南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

.5

RecGPT: Generative Personalized Prompts for Sequential Recommendation via ChatGPT Training Paradigm

Yabin Zhang[1], Wenhui Yu[2,(✉)], Erhan Zhang[3], Xu Chen[1], Lantao Hu[2], Peng Jiang[2], Kun Gai[4]

[1] Gaoling School of Artificial Intelligence, Renmin University of China, China
{yabin.zhang,xu.chen}@ruc.edu.cn
[2] Kuaishou Technology, Beijing, China
yuwenhui07@kuaishou.com,hulantao@gmail.com,jp2006@139.com
[3] Peking University, Beijing, China,
zhangeh-ss@stu.pku.edu.cn;
[4] Unaffliated, Beijing, China,
{gai.kun}@qq.com

.5

RecGPT：基于ChatGPT训练范式的生成式个性化序列推荐算法

., 2024/11/11

引言——背景问题

现有基于ChatGPT的推荐模型存在以下问题：

现有方法都是在语义空间中表示用户和项目，在表示复杂的偏好关系（上下文、交互时序、兴趣变化）时存在局限性。

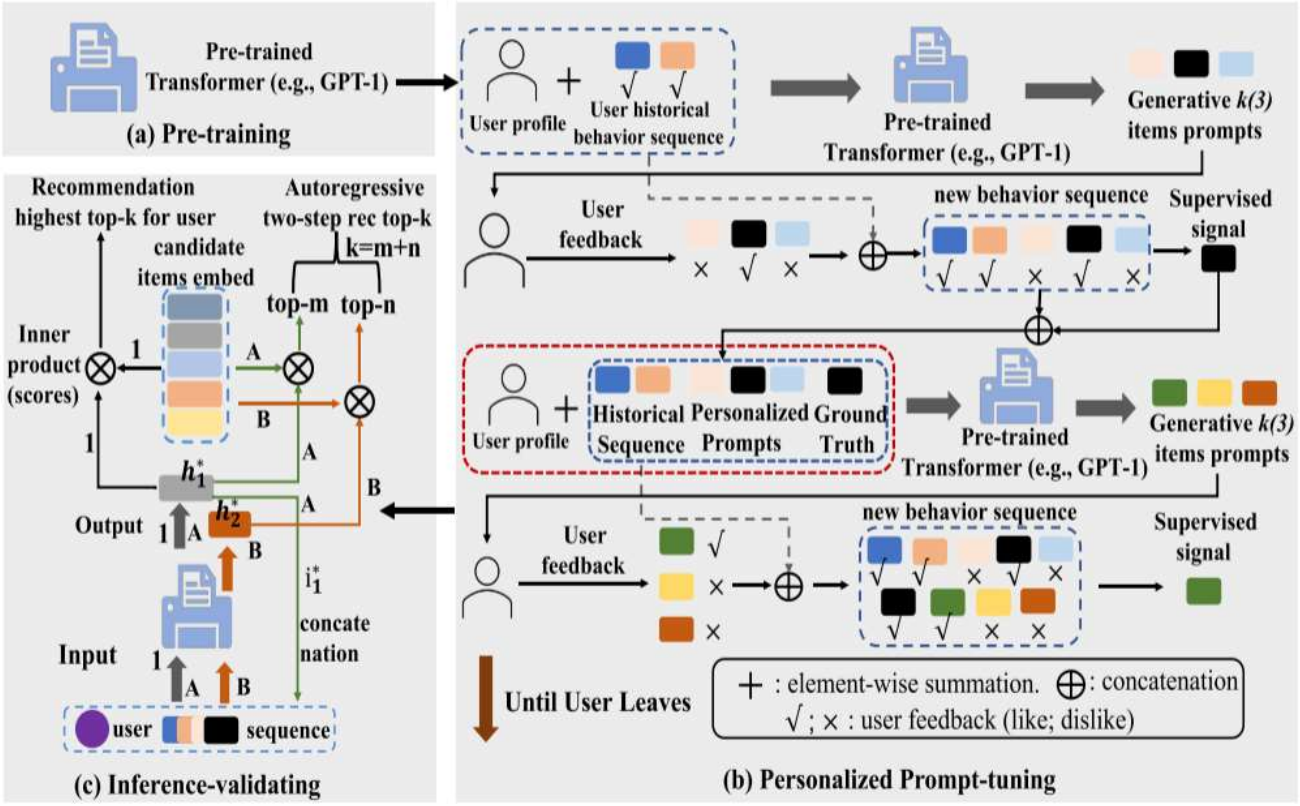
大语言模型对于在线推荐服务来说过于繁重。

现有的序列推荐系统存在以下问题：

用户的历史行为序列与整个交互空间相比非常稀疏，这可能导致模型无法充分利用所有可用的交互数据。

现有方法现有方法只关注用户点击的项目，而忽略了未点击项目的信息。和现实世界逻辑不符。

方法——全流程

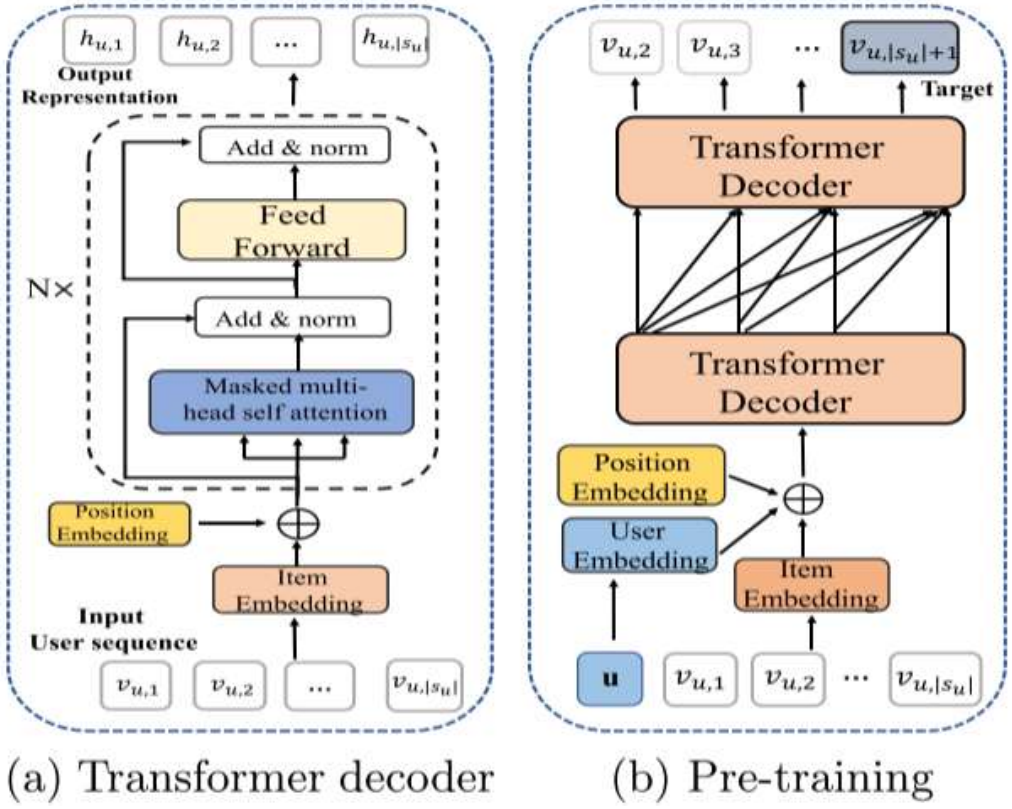


用历史序列预训练

两步推理验证

生成式个性化微调

方法——预训练



(a) Transformer decoder (b) Pre-training

$$L_{SR} = -\sum_{u \in \mathcal{U}} \sum_{t \in [2, |s_u|]} \{\log(\sigma([h_{u,t}]^T \boldsymbol{e}_{u,t+1})) - \sum_{s_{u,neg} \in O^-_{u,t}} \log(1 - \sigma([h_{u,t}]^T \boldsymbol{e}_{u,neg}))\}, \quad .6$$

结合序列内的每个正样本，和对应的负样本集，binary交叉熵计算损失函数。最大化对正样本的预测概率并最小化对负样本的预测概率。

.6 该概率

$$p(v_u, |s_u| + 1 | s_u, 1 * , s_u, 2 + K * , s_u, 3 + 2K * ,$$

...

;
θ
L
P
)
p(v
u,|s
u

|+1

|s
u,1
∗

,s
u,2+K
∗

,s
u,3+2K
∗

,...;θ
LP

) 表示在给定用户
u
u 的提示增强序列
s
u
∗
s

u
∗

下，模型预测用户的下一次可能交互物品
v
u
,
|
s
u
|
+
1
v
u,|s
u

|+1

的条件概率。
., 2024/11/11

方法——微调1 生成个性化提示词

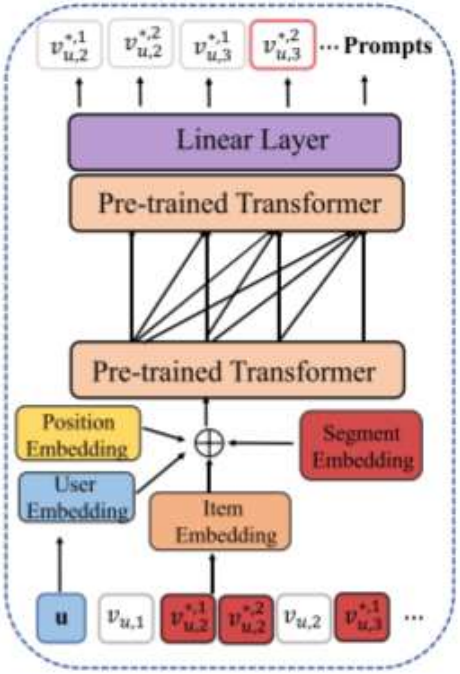
推荐需要针对每个用户定制化，提示也需根据用户偏好进行个性化。因此，在实际系统中，需要对大量用户生成个性化提示，无法手动设计每个用户的提示内容。

Algorithm 1 Pseudocode to generate prompts sequence s_u^*

Input: User id u; Pre-trained "Transformer_block"; User behavior sequence $s_u = \{v_{u,1}, \cdots, v_{u,|s_u|}\}$; User embedding matrix : W_u; Item embedding matrix : W_e; Position embedding matrix : W_p; Segment embedding matrix : W_s; Linear output layer W_l; Whole items set $\mathcal{V}$; The number of personalized prompts generated: K.
Output: New sequence s_u^*.

Let $t = 1$; $s_u^* = \{v_{u,1}\}$
while $t < |s_u|$ **do**
  $k = 1$
  if $k < K$ **then**
    $h_u^0 = uW_u + s_u^* W_e + W_p + W_s$;
    h_u^l = Transformer_block(h_u^{l-1})
    $P_{u,t+1}$=softmax$(h_{u,|s_u|}^N W_l^T)$
    $v_{u,t+2}^{*,k}$=argmax$\mathcal{V}(P_{u,t+1})$
    $s_u^* \leftarrow$ Concatenation$(s_u^*, v_{u,t+2}^{*,k})$
    $k = k + 1$
  end if
  $s_u^* = [s_u^*, v_{u,t+2}]$
end while
return s_u^*

方法——微调2 Prompt-tuning



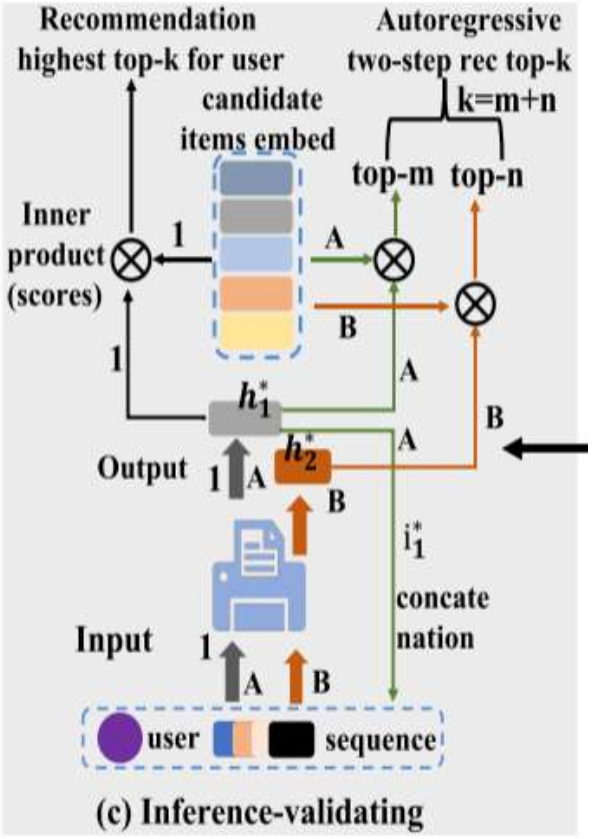
(c) Generative Prompts

$$L = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} -\log p(v_{u,|s_u|+1} | s^*_{u,1}, s^*_{u,2+K}, s^*_{u,3+2K}, \cdots, s^*_{u,|s_u|+(|s_u|-1)K}; \theta_{LP})$$

负对数似然估计。最小化预测值和真实值的差异。

$$s^*_u = \{v_{u,1}, \underline{v^{*,1}_{u,2}, \cdots, v^{*,K}_{u,2}}, v_{u,2}, \underline{v^{*,1}_{u,3}, \cdots, v^{*,K}_{u,3}}, v_{u,3}, \cdots, \underline{v^{*,1}_{u,|s_u|-1}, \cdots, v^{*,K}_{u,|s_u|-1}}, v_{u,|s_u|}\}$$

方法——推理验证



One step callback

Two step callback

实验——数据集 预处理 评价指标

Table 1: Statistics of the datasets after preprocessing.

Dataset	#Users	#Items	#AveLen	Actions	Sparsity
Sports	35,598	18,357	8.3	296,337	99.95%
Beauty	22,363	12,101	8.9	198,502	99.73%
Toys	19,412	11,924	8.6	167,597	99.93%
Yelp	30,431	20,033	10.3	316,354	99.95%

5-core过滤：要求每个用户至少与5个项目有交互（例如点击、购买等），每个项目也至少被5个用户交互。

交互记录处理：将用户的交互记录视为正样本，对于评分数据，将具有评分的记录或带有评论的记录视为正样本，未满足条件的视为负样本。这种操作确保模型能更好地捕捉用户的偏好。

命中率（HR@k）：
HR@k用于评估推荐列表的召回能力。HR@k的值介于0到1之间，值越高表示推荐系统推荐的K个项目中有用户感兴趣项目的可能性越大。

归一化折损累计增益（NDCG@k）：
NDCG@k衡量推荐列表中项目的排序质量，值也介于0到1之间，值越高表示推荐列表中用户感兴趣的项目排序越靠前，推荐效果越好。

幻灯片 8

.1 前三个数据集来自亚马逊评论数据集，第四个是一个常用的商业推荐数据集

., 2024/11/11

.3 HR@k衡量的是模型推荐成功的频率，但不考虑推荐项目的排序或位置。

., 2024/11/11

实验——实验一 性能比较

Model	Sports and Outdoors				Beauty			
	Metric@5		Metric@10		Metric@5		Metric@10	
	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
PopRec	0.0052	0.0028	0.0081	0.0037	0.0064	0.0030	0.0098	0.0040
BPR	0.0030	0.0019	0.0055	0.0027	0.0038	0.0026	0.0060	0.0033
STAMP	0.0079	0.0051	0.0119	0.0064	0.0077	0.0045	0.0114	0.0057
GRU4Rec	0.0113	0.0072	0.019	0.0097	0.0162	0.0097	0.0300	0.0141
NARM	0.0132	0.0085	0.0234	0.0118	0.0230	0.0142	0.0401	0.0197
S^3-Rec_{IPS}	0.0124	0.0086	0.0205	0.0111	0.0202	0.0119	0.0336	0.0163
BERT4Rec	0.0200	0.0130	0.0313	0.0166	0.0382	0.0210	0.0592	0.0319
SASRec	0.0203	0.0135	0.0324	0.0174	0.0398	0.0261	0.0614	0.0331
Pre-train	0.0203	0.0132	0.0324	0.0171	0.0418	0.0272	0.0610	0.0334
Fine-tuning	0.0212	0.0141	0.0328	0.0178	0.0421	0.0275	0.0618	0.0338
$RecGPT_1$	0.0213	0.0141	0.0333	0.0180	0.0426	0.0283	0.0651	0.0356
RecGPT	**0.0219**	**0.0143**	**0.0339**	**0.0181**	**0.0440**	**0.0289**	**0.0654**	**0.0357**
Improved	3.302%	1.418%	3.354%	1.685%	4.513%	5.091%	5.825%	5.621%
	Toys and Games				Yelp			
PopRec	0.0039	0.0021	0.0070	0.0031	0.0045	0.0023	0.0091	0.0038
BPR	0.0030	0.0019	0.0047	0.0024	0.0023	0.0015	0.0039	0.0021
STAMP	0.0033	0.0019	0.0061	0.0028	0.0041	0.0025	0.0071	0.0035
GRU4Rec	0.0157	0.0097	0.0274	0.0134	0.0122	0.0076	0.0219	0.0107
NARM	0.0257	0.0174	0.0405	0.0221	0.0152	0.0095	0.0256	0.0129
S^3-Rec_{IPS}	0.0227	0.0146	0.0397	0.0200	0.0159	0.0094	0.0269	0.0129
BERT4Rec	0.0455	0.0307	0.0683	0.0380	0.0156	0.0096	0.0262	0.0130
SASRec	0.0479	0.0334	0.0698	0.0405	0.0161	0.0100	0.0290	0.0142
Pre-train	0.0487	0.0342	0.0630	0.0388	0.0166	0.0102	0.0278	0.0138
Fine-tuning	0.0492	0.0335	0.0701	0.0402	0.0168	0.0103	0.0287	0.0141
$RecGPT_1$	0.0515	0.0355	0.0711	0.0418	0.0172	0.0105	0.0293	0.0144
RecGPT	**0.0529**	**0.0359**	**0.0721**	**0.0419**	**0.0177**	**0.0107**	**0.0297**	**0.0146**
Improved	7.520%	4.971%	2.853%	3.457%	5.357%	3.883%	3.484%	3.546%

非序列化模型：PopRec 、BPR
序列化模型：STAMP 、GRU4Rec 、NARM 、S3RecIPS 、BERT4Rec 、SASRec
Pre-training：直接在测试集上预训练的模型
Fine-tuning：只使用与预训练阶段**相同**的用户数据进行微调

- 序列化模型更优。
- 使用transformer的（SASRec）的更优。（相比GRU 4 Rec，Caser，NARM和STAMP）
- 使用额外监督信号的并不一定优。（BERT4Rec和S3RecIPS不如SASRec。）
- 基于个性化提示的模型RecGPT1和RecGPT在统计测试中分别在88%和96%的情况下超过了最好的对比基线模型。

实验——实验二 参数分析

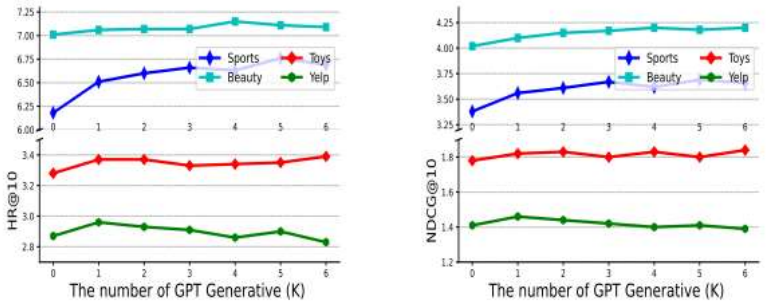


Fig. 4: Sensitive of the size of window K in term of HR@10 and NDCG@10.

超参数K的影响：

定义：K控制生成个性化提示窗口的大小，范围为[0, 6]。

结果：增加K的值会提升模型的性能，主要是因为个性化提示的有效相似性偏好匹配。

最佳性能：最佳性能在K的窗口大小为[1, 3]时获得。

原因：如果K的值过大，可能会生成一些无关的提示，从而削弱原始序列的相关性。

实验——实验二 参数分析

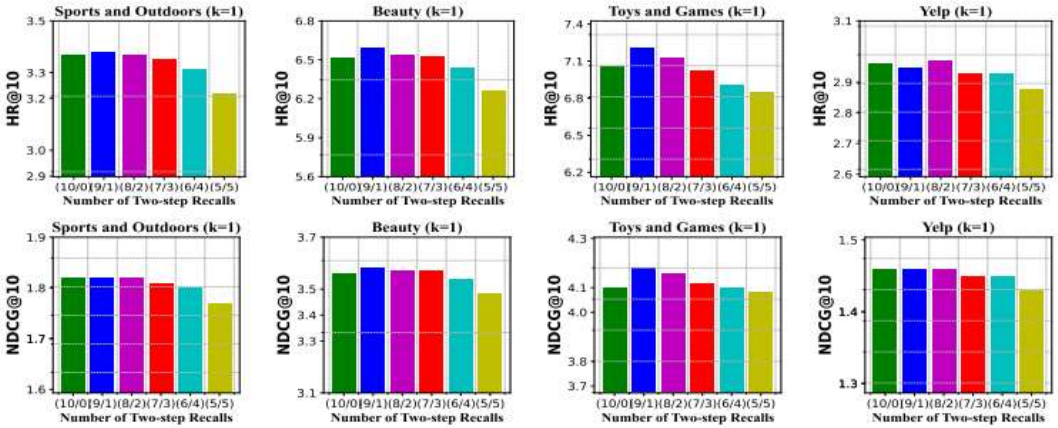


Fig. 5: Sensitive of parameter m_n in term of HR@10 and NDCG@10.

超参数mn的影响：
定义：mn控制两步召回的比例。

结果：报告的结果范围为(10, 0)到(5, 5)，主要关注HR@10指标。

最佳设置：在mn设置为**(9, 1)**和**(8, 2)**时，模型表现最佳。

原因：这一结果归因于序列数据的稀疏性，自回归召回需要大量数据。此外，序列数据中两个行为项目之间的联系最强，使得第一步生成的表示召回比第二步更为关键。

实验——实验三 消融实验

变体一	变体二	变体三
直接使用预训练模型进行两步召回，没有应用提示调整	将两步召回替换为传统的内积方法进行评估，即RecGPT1	同时移除了提示调整和两步召回。即，直接使用预训练模型一次召回K个。

Table 3: Performance of ablation on different components.

Datasets	Metric	RecGPT	Variant_1	Variant_2	Variant_3
Sports	HR@10	**0.0339**	0.0226	0.0333	0.0324
	NDCG@10	**0.0181**	0.0135	0.0180	0.0171
Beauty	HR@10	**0.0654**	0.0560	0.0651	0.0610
	NDCG@10	**0.0357**	0.0315	0.0356	0.0334
Toys	HR@10	**0.0721**	0.0603	0.0711	0.0630
	NDCG@10	**0.0419**	0.0364	0.0415	0.0388
Yelp	HR@10	**0.0297**	0.0261	0.0293	0.0278
	NDCG@10	**0.0146**	0.0133	0.0144	0.0138

个性化微调和两步召回对模型性能都有增益。（**RecGpt>2>3**）

只使用预训练模型而不进行微调时，两步召回效果就不好了。**(3>1)**

实验——实验四 线上A/B测试

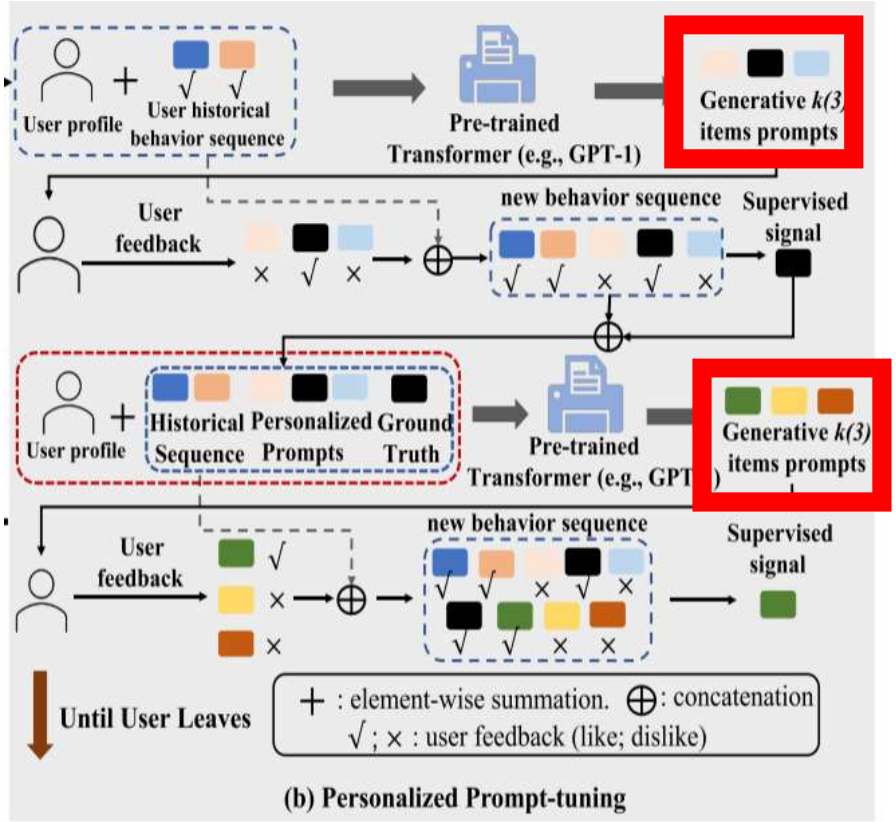
在快手视频APP推荐平台上进行[4]的实时A/B测试进一步验证了我们的方法。与基线ComiRec进行比较，实验时间5天。

模型在评论、转发、播放、关注和观看时间上分别增加了**+0.772%、+0.336%、+0.143%、+0.027%**和**+0.017%**。

.4 A/B测试是一种在线实验方法，通过将用户随机分配到不同的组别（实验组和对照组），来比较不同算法或功能的效果。

., 2024/11/11

展望



在未来的研究中，计划在公共数据集中引入强化策略，**以手动标注真实反馈**，用于我们提出的模型上的个性化提示。

南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

Thanks