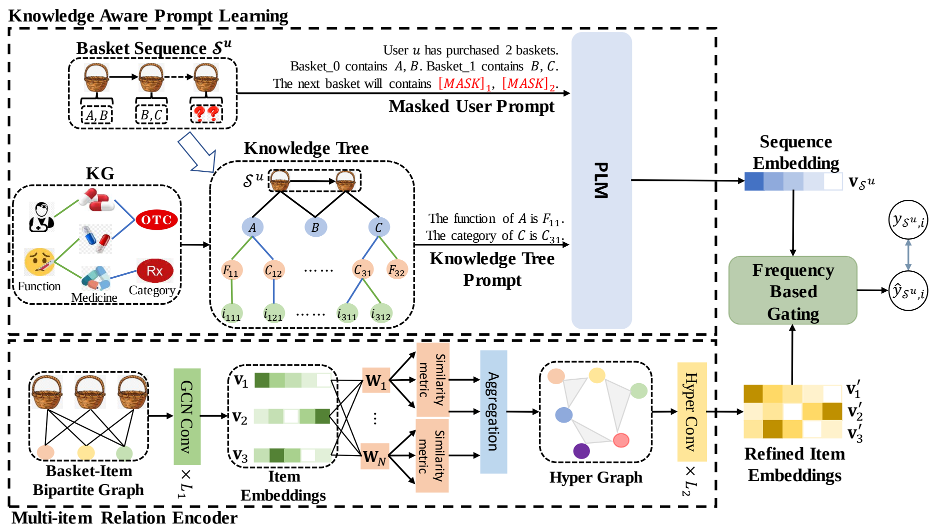


Fig. 2: The framework of HEKP. We first construct MUP and KTP to obtain the sequence embedding by PLM. Then, the user-item bipartite graph is encoded by the multi-item relation encoder. Lastly, we obtain the probability of each item in the next basket guided by frequency.

a subset of $\mathcal{V}$, indicating that multiple vertices are connected by e. The degree of a vertex v, denoted by d_v, is the number of hyperedges that v is incident to. Similarly, the degree of a hyperedge e, denoted by d_e, is the number of vertices that are incident to e.

3.2 Overview

In the following sections, we will describe three main components of HEKP in detail: knowledge aware prompt learning module (Section 3.3), multi-item relation encoder module (Section 3.4) and frequency based gating module (Section 3.5). The overall framework of HEKP is shown in Fig. 2.

3.3 Knowledge Aware Prompt Learning

Masked User Prompt (MUP) Inspired by [4], we construct a masked user prompt to reformulate the NBR task into a mask prediction task. Specifically, we manually design a template $\mathcal{T}_{MUP}$ to transform the basket sequence $\mathcal{S}^u$ into a MUP. For example, suppose the basket sequence is $[\{A, B\}, \{B, C\}, \{C, D\}]$, its corresponding MUP is "*User u has purchased 2 baskets. Basket_ 0 contains A, B. Basket_ 1 contains B, C. The next basket will contains* $[MASK]_1$, $[MASK]_2$", where the two $[MASK]$ tokens represent the ground truth items, i.e., C and D.

Knowledge Tree Prompt (KTP) As mentioned above, OOV item IDs may not be as simple as A, B or C, but words that are beyond PLM's knowledge, or even non-English words. Consider that when we meet an unfamiliar word such as "Artemisinin", we prefer to look up a dictionary and understand that "*The function of Artemisinin is antimalarial.*" than to force ourselves to remember it. Therefore, we construct another KTP as an explanation of the OOV items.

Many researches [25,29] have shown that introducing KG as side information can improve recommendation. However, it's still an open problem about how to obtain high quality representations of the entities in KG. We propose to readout a KTP from KG by building a knowledge tree that includes the most relevant knowledge about the items in the basket sequence $\mathcal{S}^u$.

Specifically, we first augment the original KG following Eq. (1),

$$\mathcal{KG}' = \mathcal{KG} \cup \bigcup_{u\in\mathcal{U}, b_j^u\in\mathcal{S}^u, i\in b_j^u} \left\{\left(b_j^u, \texttt{contain}, i\right)\right\} \tag{1}$$

where `contain` is a new relation. Note that a KG is a set of triplet denoted by $\mathcal{KG} = \{(h,r,t)\}$ where each triplet (h,r,t) indicates that there is a relation r between the head entity h and the tail entity t.

Given a user u, we perform beam search on $\mathcal{KG}'$ starting from all baskets $\left\{b_j^u\right\}$ to all entities within n hops, after which the beam search tree is served as the knowledge tree (as shown in Fig. 2). After the beam search steps, the first layer of the knowledge tree is $\left\{b_j^u\right\}$ representing all baskets in $\mathcal{S}^u$, while the second layer is consists of all items in those baskets. The deeper nodes are the closely-related entities of those items. In other words, we filter the most relevant knowledge of the basket sequence into the knowledge tree. Next, we traverse the knowledge tree in breadth-first manner and record every triplets during the traversal to create a triplet sequence $\mathcal{S}_{tri}^u$,

$$\mathcal{S}_{tri}^u = \left[\left(h_1, r_1, t_1\right), \ldots, \left(h_L, r_L, t_L\right)\right] \tag{2}$$

where L is the number of triplets. The KTP can be defined as Eq. (3),

$$\mathrm{KTP} = \mathrm{concat}\left(\mathcal{T}_{KTP}\left(h_1, r_1, t_1\right), \ldots, \mathcal{T}_{KTP}\left(h_L, r_L, t_L\right)\right) \tag{3}$$

where $\mathcal{T}_{KTP}$ is a template for KTP. In our implementation, the template is defined as $\mathcal{T}_{KTP}(h,r,t) =$"*The r of h is t*".

PLM Encoding and Fine-tuning After we construct MUP and KTP based on user's basket sequence and KG, we input them into PLM and obtain the output embeddings of $[MASK]$ tokens. Since there may be multiple $[MASK]$ tokens, we just simply employ a mean pooling over all $[MASK]$ embeddings to get the sequence embedding $\mathbf{v}_{\mathcal{S}^u} \in \mathbb{R}^{d_1}$ following Eq. (4), where d_1 denotes hidden size of PLM.

$$\mathbf{v}_{\mathcal{S}^u} = \mathrm{MeanPool}\left(\mathrm{PLM}_{[MASK]}\left(\mathrm{MUP}, \mathrm{KTP}\right)\right). \tag{4}$$

Following [4], we use the following negative log likelihood loss to fine-tune the PLM by Eq. (5),

$$\mathcal{L}_{PLM} = -\sum_{j=1}^{|\mathbf{y}|} \log P\left(\mathbf{y}_j \mid \mathbf{y}_{<j}, \mathbf{X}_{\text{MUP}}, \mathbf{X}_{\text{KTP}}\right) \tag{5}$$

3.4 Multi-item Relation Encoder

Item Similarity Encoder Since we do not have any pretrained representation of baskets or items, we first conduct L_1 layers of GCN [10] on the basket-item bipartite graph $\mathcal{G}_{B-I}$ to obtain an initial embedding for each item i and basket b, denoted by $\mathbf{v}_i \in \mathbb{R}^{d_2}$ and $\mathbf{v}_b \in \mathbb{R}^{d_2}$ respectively, where d_2 represents the size of embeddings.

In order to encode the similarity between item i and item j, denoted by $\pi_{i,j}$, we employ a Mixture-of-Experts (MoE) model following Eq. (6) and Eq. (7),

$$\pi_{i,j}^n = \cos\left(\mathbf{W}_n \mathbf{v}_i, \mathbf{W}_n \mathbf{v}_j\right) \tag{6}$$

$$\pi_{i,j} = \text{MeanPool}\left(\pi_{i,j}^1, \pi_{i,j}^2, \ldots, \pi_{i,j}^N\right) \tag{7}$$

where N represents the number of experts, $\mathbf{W}_n \in \mathbb{R}^{d_2 \times d_3}$ is learnable parameter, and d_3 is the hidden dimension of each expert. $\cos(\cdot,\cdot)$ is the cosine similarity and $\text{MeanPool}(\cdot)$ is the mean pooling operation.

We design two pair-wise ranking losses of two different levels. The first basket-item level loss $\mathcal{L}_{B-I}$ is based on the prior that the embedding of a basket b should be more similar to the embedding of an item i^+ that is contained in basket b than to that of an item i^- that is excluded from it. Given a basket b, we randomly sample a positive item $i^+ \in b$ and another negative item $i^- \notin b$ then compute $\mathcal{L}_{B-I}$ as Eq. (8),

$$\mathcal{L}_{B-I} = -\sum_{b \in \mathcal{B}} \log\left(\sigma\left(\mathbf{v}_b \cdot \mathbf{v}_{i^+} - \mathbf{v}_b \cdot \mathbf{v}_{i^-}\right)\right) \tag{8}$$

where $\cdot$ represents the inner product operation and σ is the sigmoid function.

The second item-item level loss $\mathcal{L}_{I-I}$ is based on the prior that the similarity of an item pair (i, i^+) that is contained in the same basket b should be higher than that of an item pair (i, i^-) that is not. Given a basket b and for each item $i \in b$, we randomly sample an positive item $i^+ \in b / \{i\}$ and another negative item $i^- \notin b$ then compute $\mathcal{L}_{I-I}$ as Eq. (9).

$$\mathcal{L}_{I-I} = -\sum_{b \in \mathcal{B}} \frac{1}{|b|} \sum_{i \in b} \log\left(\sigma\left(\pi_{i,i^+} - \pi_{i,i^-}\right)\right) \tag{9}$$

Item Hypergraph Convolution Module We then construct an item hypergraph $\mathcal{HG} = (\mathcal{I}, \mathcal{E}_{hyper})$ by connecting each item with its top-k most similar items using a hyperedge. Specifically, we denote the adjacent matrix of $\mathcal{HG}$ as $\mathcal{M} \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{I}|}$, whose (i,j)-th entry $m_{i,j}$ is obtained by Eq. (10),

$$m_{i,j} = \begin{cases} \pi_{i,j} & j \in \underset{i' \in \mathcal{I}}{\text{topk}}\left(\{\pi_{i,i'}\}\right) \\ 0 & otherwise \end{cases} \tag{10}$$

where $\mathrm{topk}(\cdot)$ returns the indices of the largest k elements.

Inspired by [26], we design a hypergraph convolution module to capture the correlations among multiple items by passing and aggregating message along the hyperedges. Specifically, the l-th layer of the hypergraph convolutional module is defined as Eq. (11),

$$\mathbf{H}^{(l)} = \mathrm{FFN}\left(\mathrm{diag}\left(\frac{1}{d_v}\right) \cdot \mathcal{M} \cdot \mathrm{diag}\left(\frac{1}{d_e}\right) \cdot \mathcal{M}^\top \cdot \mathbf{H}^{(l-1)}\right), l \in [L_2] \tag{11}$$

where d_v and d_e are the degree of nodes and hyperedges respectively. The i-th row of $\mathbf{H}^{(0)}$, denoted by $\mathbf{h}_i^{(0)}$, is initialized by $\mathbf{v}_i$. $\mathrm{FFN}(\cdot)$ represents the learnable feed forward network. L_2 is the layer of convolution and we take the last hidden embedding as the refined item embedding, i.e., $\mathbf{v}'_i = \mathbf{h}_i^{(L_2)} \in \mathbb{R}^{d_2}$.

3.5 Frequency Based Gating (FBG)

In this section, we propose a gating mechanism based on user interaction frequency to fuse the sequence embedding $\mathbf{v}_{\mathcal{S}^u}$ and refined item embeddings $\mathbf{v}'_i$. The idea is built upon the prior knowledge that user has higher preferences to his/her frequently interacted items [7, 8, 14].

Specifically, given a user u, we first count the normalized frequency vector $\gamma^u \in \mathbb{R}^{|\mathcal{I}|}$ whose i-th entry is defined by Eq. (12),

$$\gamma_i^u = \frac{\mathrm{cnt}(u,i)}{\sum_{j\in\mathcal{I}} \mathrm{cnt}(u,j)} \tag{12}$$

where $\mathrm{cnt}(u,i)$ is the is the number of times that user u interact with item i.

Then, the logits of item i is computed as Eq. (13),

$$\hat{y}_{\mathcal{S}^u,i} = \frac{1}{\sqrt{2d_2}}\left(\mathbf{W}_1^\top\left(\mathbf{v}_{\mathcal{S}^u} \oplus \mathbf{v}'_i\right)\left(1-\beta_i^u\alpha_i^u\right) + \gamma_i^u\alpha_i^u\right) \tag{13}$$

where $\mathbf{W}_1 \in \mathbb{R}^{(d_2+d_1)\times 1}$ and β^u is an indicator vector of the positive entry in γ^u, i.e., $\beta^u = \mathbb{I}(\gamma^u > 0)$. β^u is used to control that the interacted items can get integrate information from both the learned embeddings $(\mathbf{v}_{\mathcal{S}^u} \oplus \mathbf{v}'_i)$ and the frequency vector (γ^u) adaptively, while the uninteracted items can only obtain information from the embeddings. The gating vector $\alpha^u = \sigma(\mathbf{W}_2\gamma^u + \mathbf{b}_2)$ is used to control the balance of these two information. $\mathbf{W}_2 \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$ and $\mathbf{b}_2 \in \mathbb{R}^{|\mathcal{I}|}$ are learnable parameters. $\oplus$ represents the vector concatenate operation.

The recommendation loss is calculated as Eq. (14),

$$\mathcal{L}_{Rec} = -\frac{1}{|b_u^+|}\sum_{i\in b_u^+} \hat{y}_{\mathcal{S}^u,i}\log y_{\mathcal{S}^u,i} - \frac{1}{|\mathcal{I}/b_u^+|}\sum_{i\in\mathcal{I}/b_u^+}\left(1-\hat{y}_{\mathcal{S}^u,i}\right)\log\left(1-y_{\mathcal{S}^u,i}\right) \tag{14}$$

where b_u^+ denotes the ground truth basket and $y_{\mathcal{S}^u,i} = 1$ iff. $i \in b_u^+$.

Finally, We can optimize HEKP by multi-task learning as Eq. (15),

$$\mathcal{L} = \mathcal{L}_{PLM} + \mathcal{L}_{Rec} + \mathcal{L}_{B-I} + \mathcal{L}_{I-I} \tag{15}$$

Table 1: Statistics of the datasets.

	Statistics	TaFeng	Poultry	Medicine
Interactions	Users	8,717	11,723	12,946
	Items	4930	238	6262
	Interactions	215,188	308,788	278,561
	Baskets	51,231	45,746	63,072
KG	Entities	4,934	253	6,324
	Relations	1	3	2
	Triplets	9,860	1,428	25,048

4 Experiments

4.1 Experiments Settings

Datasets We first obtain a commonly used NBR dataset Tafeng [20] which contains a Chinese grocery store transaction data from November 2000 to February 2001. Furthermore, since OOV items rarely appear in commonly used NBR datasets, we obtain another two real world datasets. Poultry dataset contains users' interactions with a large agribusiness in China. The item names and attributes in the Poultry dataset are all in Chinese which is not easy to translate into English. Medicine dataset is built upon the sales data of a medicine store in Shenzhen, China. The names of the medicine are either medical terminologies or Chinese herbal medicine.

Following [20], we filter out the baskets that contain fewer than 2 items and randomly select 5 items for those containing more than 5 items. Similarly, we drop the basket sequences whose length are less than 4 and keep the last 10 baskets for those longer than 10. For each datasets, we divide all basket sequences into training, validating and testing set by 8:1:1.

We manually build a KG for each dataset according to the item meta data. For the TaFeng dataset, we take the average prices of an item as its price and divide all price into four buckets, denoting the level of each item by the bucket it belongs to. We construct a KG for TaFeng by connecting each item with its level by a relation `level_is`. For Poultry dataset, we connect each kind of poultry to its grade, gender and category using relation `grade_is`, `gender_is` and `category_is`. For Medicine dataset, we connect each kind of medicine to its therapeutic function and category using relation `function_is` and `category_is`.

Some statistics of three datasets after preprocessing are shown in Table 1.

Evaluation Metrics Following [14, 29], we adopt two widely used metrics to evaluate the performance of HEKP, namely Hit Rate@k (H@k) and Normalized Discounted Cumulative Gain@k (N@k).

Baselines To verify the effectiveness of our HEKP model, we choose the following state-of-the-art NBR methods as baselines in our experiments:

- **KGAT** [25] explicitly models the high-order connectivities in KG by encoding the node embedding in a KG.
- **CLEA** [15] designs a denoising generator to extract items relevant to the target basket automatically and proposes a two-stage contrastive learning.
- **Sets2Sets** [7] (abbreviated as S2S) is an encoder-decoder framework that formulate NBR problem as a sequential sets to sets learning problem.
- **TIFU-KNN** [8] (abbreviated as TIFU) is a simple KNN methods that utilize the personalized item frequency (PIF) information.
- $\mathbf{M^2}$ [14] is a NBR method that takes the users' general preferences, items' global popularities and transition patterns among items into consideration.
- **P5** [4] is a PLM-Rec method that unifies multiple tasks into a single framework by designing personalized prompts.
- **SINE** [22] attempts to infer a set of interests for each user adaptively to model the current intention of users.
- **HapCL** [20] mines information from multiple views and patterns with the help of polar contrastive learning.

Implementation Details In our experiment, we utilize T5-small [16] as the PLM backbone of HEKP. T5 is a widely used encoder-decoder PLM, with a hidden dimension $d_1 = 512$. We set the maximum length of input tokens to 512 and set $n = 3$ when constructing KTPs. We only set the number of GCN layer and hypergraph convolutional layer, i.e. L_1 and L_2, to 2, considering the over-smoothing issue occurring with too-deep GNNs. For the embeddings dimensions, we set $d_2 = 128$ and $d_3 = 64$. We use 1 NVIDIA A800 GPU to train our model with a batch size of 64 for 100 epoch. We fully fine-tune the T5-small backbone with a learning rate of 1e-5 and optimize the overhead with a learning rate of 1e-4 using AdamW optimizer.

4.2 Performance Comparison

The performance comparison results of all baselines and the proposed HEKP are shown in Table 2. The bold and the underlined scores in each column represent the highest and the second highest results of all methods respectively. According to the experimental results, we may conclude the following observations.

HEKP shows significant improvements across all datasets. On one hand, the improvements in NDCG suggest that our method has a strong understanding of the importance and relevance between the next basket and each potential item. This is attributed to the prior knowledge in the PLM and KG, which help our model to obtain better representations of OOV items. On the other hand, our method also achieves sizable improvements in the HR metric, indicating its ability to cover a large proportion of the ground truth items in the next basket. We argue that this improvement stems from the modeling of correlations among multiple items by the multi-item relation encoder. It's worth noting that P5 achieves almost the best performance among all baselines. This demonstrates the great potential of PLM-Rec. Nevertheless, P5 requires more training time

Table 2: Performance comparison results in three datasets. Imp. stands for the improvement of HEKP compared to the best baseline

Metrics		Methods									Imp.
		KGAT	CLEA	S2S	TIFU	M^2	P5	SINE	HapCL	HEKP	
TaFeng	H@5	0.161	0.171	0.140	0.144	0.131	0.203	0.177	0.149	**0.228**	12.31%
	N@5	0.160	0.174	0.165	0.148	0.142	0.212	0.175	0.149	**0.251**	18.39%
	H@10	0.239	0.251	0.207	0.222	0.200	0.263	0.259	0.222	**0.290**	10.26%
	N@10	0.198	0.226	0.186	0.185	0.176	0.224	0.215	0.185	**0.308**	36.28%
Poultry	H@5	0.493	0.607	0.306	0.634	0.652	0.635	0.579	0.441	**0.793**	24.88%
	N@5	0.462	0.589	0.520	0.602	0.594	0.516	0.571	0.409	**0.807**	34.05%
	H@10	0.638	0.801	0.324	0.813	0.791	0.818	0.762	0.671	**0.896**	9.53%
	N@10	0.531	0.677	0.601	0.688	0.670	0.691	0.656	0.514	**0.783**	13.31%
Medicine	H@5	0.240	0.237	0.262	0.298	0.266	0.246	0.231	0.229	**0.368**	23.48%
	N@5	0.274	0.331	0.344	0.340	0.342	0.341	0.328	0.326	**0.447**	29.94%
	H@10	0.262	0.264	0.281	0.350	0.315	0.354	0.259	0.254	**0.377**	6.49%
	N@10	0.288	0.342	0.367	0.368	0.362	0.376	0.339	0.336	**0.428**	13.82%

than HEKP to converge. In light of this, we argue that reformulating the recommendation task into a pretraining task following the prompt learning paradigm significantly contributes to the model's convergence speed.

Some baselines (Sets2Sets, CLEA) attempt to make use of sequential encoders, such as GRU and RNN, to accurately model the transition patterns from basket to basket in the basket sequence. While these methods have produced competitive results, relying solely on item IDs means that these models treat each item independently. Consequently, they can only capture shallow transition patterns and thus fail to discern deeper semantic-level correlations. Some baselines (KGAT) make efforts to integrate KG at an early stage to obtain better item representations. While these methods do improve performance, they still cannot achieve the best results. This is mainly because obtaining high quality representations in large scale KG is a challenging task due to intrinsic noise in the KG. Moreover, the pretrained representations may not be suitable for the downstream task. Additionally, we also notice that TIFU-KNN, despite its simplicity, it still achieve considerable performance in Poultry and Medicine dataset. This is because users in these two datasets exhibit more obvious re-purchase pattern, indicating that modeling interaction frequency of users can be beneficial in some real world NBR scenarios.

Other baselines (SINE, HapCL) introduce graph structure to mine correlation signals and thus achieves relatively competitive performance. Specifically, SINE mines different aspects of user interest from the user's session graph and performs better than HapCL on Poultry dataset. This indicates that mining multiple patterns from the interaction sequence is beneficial, which shares the same idea with MoE component in HEKP. However, none of these methods surpasses HEKP. This is because these methods are based on message passing in a

Table 3: Ablation study results of HEKP and its four variants.

Methods	Poultry		Medicine		TaFeng	
	H@5	N@5	H@5	N@5	H@5	N@5
HEKP	**0.793**	**0.807**	**0.368**	**0.447**	**0.228**	**0.251**
w/o GCN	0.577	0.545	0.319	0.392	0.152	0.170
w/o Hyper-GCN	0.692	0.684	0.291	0.363	0.172	0.188
w/o FBG	0.591	0.605	0.258	0.348	0.205	0.220
w/o KTP	0.694	0.632	0.325	0.398	0.144	0.151

plain graph to model binary item relation, ignoring that item dependencies may be higher order.

4.3 Ablation Study

We validate the effectiveness of several components in HEKP by removing the GCN (Section 3.4), Hyper-GCN (Section 3.4), FBG (Section 3.5) and KTP (Section 3.3).The results of the ablation study are shown in Table 3.

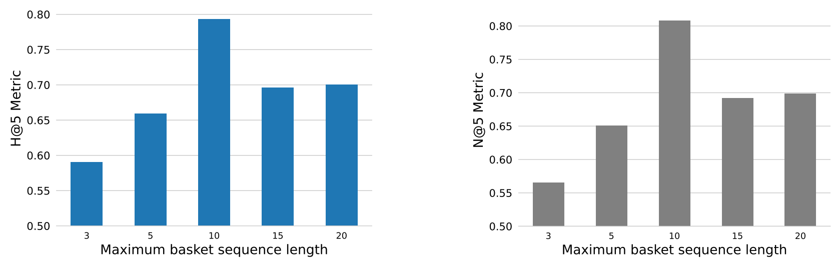
We can see that the performance of all variants degenerate, verifying the effectiveness of our architecture. Specifically, it's evident that on the Poultry dataset, which contains relatively fewer items and denser interactions, the absence of GCN and FBG will significantly degrades performance by more than 30%. This suggests that exploiting the collaborative signal between baskets and items in the basket-item bipartite graph can be beneficial to obtain better item embeddings. Similarly, fewer items imply denser purchase frequency vectors γ of users, meaning that users have more distinct preferences and repurchase patterns. In this case, the proposed FBG is more powerful in mining these patterns.

On the larger and sparser datasets Medicine and TaFeng, the hypergraph convolutional module plays an important role, which will cause over 25% degradation if removed. This is because there are more combinations of items and preferences varying among users in the Medicine dataset while the hypergraph convolutional module is suitable for modeling semantic correlations of different item combinations.

We also find that leveraging KG as side information can improve the performance by about 20% for three datasets respectively. We can conclude that the proposed KTP can help our model to encode the OOV items by mining the deep semantic information of those items, thus improving the performance.

4.4 Hyperparameter Analysis

In Section 4.2 and Section 4.3, we have shown that providing KTP together with MUP into PLM can benefit the recommendation performance. However, since the input token sequence length of PLM cannot be infinite, how to balance between the length of MUP and KTP becomes a trade-off problem since that a longer basket sequence means a longer MUP and thus a shorter KTP. We try to investigate a reasonable proportion between the two lengths in this section.



(a) Results on H@5 metric. (b) Results on N@5 metric.

Fig. 3: Hyperparameter analysis on the maximum basket sequence length.

We first fix the input token length of PLM to 512 and then change the maximum basket sequence length to 3, 5, 10, 15 and 20 respectively. We examine the performance corresponding to different maximum length and report the results on Poultry dataset in Fig. 3.

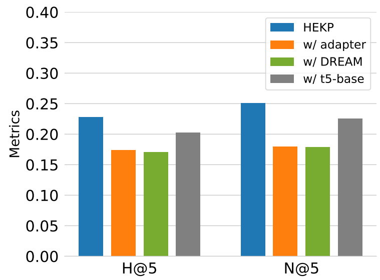
We can see that all metrics peak when the maximum basket sequence length is 10, which is consistent with our setting in the previous study. As the maximum length decreases, all metrics decrease sharply. This implies that the importance of the user's interactions outweighs the prior knowledge of items. When the interaction is insufficient, the model exhibits significant bias in modeling user preferences. However, a longer basket sequences do not necessarily guarantee better performance. This is because the recent interactions tend to represent the user preferences more accurately while those from too long ago will introduce noise. In addition, a too-long MUP can result in insufficient knowledge input to the PLM, leading to performance degradation.

4.5 Impact of PLM

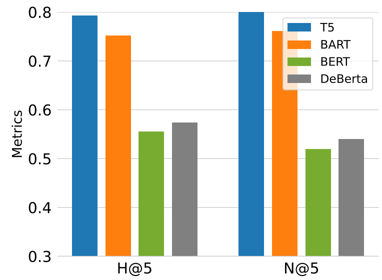
Experiment on Different Variants of PLM First, we design the following 3 variants to investigate the impact of PLM:

- w/ adapter: we freeze the pretrained parameters of PLM and only train an MLP as an adapter to the downstream task.
- w/ DREAM: we replace the PLM with a well-known NBR model DREAM [27] as a sequence encoder.
- w/ t5-base: we use a larger PLM T5-base. Due to the limited computing resources, we use the same settings as w/ adapter.

The comparison of HEKP and these three variants are listed in Table 4a. We can draw the following observations from the results. First of all, fully finetuning the whole PLM can lead to better performance. Secondly, we can see that the two variants w/ adapter and w/ DREAM show similar performance, but the trainable parameters in the first variant are much smaller than in the



(a) Different variants of PLMs.



(b) Different types of PLMs.

Fig. 4: Experiments on the different variants (left) and types (right) of PLMs.

second one. Therefore, utilizing prior knowledge in the PLM can lead to relatively good performance with a much smaller training cost. Lastly, using a larger PLM can boost the performance in a certain degree but it will introduce heavy computation cost for training.

Experiments on Different Types of PLM Next, we wonder what kind of PLM is more appropriate for HEKP, so we examine the overall performance with respect to four different PLM backbones, which can be divided into two architectures: encoder-decoder (T5 [16], BART [12]) and encoder-only (BERT [3], DeBerta [5]). The results on Poultry dataset are shown in Fig. 4b. We can conclude the following observations from the results.

First of all, the encoder-decoder architecture PLMs significantly surpass those of the encoder-only architecture. This is because encoder-decoder PLMs are more suitable for sequence-to-sequence generation tasks such as mask prediction, translation and question answering. On the contrary, encoder-only PLMs are more effective in extracting global contextual features and thus perform better in global comprehension tasks like sentence classification.

In addition, more parameters are not sufficient for better performance, while the gap between the pretraining and the downstream task has a significant impact. Although BART has more parameters than T5, it fails to outperform it. This is because BART is pretrained by reconstructing documents corrupted by text infilling and sentence shuffling [12], while T5 is pretrained by BERT-style mask prediction task [16] which is perfectly suited to the recommendation task. Therefore, we are able to bridge the gap between the T5's pretraining and downstream task by MUP.

5 Conclusion

In this paper, we propose HEKP, which is a hypergraph enhanced PLM-based NBR method. Firstly, we transform KG into KTPs to help PLM encode the OOV items in the basket sequence. In this way, we are able to reformulate the NBR task into the mask prediction task following the prompt learning paradigm.

Secondly, we propose a novel multi-item relation encoder. Specifically, we construct a hypergraph based on the item similarities measured by an MoE model from different aspects and employ convolution on the hypergraph to model correlations among multiple items. Finally, extensive experiments demonstrate the effectiveness of HEKP. In the future work, we will continue to explore how to use large scale PLMs to boost performance in a parameter efficient way.

Acknowledgments

The work was supported by National Key Research and Development Program of China (2021YFF1201200), NSFC (61876193), NSF (III-2106758), Guangdong Basic and Applied Basic Research Foundation (2022B1515120059), Guangdong Provincial Key Laboratory of Intellectual Property and Big Data (2018B030322016), and Open Project Program of Guangxi Key Laboratory of Digital Infrastructure under Grant No. GXDIOP2024011. Chang-Dong Wang is the corresponding author.

References

1. Bao, K., Zhang, J., Zhang, Y., Wang, W., Feng, F., He, X.: Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In: RecSys. pp. 1007–1014 (2023)
2. Deng, Z., Li, J., Guo, Z., Liu, W., Zou, L., Li, G.: Multi-view multi-aspect neural networks for next-basket recommendation. In: SIGIR. pp. 1283–1292 (2023)
3. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL. pp. 4171–4186 (2019)
4. Geng, S., Liu, S., Fu, Z., Ge, Y., Zhang, Y.: Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5). In: RecSys. pp. 299–315 (2022)
5. He, P., Liu, X., Gao, J., Chen, W.: Deberta: decoding-enhanced bert with disentangled attention. In: ICLR (2021)
6. Hou, Y., Mu, S., Zhao, W.X., Li, Y., Ding, B., Wen, J.: Towards universal sequence representation learning for recommender systems. In: KDD. pp. 585–593 (2022)
7. Hu, H., He, X.: Sets2sets: Learning from sequential sets with neural networks. In: KDD. pp. 1491–1499 (2019)
8. Hu, H., He, X., Gao, J., Zhang, Z.: Modeling personalized item frequency information for next-basket recommendation. In: SIGIR. pp. 1071–1080 (2020)
9. Hua, W., Xu, S., Ge, Y., Zhang, Y.: How to index item ids for recommendation foundation models. In: SIGIR. pp. 195–204 (2023)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
11. Le, D., Lauw, H.W., Fang, Y.: Correlation-sensitive next-basket recommendation. In: IJCAI. pp. 2808–2814 (2019)
12. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: ACL. pp. 7871–7880 (2020)

13. Li, R., Zhang, L., Liu, G., Wu, J.: Next basket recommendation with intent-aware hypergraph adversarial network. In: SIGIR. pp. 1303–1312 (2023)
14. Peng, B., Ren, Z., Parthasarathy, S., Ning, X.: M^2: Mixed models with preferences, popularities and transitions for next-basket recommendation. TKDE **35**(4), 4033–4046 (2023)
15. Qin, Y., Wang, P., Li, C.: The world is binary: Contrastive learning for denoising next basket recommendation. In: SIGIR. pp. 859–868 (2021)
16. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 140:1–140:67 (2020)
17. Rajput, S., Mehta, N., Singh, A., Keshavan, R.H., Vu, T., Heldt, L., Hong, L., Tay, Y., Tran, V.Q., Samost, J., Kula, M., Chi, E.H., Sathiamoorthy, M.: Recommender systems with generative retrieval. In: NeuIPS (2023)
18. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: WWW. pp. 811–820 (2010)
19. Su, T., He, Z., Chen, M., Wang, C.: Basket booster for prototype-based contrastive learning in next basket recommendation. In: PKDD. vol. 13713, pp. 574–589 (2022)
20. Su, T., Wang, C., Xi, W., Lai, J., Yu, P.S.: Hierarchical alignment with polar contrastive learning for next-basket recommendation. TKDE **36**(1), 199–210 (2024)
21. Sun, W., Xie, R., Zhang, J., Zhao, W.X., Lin, L., Wen, J.: Generative next-basket recommendation. In: RecSys. pp. 737–743. ACM (2023)
22. Tan, Q., Zhang, J., Yao, J., Liu, N., Zhou, J., Yang, H., Hu, X.: Sparse-interest network for sequential recommendation. In: WSDM. pp. 598–606 (2021)
23. Wang, S., Hu, L., Wang, Y., Sheng, Q.Z., Orgun, M.A., Cao, L.: Intention nets: Psychology-inspired user choice behavior modeling for next-basket prediction. In: AAAI. pp. 6259–6266 (2020)
24. Wang, S., Hu, L., Wang, Y., Sheng, Q.Z., Orgun, M.A., Cao, L.: Intention2basket: A neural intention-driven approach for dynamic next-basket planning. In: IJCAI. pp. 2333–2339 (2020)
25. Wang, X., He, X., Cao, Y., Liu, M., Chua, T.: KGAT: knowledge graph attention network for recommendation. In: KDD. pp. 950–958 (2019)
26. Yang, Y., Huang, C., Xia, L., Liang, Y., Yu, Y., Li, C.: Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In: KDD. pp. 2263–2274 (2022)
27. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: A dynamic recurrent model for next basket recommendation. In: SIGIR. pp. 729–732 (2016)
28. Yu, Y., Yang, E., Guo, G., Jiang, L., Wang, X.: Basket representation learning by hypergraph convolution on repeated items for next-basket recommendation. In: IJCAI. pp. 2415–2422 (2023)
29. Zhai, J., Zheng, X., Wang, C., Li, H., Tian, Y.: Knowledge prompt-tuning for sequential recommendation. In: ACM MM. pp. 6451–6461 (2023)
30. Zheng, B., Hou, Y., Lu, H., Chen, Y., Zhao, W.X., Chen, M., Wen, J.: Adapting large language models by integrating collaborative semantics for recommendation. In: ICDE. pp. 1435–1448 (2024)