VQ-Rec

1. Motivation

（1） 基于自然语言文本处理的序列推荐模型大多直接采用预训练得到的文本表征作为物品表征，这样会导致文本与项目结合过于紧密，而忽略了序列信息；

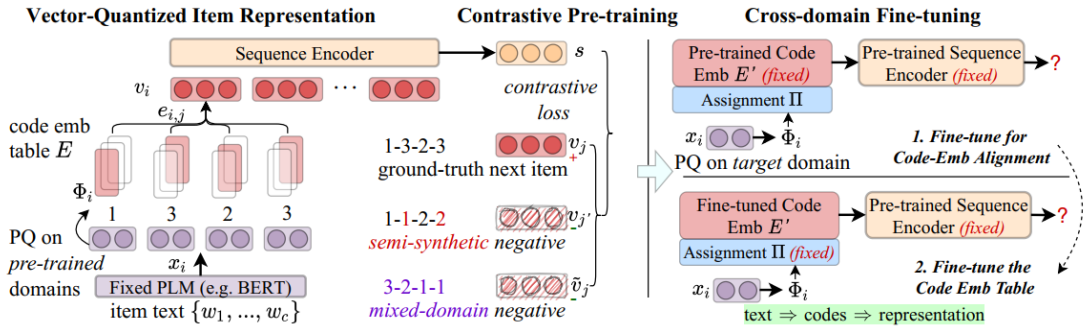
（2） 来自不同领域的文本表征在一个统一的语义空间不对齐。

基于上述问题，解决方案是：

（1） 将文本转换为离散代码，用代码向量作为物品表征，解除文本与物品表征之间的强绑定；

（2） 在不同域上预训练以调整代码表示。

2. FrameWork



模型整体分为三个部分：

（1） Vector-Quantized Item Representation：利用优化乘积量化（OPQ）技术将预训练得到的文本编码转换为离散的代码向量；

（2） Contrastive Pre-training：构建负样本对预训练进行对比学习；

（3） Cross-domain Fine-tuning：在跨域上进行代码微调

序列推荐部分采用的是 TRM，和代码表征部分是解耦的，可以自己更换实现序列推荐的模块。

2.1. Vector- Quantized Item Representation

目的是获得物品的文本对应的离散代码向量。

2.1.1. Vector-Quantized Code Learning

1. **item code→text encodings**：物品的文本首先通过预训练器（比如 BERT）得到初始表征:

$$\boldsymbol{x}_i = \text{BERT}([\text{[CLS]}; w_1; \ldots; w_c]),$$

$\boldsymbol{x}_i \in \mathbb{R}^{d_W}$ 每个文本都添加 CLS 标记，其经过 PLM 训练后的输出代表具有上下文信息的表征x_i。

2. **text-encodings→discrete codes**：采用乘积量化（PQ）技术，大小为d_w的向量划分为 D 个子空间，每个子空间维度为$^{d_w}/_{D}$，然后在每个子空间上使用 kmeans 方法聚类，得到 M 类，也就是每个子空间都有 M 个中心点（centroids）。这 M 个中心点构成该子空间下的一个码书。$\boldsymbol{a}_{k,j} \in \mathbb{R}^{d_W/D}$ 代表第 k 个子空间的第 j 个中心点。对于文本表征x_i，首先划分为 D 份：

$$\boldsymbol{x}_i = [\boldsymbol{x}_{i,1}; \ldots; \boldsymbol{x}_{i,D}]$$

以第 k 份为例，通过下面的公式找到第 k 个子空间下距离$x_{i,k}$最近的中心点的索引：

$$c_{i,k} = \arg\min_j \|\boldsymbol{x}_{i,k} - \boldsymbol{a}_{k,j}\|^2 \in \{1, 2, \ldots, M\}, \qquad (2)$$

最终得到物品 i 的离散代码表示：

$$\boldsymbol{c}_i = (c_{i,1}, \ldots, c_{i,D})$$

2.1.2. Code Embedding Lookup as Item Representation

旨在基于索引得到对应的向量。

每个子空间都有一个码表，是中心点的向量表征矩阵：$\boldsymbol{E}^{(k)} \in \mathbb{R}^{M \times d_V}$。通过上面得到的索引在该矩阵中获取对应的向量$\{\boldsymbol{e}_{1,c_{i,1}}, \ldots, \boldsymbol{e}_{D,c_{i,D}}\}$。e 并不是上述的 a，而是经过随机初始化得到的。最后将不同空间下的表示平均得到物品的表征：

$$\boldsymbol{v}_i = \text{Pool}\left([\boldsymbol{e}_{1,c_{i,1}}; \ldots; \boldsymbol{e}_{D,c_{i,D}}]\right), \qquad (3)$$

经过上述 text→code→representation 的转换，在一定程度上保持文本语义相似度的同时，减少了文本和物品之间的依赖。同时，通过 Kmeans 的方法，能够避免

两个物品之间的离散代码碰撞（相同）。

2.2. Contrastive Recommender Pre-training

旨在为了采用对比学习，选择负样本。包括：半合成离散代码负样本以及混合域负样本。

2.2.1. Semi-synthetic negatives

对物品 i 的离散代码表示 $\boldsymbol{c}_i = (c_{i,1}, \ldots, c_{i,D})$ 以一定概率将其采样替换为同一子空间下其他的聚类中心的索引。即：

$$\widetilde{\boldsymbol{v}}_i = \text{Emb-Pool}\left(\text{G}(\boldsymbol{c}_i)\right), \tag{6}$$

$$\text{G}(c_{i,j}) = \begin{cases} \text{Uni}(\{1, \ldots, M\}), & X = 1 \\ c_{i,j}, & X = 0 \end{cases}, \tag{7}$$

X 服从伯努利分布$B(\rho)$

2.2.2. Mix-domain negatives

从多个域中随机挑选序列构成 batch，在将一个 batch 内非本序列的其他序列作为负样本。

最终的对比学习损失函数：

$$\ell = -\frac{1}{B}\sum_{j=1}^{B} \log \frac{\exp\left(\boldsymbol{s}_j \cdot \boldsymbol{v}_j/\tau\right)}{\underbrace{\exp\left(\boldsymbol{s}_j \cdot \widetilde{\boldsymbol{v}}_j/\tau\right)}_{\text{semi-synthetic}} + \sum_{j'=1}^{B} \underbrace{\exp\left(\boldsymbol{s}_j \cdot \boldsymbol{v}_{j'}/\tau\right)}_{\text{mixed-domain}}}, \tag{8}$$

2.3. Cross-domain Recommender Fine-tuning

跨域，只对物品表征进行微调，从而只考虑迁移码表。因此分为两个阶段：微调离散代码排列和微调码表。

新域下的物品参照公式（1）（2）得到新的离散代码。对于代码索引表，保留索引集合，更改索引到嵌入的映射。即对代码嵌入表行进行排列组合（行变换）：

$$\hat{E}^{(k)} = \Pi_k \cdot E^{(k)}. \quad (9)$$

$\widehat{E^{(k)}}$代表新领域第 k 个子空间下的代码嵌入。Π_k是一个置换矩阵，因为任意一个双随机矩阵可以视为一个同阶的置换矩阵的凸组合，所以通过初始化一个参数矩阵，并通过 Gumbel-Sinkhorn 将其转换为双随机矩阵，然后通过如下公式：

$$P(i_{t+1}|i_1, \ldots, i_t) = \text{Softmax}(\hat{s} \cdot \hat{v}_{i_{t+1}}), \quad (10)$$

以交叉损失进行预训练得到Π_k进而得到$\widehat{E^{(k)}}$。其中，$\hat{s}$，$\hat{v}_{t+1}$是新域下的序列表征和真实的物品表征。

3. Experiments

预训练的域：Amazon Food、Home、CDs、Kindle、Movies
跨域：Scientific、Pantry、Instruments、Arts、Office
以及跨平台：Online Retail。对于 amazon 文本，是将 title、categories 和 brand 拼接而成，Online 使用 description。其数据统计如下：

Datasets	#Users	#Items	#Inters.	Avg. n	Avg. c
Pre-trained	1,361,408	446,975	14,029,229	13.51	139.34
Scientific	8,442	4,385	59,427	7.04	182.87
Pantry	13,101	4,898	126,962	9.69	83.17
Instruments	24,962	9,964	208,926	8.37	165.18
Arts	45,486	21,019	395,150	8.69	155.57
Office	87,436	25,986	684,837	7.84	193.22
Online Retail	16,520	3,469	519,906	26.90	27.80

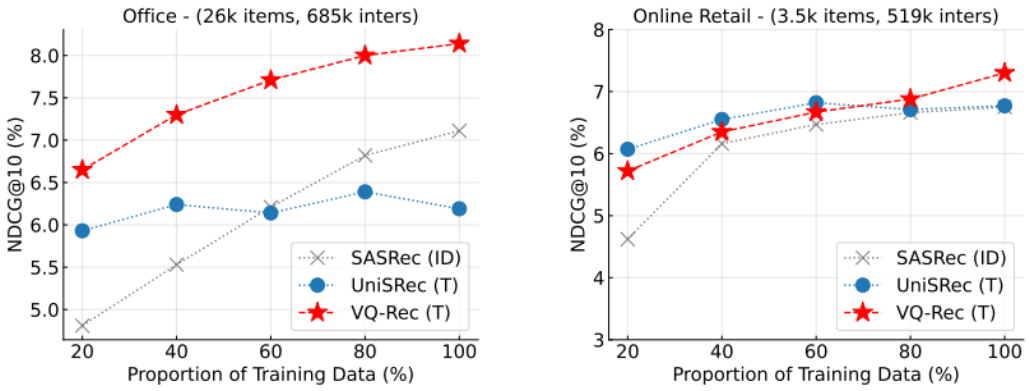
3.1. 对比实验

Scenario	Dataset	Metric	SASRec $_{ID}$	BERT4Rec $_{ID}$	FDSA $_{ID+T}$	S^3-Rec $_{ID+T}$	RecGURU $_{ID}$	SASRec $_{T}$	ZESRec $_{T}$	UniSRec $_{T}$	VQ-Rec $_{T}$
Cross-Domain	Scientific	R@10	0.1080	0.0488	0.0899	0.0525	0.1023	0.0994	0.0851	0.1188	**0.1211**
		N@10	0.0553	0.0243	0.0580	0.0275	0.0572	0.0561	0.0475	0.0641	**0.0643**
		R@50	0.2042	0.1185	0.1732	0.1418	0.2022	0.2162	0.1746	**0.2394**	0.2369
		N@50	0.0760	0.0393	0.0759	0.0468	0.0786	0.0815	0.0670	**0.0903**	0.0897
	Pantry	R@10	0.0501	0.0308	0.0395	0.0444	0.0469	0.0585	0.0454	0.0636	**0.0660**
		N@10	0.0218	0.0152	0.0209	0.0214	0.0209	0.0285	0.0230	**0.0306**	0.0293
		R@50	0.1322	0.1030	0.1151	0.1315	0.1269	0.1647	0.1141	0.1658	**0.1753**
		N@50	0.0394	0.0305	0.0370	0.0400	0.0379	0.0523	0.0378	**0.0527**	**0.0527**
	Instruments	R@10	0.1118	0.0813	0.1070	0.1056	0.1113	0.1127	0.0783	0.1189	**0.1222**
		N@10	0.0612	0.0620	**0.0796**	0.0713	0.0681	0.0661	0.0497	0.0680	0.0758
		R@50	0.2106	0.1454	0.1890	0.1927	0.2068	0.2104	0.1387	0.2255	**0.2343**
		N@50	0.0826	0.0756	0.0972	0.0901	0.0887	0.0873	0.0627	0.0912	**0.1002**
	Arts	R@10	0.1108	0.0722	0.1002	0.1003	0.1084	0.0977	0.0664	0.1066	**0.1189**
		N@10	0.0587	0.0479	**0.0714**	0.0601	0.0651	0.0562	0.0375	0.0586	0.0703
		R@50	0.2030	0.1367	0.1779	0.1888	0.1979	0.1916	0.1323	0.2049	**0.2249**
		N@50	0.0788	0.0619	0.0883	0.0793	0.0845	0.0766	0.0518	0.0799	**0.0935**
	Office	R@10	0.1056	0.0825	0.1118	0.1030	0.1145	0.0929	0.0641	0.1013	**0.1236**
		N@10	0.0710	0.0634	**0.0868**	0.0653	0.0768	0.0582	0.0391	0.0619	0.0814
		R@50	0.1627	0.1227	0.1665	0.1613	0.1757	0.1580	0.1113	0.1702	**0.1957**
		N@50	0.0835	0.0721	**0.0987**	0.0780	0.0901	0.0723	0.0493	0.0769	0.0972
Cross-Platform	Online Retail	R@10	0.1460	0.1349	0.1490	0.1418	0.1467	0.1380	0.1103	0.1449	**0.1557**
		N@10	0.0675	0.0653	0.0719	0.0654	0.0658	0.0672	0.0535	0.0677	**0.0730**
		R@50	0.3872	0.3540	0.3802	0.3702	0.3885	0.3516	0.2750	0.3604	**0.3994**
		N@50	0.1201	0.1131	0.1223	0.1154	0.1188	0.1137	0.0896	0.1149	**0.1263**

在交互数据多的数据集上，基于文本表征的模型不如基于 ID 的验证了基于文本的过度依赖文本相似性。

3.2. 消融实验

Variants	Scientific				Office				Online Retail			
	R@10	N@10	R@50	N@50	R@10	N@10	R@50	N@50	R@10	N@10	R@50	N@50
(0) VQ-Rec	**0.1211**	**0.0643**	**0.2369**	**0.0897**	0.1236	0.0814	**0.1957**	0.0972	0.1557	0.0730	**0.3994**	0.1263
(1) *w/o* Pre-training	0.1104	0.0593	0.2238	0.0839	0.1108	0.0666	0.1804	0.0818	0.1535	0.0717	0.3953	0.1244
(2) *w/o* Semi-synthetic NS	0.1194	0.0631	0.2358	0.0886	0.1227	0.0809	0.1951	0.0968	0.1529	0.0702	0.3938	0.1230
(3) *w/o* Fine-tuning	0.0640	0.0361	0.0909	0.0421	0.0261	0.0150	0.0373	0.0174	0.0197	0.0095	0.0419	0.0142
(4) Reuse PQ Index Set	0.1168	0.0618	0.2267	0.0859	0.1182	0.0773	0.1869	0.0922	0.1521	0.0707	0.3917	0.1232
(5) *w/o* Code-Emb Alignment	0.1183	0.0612	0.2355	0.0867	**0.1242**	0.0824	0.1956	**0.0980**	0.1515	0.0695	0.3924	0.1224
(6) Random Code	0.0905	0.0494	0.1769	0.0683	0.1134	**0.0837**	0.1698	0.0960	**0.1589**	**0.0769**	0.3933	**0.1282**



数据越稀疏，性能越好。