

融合预训练语义初始化的序列推荐模型研究

课程名称 WEB 搜索与推荐系统导论

课程代码 M502075B

学号 25120323

姓名 柯劲帆

指导教师 刘华锋

提交日期 2026 年 1 月 4 日

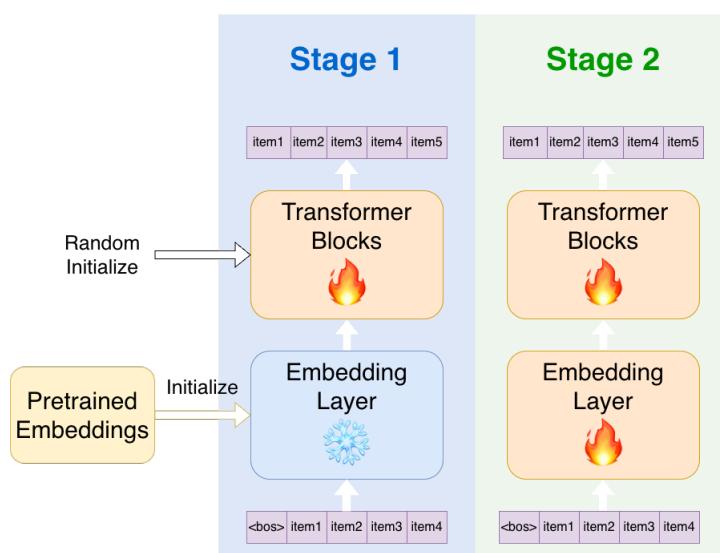


图 1: 融合预训练语义初始化的多阶段训练

1 引言

序列推荐 (Sequential Recommendation) 旨在根据用户历史交互序列预测其下一个感兴趣的物品。尽管基于 Transformer [13] 的 SASRec 模型在捕捉长距离依赖方面取得了显著成效 [5]，但仍面临许多挑战。其核心痛点在于语义缺失与冷启动，传统的 ID Embedding 采用随机初始化，完全依赖交互数据进行学习。在交互稀疏或新物品场景下，模型无法利用物品本身的丰富语义（如标题、描述），导致推荐准确率大幅下降；传统推荐模型尚未充分吸收 LLM 时代的架构红利，例如 SASRec 使用可学习位置编码，在捕捉序列中物品间相对位置关系上的效率较低，且外推性较差。

本项目提出了一种基于预训练语义初始化的序列推荐模型，旨在通过引入外部语义信息解决推荐系统中的冷启动与稀疏性问题。本项目使用预训练语义初始化方法为推荐模型注入外部知识，利用预训练语言模型（如 Qwen [16]、BERT [1]）提取物品标题与描述的深层语义特征，作为 ID Embedding 的初始化权重。针对未登录物品（Unknown Items），设计了基于共现邻居聚合的语义补全策略，有效缓解了冷启动问题；为了更好地适配语义特征并提升模型训练稳定性，本项目还集成多种现代 LLM 架构组件，比如使用旋转位置编码（RoPE）[11] 替代可学习位置编码，并使用 RMSNorm [15] 进行归一化，并在语义增强的基础上，进一步对比了门控多层感知机（gMLP）[6]、混合专家（MoE）[10] 等不同模型组件的效果，验证了语义初始化在不同架构下的鲁棒性。

本方案有如下特点和优势：

- 语义增强**: 通过引入预训练嵌入模型的语义向量，显著缓解了冷启动问题，提升了模型对物品内容的理解。在验证集上的实验表明，引入预训练语义初始化后，模型性能显著提升，最佳方案 (MRR@10: 0.1220) 相比随机初始化的 Baseline (MRR@10: 0.1100) 提升了约 **10.9%**。
- 多 LLM 组件实验**: 本项目使用多种先进的 LLM 组件进行实验，如旋转位置编码 (RoPE) [11]、门控多层感知机 [6]、SiLU 模块 [3] 和 DPO [8] 微调方法等。消融实验表明，RoPE 使得模型能够更好地处理变长序列，并具备更好的长度外推能力。

代码开源在Github。

2 相关工作

2.1 序列推荐模型

早期的序列推荐模型如 FPMC [9] 结合了马尔可夫链和矩阵分解。随着深度学习的发展，GRU4Rec [2] 将 RNN 引入该领域。SASRec (Self-Attentive Sequential Recommendation) 利用自注意力机制 [13] 并行捕捉长短期依赖，成为当前的 SOTA 基准 [5]。BERT4Rec [12] 则引入了双向注意力机制和完形填空任务。

2.2 预训练语言模型在推荐中的应用

近年来，利用 BERT [1]、RoBERTa [17] 等 PLM 提取物品文本特征以增强 ID Embedding 成为趋势。一些工作尝试直接用 LLM 进行推荐 (如 P5)，但推理成本高昂 [4]。本项目采用“PLM Feature as Initialization”的轻量级范式，既利用了语义信息，又保持了 ID 类模型的高效性。

不同于以往工作，本项目：1) 系统性地探究了 LLM 时代的架构组件 (RoPE, RMSNorm, SiLU, MoE) 在经典 SASRec 模型上的适用性；2) 提出了一种结合语义相似度和共现邻居的 Embedding 补全策略，有效处理了预训练特征缺失的“未知物品”。

3 方法

3.1 问题定义

给定用户 u 的历史交互序列 $S_u = \{x_1, x_2, \dots, x_n\}$ ，其中 x_t 表示用户在时刻 t 交互的物品。目标是预测用户下一个可能交互的物品 x_{n+1} 。

3.2 算法设计

本项目将每一个商品作为一个词元 (token)，算法实际上是根据用户历史交互词元序列，对下一词元的预测。

语义初始化层

不同于随机初始化 $E \sim \mathcal{N}(0, 1)$ ，我利用预训练模型 Φ (如 Qwen3-Embedding¹) 对物品标题 T_i 进行编码：

$$\mathbf{e}_i^{text} = \Phi(T_i)$$

对于未出现在预训练语料中的物品 (Unknown Items)，我利用其在训练集中的共现邻居集合 $N(i)$ 进行聚合补全：

$$\mathbf{e}_i^{text} = \frac{1}{|N(i)|} \sum_{j \in N(i)} \mathbf{e}_j^{text}$$

¹<https://huggingface.co/Qwen/Qwen3-Embedding-8B>

最终的 Embedding 矩阵 $E \in \mathbb{R}^{|I| \times d}$ 由 \mathbf{e}^{text} 初始化，并在训练过程中微调。

在实验中，模型分为两个训练阶段，如图1所示，第一阶段冻结使用预训练嵌入向量初始化的嵌入层参数，仅训练 Transformer Blocks 参数；第二阶段解冻嵌入层，训练全部模型参数。

仅解码器（Decoder-only）的下一商品预测网络

模型采用标准的 Decoder-only Transformer 架构 [7] 作为骨干网络。给定输入序列 Embedding E ，模型通过 L 层 Transformer Block 进行处理。每一层包含多头自注意力机制（Multi-Head Self-Attention, MHSA）和前馈神经网络（Feed-Forward Network, FFN），并在各子层之间采用 RMSNorm 进行归一化以提升训练稳定性。记 $\mathbf{H}^{(0)} = E$ ，则第 l 层输出为：

$$\mathbf{H}^{(l)} = \text{FFN}(\text{MHSA}(\mathbf{H}^{(l-1)})).$$

为了最大效率利用数据，训练时模型以并行的方式对批次序列中每个商品词元预测下一个商品词元。为了保证模型只依赖历史行为而不访问未来信息，在自注意力计算中引入因果掩码（Causal Mask），使得位置 t 只能关注自身及之前的序列位置，从而符合“根据过去预测未来”的序列建模约束。

在这种设置下，模型以自回归方式建模用户交互序列 $S_u = \{x_1, \dots, x_n\}$ 的条件概率分布：

$$P(S_u) = \prod_{t=1}^n P(x_t | x_1, \dots, x_{t-1}).$$

经过 L 层编码后，取位置 t 的隐藏表示 \mathbf{h}_t 与物品 Embedding 矩阵 E 做点积并经过 softmax 得到下一物品的预测分布：

$$P(x_t | x_1, \dots, x_{t-1}) = \text{Softmax}(\mathbf{h}_t E^\top).$$

训练阶段采用最大似然目标（等价于交叉熵损失），在所有序列与所有预测位置上最小化负对数似然：

$$\mathcal{L} = - \sum_u \sum_t \log P(x_t | x_1, \dots, x_{t-1}).$$

在推理阶段，取序列最后一个位置的表示 $\mathbf{h}_{|S_u|}$ 计算：

$$P(x_{next} | S_u) = \text{Softmax}(\mathbf{h}_{|S_u|} E^\top),$$

并选择概率最大的物品作为下一步推荐结果。

4 实验

4.1 实验设置

数据集

本项目使用课程提供的用户交互数据集，包含 55,223 条交互序列和 17,408 个物品。对于榜 2 提供的测试集，平均交互序列长度为 8.01，长度分布如图2所示。数据清洗后得到 16,869 个唯一的商品名称。

在实验中，本项目将原榜 2 测试集的用户交互记录按 80%:20% 划分，得到 49,701 条训练序列和 5,522 条验证序列。在训练时，只将每条验证序列的最后一个物品用于与模型预测结果进行比对验证，验证序列的其余物品与完整训练序列一起作为训练集对模型进行优化训练。

评价指标

本实验使用 MRR@10 作为评价指标。由于测试平台的提交次数有限，实验无法汇报完整的测试结果，只能使用验证集的评价指标作为模型比较的标准。

模型设置

实验使用的模型参考了 SASRec [5] 的实现并对关键组件进行了修改：1) 将可学习绝对位置编码修改为 RoPE 位置编码；2) 将带 ReLU 激活函数的 Point-Wise 前馈网络修改为带 SiLU 激活函数 [3] 的多层感知机。

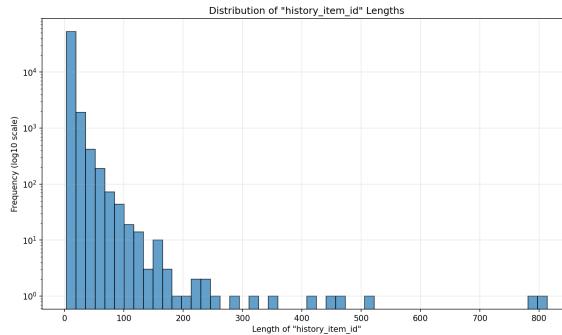


图 2: 数据集用户交互记录分布

实验使用 Qwen3-Embedding-8B² 获取商品语义信息的嵌入表示。

具体实验环境和训练超参数配置见附录A。

基线与消融实验

实验以 SASRec [5] 作为基线，但将原实现中的层归一化改为了效率更高的 RMSNorm [15]，且本实验中所有模型都沿用这一设置。

本项目针对模型中的关键组件进行了消融：

对于**预训练语义初始化**模块，我进行三组实验，分别是 1) 随机初始化，2) 仅使用商品名称的语义嵌入初始化和 3) 使用增强数据的语义嵌入初始化。如图3所示，增强数据使用 wikipediaapi³ 和 steam-games-dataset⁴ 对原始商品信息进行了增强以得到更加丰富的语义信息。其中使用 Qwen3-32B⁵ 将检索到的 wikipedia 脚条信息解析为与 steam-games-dataset 对齐的结构化信息。附录B展示了一些增广数据样例。

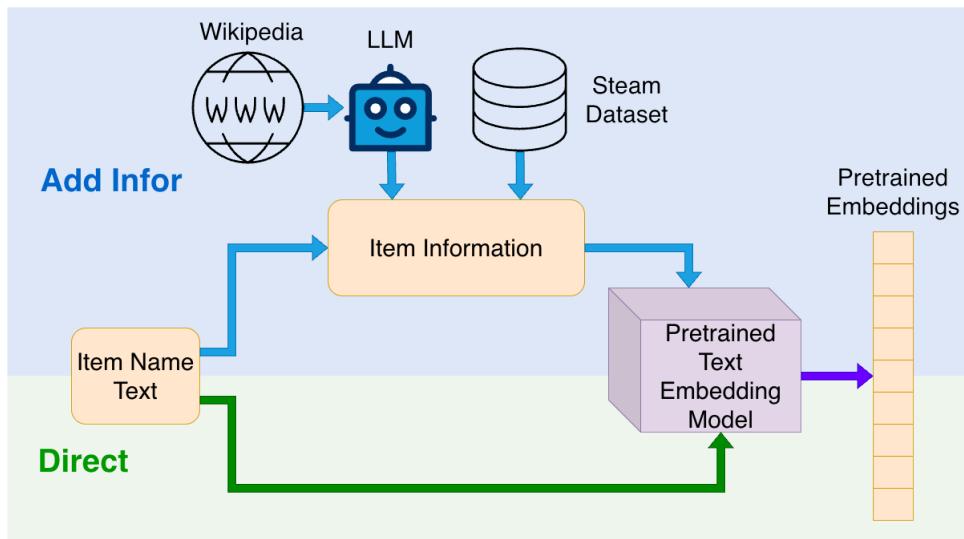


图 3: 数据增强流程

对于**模型架构组件**，我分别 1) 将 RoPE 位置编码替换为原 SASRec 中可学习的位置编码，2) 将带 SiLU 激活函数的多层感知机修改为带门控模块的多层感知机（参考 Qwen3 [14] 的实现），3) 将多层感知机修改为 4 专家（2 激活专家）的 MoE 模块。为了保证公平，上述模型除修改部分外参数量一致、训练数据量一致。

对于**训练范式**，我还尝试使用 DPO [8] 在最佳模型的基础上进行继续优化。我仅对训练集中每条序列

²<https://huggingface.co/Qwen/Qwen3-Embedding-8B>

³<https://github.com/goldsmith/Wikipedia>

⁴<https://huggingface.co/datasets/FronkonGames/steam-games-dataset>

⁵<https://huggingface.co/Qwen/Qwen3-32B>

的最后一个商品进行偏好优化训练。对于负样本的选取，我使用最佳模型的嵌入层来选取与正样本相似的商品。在构造 DPO 数据集时，按照 30%:70% 的比例选取困难负样本（相似度 top10）和简单负样本（相似度 top11 到 top200）。

4.2 实验结果

表4.2展示了不同变体在验证集上的 MRR@10 表现。

模型变体	描述	MRR@10
SASRec(Baseline)	随机初始化 + Learnable PE	0.1100
Ours	预训练嵌入(商品标题) + RoPE + SiLU MLP	<u>0.1220</u>
Add-info	预训练嵌入(商品增强数据) + RoPE + SiLU MLP	0.1184
Learnable PE	预训练嵌入(商品标题) + Learnable PE	0.1187
Gated MLP	预训练嵌入(商品标题) + RoPE + Gated MLP	0.1208
MoE	预训练嵌入(商品标题) + RoPE + MoE	0.1139
+ DPO	Ours + DPO	0.1076

图4展示了两次测试集打榜排名的结果。其中第二次排名结果对应表4.2中的“Learnable PE”模型变体 (0.1187)。

34	顾晓露	A班	0.07894265789300496	78.51%
35	陈涛	B班	0.07892681304574929	78.47%
36	柯劲帆	A班	0.07282403898306763	64.67%
37	马剑	A班	0.06928827215888973	56.68%
38	彭菲	A班	0.0667524587897442	50.94%

(a) 第一次排名第 36

#	姓名	小组	分数	提升
1	江盛贝	A班	0.08569119032241311	34.78%
2	柯劲帆	A班	0.08388755087956513	31.95%
3	范希凡	A组	0.08258595236801786	29.90%
4	张鑫豪	A班	0.0824964379633636	29.76%

(b) 第二次排名第 2

图 4: 两次测试集打榜排名结果

5 讨论

预训练初始化对性能提升决定性作用 实验数据显示，相比于 Baseline (0.1100)，仅引入预训练语义初始化 (Learnable PE, 0.1187) 即带来了巨大的性能飞跃 (+7.9%)。这有力地证明了在推荐系统中，利用 PLM 提取的通用语义信息能够有效弥补 ID Embedding 在稀疏数据下的表达缺陷，是提升模型性能的核心关键。

RoPE 的辅助增益 在预训练初始化的基础上，引入 RoPE (Main, 0.1220) 进一步提升了约 2.8% 的性能。这说明虽然语义初始化贡献了主要的性能增益，但更好的位置编码机制能帮助模型更精准地利用这些语义特征进行序列建模。

信息增广的噪声 Add-info (0.1184) 表现不如仅使用标题的 Main (0.1220)。这可能是因为外部检索的信息（如 Wiki、Steam 详情）引入了与用户行为不相关的噪声，或者 Embedding 模型未能很好地压缩复杂信

息。

模型结构的复杂性: 1) **门控 MLP** (0.1208) 表现接近但略逊于 Standard MLP, 说明在当前数据规模下, 增加参数量带来的收益有限, 甚至可能导致轻微过拟合; 2) **MoE** (0.1139) 表现较差。这可能是因为 MoE 训练难度较大, 且在推荐系统的稀疏数据上, Router 难以学习到稳定的负载均衡策略, 导致专家利用率不均。

DPO 的负面影响 在 SFT 基础上进行 DPO 训练导致性能下降。我认为源于数据集提供的用户交互序列过短, 构造的偏好对(正负样本)噪声较大。也有可能是因为 DPO 的目标函数与推荐系统的 Ranking 目标存在不一致。

6 总结

本项目重点研究了预训练语义初始化对序列推荐模型性能的影响。通过利用 PLM 提取物品文本特征并初始化 Embedding 层, 我成功解决了传统 ID 类模型在稀疏数据和冷启动场景下的痛点。实验结果表明, 语义初始化是性能提升的主要驱动力, 配合 RoPE 等现代架构组件, 模型取得了显著优于 SASRec 基线的表现。我发现, 相比于盲目堆叠复杂的模型结构(如 MoE), 提升底层特征的语义表达能力更为关键。

下一步工作可以尝试 1) 探索更高效的文本编码方式, 以减少 Add-info 中的噪声; 2) 研究针对推荐系统的 MoE 路由策略, 解决稀疏性导致的路由坍塌问题; 3) 优化 DPO 的样本构造策略。

7 建议

实验报告模板建议提供 L^AT_EX 版本。

参考文献

- [1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.
- [2] Michal Dobrovolny, Ali Selamat, and Ondrej Krejcar. Session based recommendations using recurrent neural networks - long short-term memory. In *Intelligent Information and Database Systems: 13th Asian Conference, ACIIDS 2021, Phuket, Thailand, April 7–10, 2021, Proceedings*, page 53–65, Berlin, Heidelberg, 2021. Springer-Verlag.
- [3] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018. Special issue on deep reinforcement learning.
- [4] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems, RecSys '22*, page 299–315, New York, NY, USA, 2022. Association for Computing Machinery.
- [5] Trong Dang Huu Ho and Sang Thi Thanh Nguyen. Self-attentive sequential recommendation models enriched with more features. In *Proceedings of the 2024 8th International Conference on Deep Learning Technologies, ICDLT '24*, page 49–55, New York, NY, USA, 2024. Association for Computing Machinery.

- [6] Hanxiao Liu, Zihang Dai, David R. So, and Quoc V. Le. Pay attention to mlps. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc.
- [7] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.
- [8] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: your language model is secretly a reward model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.
- [9] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 811–820, New York, NY, USA, 2010. Association for Computing Machinery.
- [10] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.
- [11] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C), February 2024.
- [12] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 1441–1450, New York, NY, USA, 2019. Association for Computing Machinery.
- [13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [14] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025.
- [15] Biao Zhang and Rico Sennrich. *Root mean square layer normalization*. Curran Associates Inc., Red Hook, NY, USA, 2019.
- [16] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025.

- [17] Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. A robustly optimized BERT pre-training approach with post-training. In Sheng Li, Maosong Sun, Yang Liu, Hua Wu, Kang Liu, Wanxiang Che, Shizhu He, and Gaoqi Rao, editors, *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China, August 2021. Chinese Information Processing Society of China.

A 实验环境及超参数设置

本实验使用 4 张 RTX3090 完成，最佳模型训练耗时约 2 分钟。

如图2所示，超过 90% 的数据序列长度都在 20 个商品以内，因此将长度大于 20 个商品的序列截断靠前的子串。

模型的词表中除了商品词元外，还增加了 <PAD> 和 <BOS> 等额外 4 个特殊词元，词表大小为 17,412。在预测 top_10 推荐物品词元时排除特殊词元。

模型的隐藏向量为 1024 维，MLP 的中间维度为 4096 维。使用 5 层 Transformer Block 堆叠，注意力模块使用 8 头注意力，每头 64 维。Dropout 比例为 0.2。模型参数量约为 0.08B（消融实验中 MoE 模型总参数量约为 0.2B，激活 0.012B）。

训练的批大小为 1024，优化器为 AdamW，使用混合精度 bf16 训练。阶段一训练 4 epochs，学习率为 5.0e-4；阶段二训练 8 epochs，学习率最高为 5.0e-4，使用带学习率预热的余弦学习率调度器。

训练使用 HuggingFace 的 Transformers Trainer⁶和 TRL⁷框架。

B 数据增强细节

数据增强组件中，商品标题和对应增广数据样例为：

```
Item Title:  
"Tomb Raider II"  
  
Augmented Information:  
{  
    "release_year": 1997, "price": "Unknown", "genres": ["Action", "Adventure"],  
    "developers": ["Core Design"], "item_type": "Game", "platform": ["PC", "PlayStation"]  
}  
  
-----  
  
Item Title:  
"Sony PlayStation 2 DualShock 2 Controller - Black"  
  
Augmented Information:  
{"release_year": "Unknown", "price": "Unknown", "genres": ["Unknown"],  
    "developers": ["Unknown"], "item_type": "Hardware", "platform": ["PlayStation"]  
}
```

⁶<https://github.com/huggingface/transformers>

⁷<https://github.com/huggingface/trl>