长序列检索器：面向推荐的超长序列基础候选检索

Qin Ren
renqin.97@bytedance.com
ByteDance
Beijing, China

Zheng Chai†
chaizheng.cz@bytedance.com
ByteDance
Hangzhou, China

Xijun Xiao
xiaoxijun@bytedance.com
ByteDance
Beijing, China

Yuchao Zheng†
zhengyuchao.yc@bytedance.com
ByteDance
Beijing, China

Di Wu
di.wu@bytedance.com
ByteDance
Beijing, China

摘要

精确建模用户超长序列对于工业推荐系统至关重要。当前的方法主要集中在利用超长序列进行排名阶段，而候选检索阶段的研究仍相对较少。本文提出了 LongRetriever，一个将超长序列融入推荐器检索阶段的实际框架。具体而言，我们提出了上下文训练和多上下文检索，这使用户序列与候选项目之间实现特定的交互，并在基于搜索的范式下确保训练和服务的一致性。在一个大规模电子商务平台上进行的广泛在线 A/B 测试表明，该框架带来了统计上显著的改进，证实了其有效性。目前，LongRetriever 已经完全部署到该平台中，影响数十亿用户。

CCS Concepts

• **Information systems** → **Recommender systems**.

Keywords

长序列建模，候选匹配，工业推荐系统

ACM Reference Format:

Qin Ren, Zheng Chai†, Xijun Xiao, Yuchao Zheng†, and Di Wu. 2018. 长序列检索器：面向推荐的超长序列基础候选检索. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 介绍

在限定时间内从庞大的项目库中检索出高度相关项，是工业推荐系统面临的重要挑战 [3, 13]。主要的解决方案采用双阶段方法 [6, 7]，其中初始阶段候选检索专注于缩小候选项目的范围，从而使得后续在第二阶段排名中能够部署更加精确但计算成本较高的模型。

†Corresponding Authors.

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

一般来说，基于流行度的过滤和协同过滤在候选检索阶段已被广泛采用。随着深度学习的进步，基于嵌入的检索（EBR）已成为主导范式 [8, 10]。EBR 使用双塔深度神经架构分别将用户和项目特征编码为密集向量表示。它在一个判别性学习框架内优化这些向量之间的距离度量（例如余弦相似度），区分相关项目（正样本）与不相关项目（负样本）。在部署过程中，项目向量预先计算并使用近似最近邻搜索技术 [1]（例如 HNSW[11]）进行索引，从而实现在线服务时以亚线性时间高效检索前 k 个项目。

尽管在效率和有效性之间存在有利的权衡，但由于计算资源和在线延迟要求的限制，EBR 通常只处理用户编码器中的短用户序列（例如，最近的 200 个行为）。此外，由于分离的双塔学习范式，候选项目与用户侧历史行为序列进行交互通常是困难的，这为深入理解用户兴趣提供了显著的空间。精确建模用户序列对于当前推荐系统至关重要，在这种情况下，典型的方法通常采用目标注意力机制来建模用户兴趣，其中最具代表性的方法包括 DIN[15], DIEN[14], CAN[2] 等。随着计算能力的进步和用户在线行为的不断累积，最近的研究转向了在推荐系统中实现对超长序列更全面建模 [5, 12]。

尽管广受欢迎，但需要注意的是，目前的长序列建模主要针对推荐系统中的排序阶段进行开发。如何利用超长序列来增强候选者检索仍然是工业推荐器中一个显著的空白。一般来说，在利用超长序列进行候选者检索时会遇到以下挑战：

- 超长序列与候选项目之间充分的交互。请注意，流行的基于目标注意力的方法适用于排序阶段，因为候选项目的数量有限（在工业界大约是几百个）。而在匹配阶段，候选项目的规模达到数百亿级别，使得超长序列与候选项目之间的交互在实践中变得不可行。
- 多样而精确的兴趣建模，具有可控的学习能力以及超长序列。传统的双塔范式通常将用户表示编码为单一的向量，这在工业规模上限制了对多样化兴趣的建模，现有的多兴趣模块 [4, 9] 一般缺乏具体的语义，导致可控制性降低。

为应对上述挑战，本文提出了一种可控且准确的多样化兴趣学习框架长检索器，用于超长序列中的候选匹配。LongRetriever 将互斥的项目类别定义为可解释的用户兴趣，并在共享模型参数的情况下对每个兴趣进行独立的候选训练。它采用基于搜索的机制从用户的终身行为序列中筛选出与各兴趣相关的子序列。该方法通过不同的兴趣领域表示多个密集向量来描述用户，而不会增加训练负担。利用展示出强大排序阶段有效性的终身用户行为序列，LongRetriever 还能够在候选匹配阶段实现更精确的兴趣表达。总体而言，主要贡献总结如下：

- 我们设计了 LongRetriever，一个基于长序列的可控多兴趣学习框架。据我们所知，这是首次尝试在候选匹配阶段引入超长序列。
- 我们设计了新颖的上下文训练和多上下文服务，以引入用户-候选交互并确保基于搜索范式下的训练和服务一致性。结合兴趣选择，这使得整个检索过程完全可解释且可控。
- 大型电子商务平台上针对数十亿用户进行的广泛在线 A/B 测试验证了 LongRetriever 的有效性。该框架现已在工业规模上全面部署，影响数十亿用户。

2 方法论

2.1 背景：基于嵌入的检索

令 U 和 I 分别表示用户集和项目集。给定一个具有原始行为序列 $B_u = [b_1^u, b_2^u, \cdots, b_T^u]$ 的用户 $u \in U$，用户基本特征 P_u 包括用户资料和上下文特征，以及目标项目 $v \in I$。基于嵌入的检索框架使用一个用户编码器 f 和一个项目编码器 g 将用户和项目的特征投影到低维稠密向量：$e_u = f(B_u, P_u)$ 和 $e_v = g(v)$。用户 u 与项目 v 的相关性，用 r_{uv} 表示，是通过 x_u 和 x_v 之间的距离度量（通常是余弦相似度）来量化。在训练过程中，相关性分数 r_{uv} 通过批内对比损失进行优化，以区分由 u 互动的项目和同一小批量内的未互动项目。在部署期间，所有项目向量 $\{e_v\}\, v \in I$ 都预先计算并使用近似最近邻搜索技术进行索引。因此，只有用户向量 e_u 需要

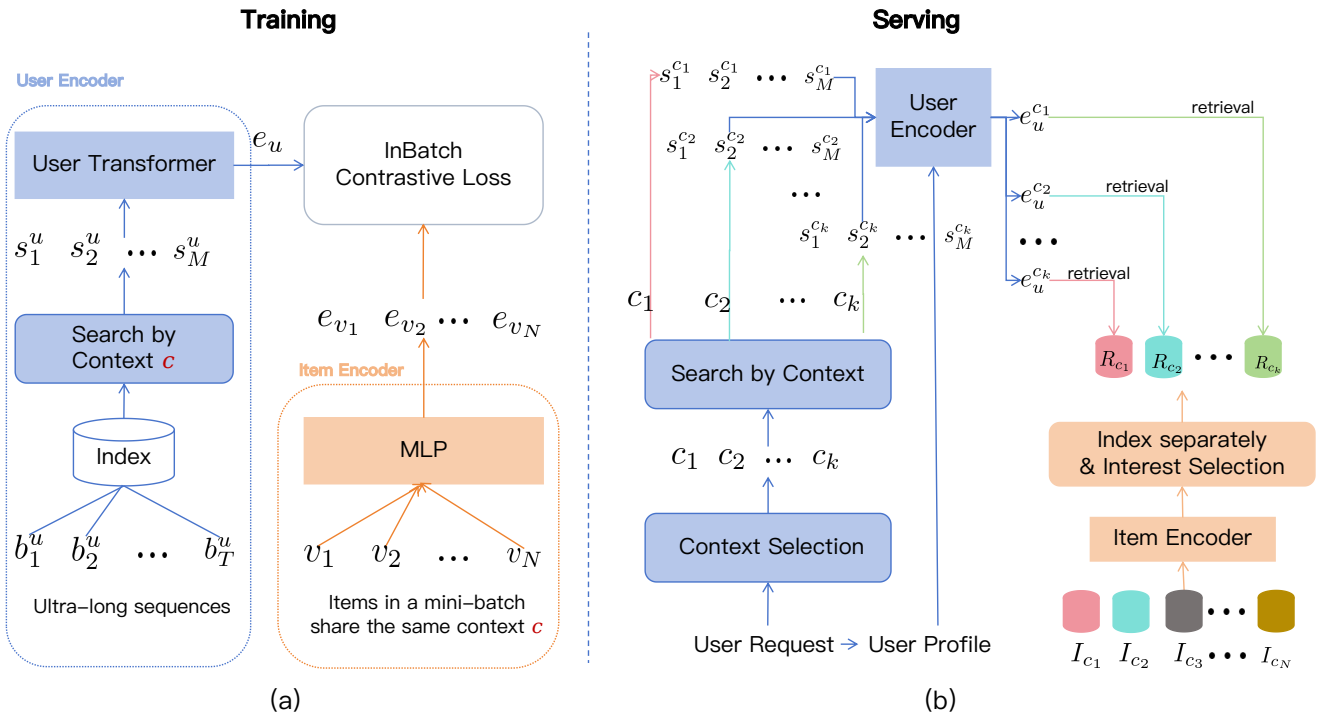

图 1: 长检索器模型框架。

实时计算，而 ANN 系统可以从项目仓库中以亚线性时间检索出与 e_u 最近的 k 个项目。

2.2 总体框架

图 1 说明了我们 LongRetriever 模型的整体架构。通常，给定用户终身行为列表 $B_u = [b_1^u, b_2^u, \cdots, b_T^u]$，其中 T 是用户行为的长度。LongRetriever 首先采用类别匹配来过滤来自 B_u 的最近 L 个行为，这些行为与候选项目的上下文完全匹配，类似于 SIM 中 Hard-Search 方法中的 [12]。这里，上下文可以是任何可以在实践中依赖的标准。在本文中，我们采用项目的类别进行硬搜索。基于上下文的搜索后，新的用户子序列为 $S_u = [s_1^u, s_2^u, \cdots, s_M^u]$。LongRetriever 随后使用一个用户变换器模型作为用户编码器 f 来提取用户向量 e_u。此 Transformer 的输入标记组成包括：- 一个 $[CLS]$ 标记以汇总所有标记的信息。- 来自用户档案特征 $P_u = [p_1^u, p_2^u, \cdots, p_K^u]$ 的 K 标记，用于增强行为序列和基本用户特征之间的细粒度交互。- L 过滤子行为 S_u。在将可学习的位置编码和标记类型嵌入添加到输入标记后，LongRetriever 使用标准的 Pre-Layer Normalization Transformer 来建模这些标记。$[CLS]$ 标记的最终表示被采纳为用户表示向量 e_u：

$$e_u = Transformer([CLS, p_1^u, p_2^u, \cdots, p_K^u, s_1^u, s_2^u, \cdots, s_M^u]) \tag{1}$$

值得注意的是，EBR 的训练需要独立计算用户和项目编码器，以满足部署期间 ANN 理念的要求。因此，候选项目特征从我们的 User Transformer 中排除。

基于模型结构设计，在 LongRetriever 中设计了两个关键组件，即上下文训练和多上下文检索。

2.3 上下文训练

直接利用基于搜索的长序列得到以下损失函数：

$$\mathscr{L}(e_u) = -\log\left(\frac{e^{e_u^{\top} e_v}}{e^{e_u^{\top} e_v} + \sum_{v' \in \mathscr{B}} e^{e_u^{\top} e_{v'}}}\right) \tag{2}$$

其中 $\mathcal{B}$ 表示批量内的负样本集合，v 表示正样本，而 v' 指示任意一个负样本。

虽然利用终身行为序列在离线评估指标上取得了显著改进，但这种改进被证明是不可靠的，因为 S_u 在 e_u 中隐含地包含了候选项目的部分信息。这在批次内对比目标下引入了数据泄露问题，在此情况下实例之间相互影响。考虑用户 u 与项目 v 之间的交互实例，其中 S_u 和 v 的上下文，即项目的类别为“食品”。这对 (u, v) 被视为正例，而与当前 mini-batch 中其他实例（可能属于“衣服”或“宠物”等类别）中的项目配对的 u 则被视为负例。在这些条件下，模型可以通过利用 S_u 的类别是否与 v 匹配来轻松降低 batch 内对比损失，从而人为地提升离线指标。

为了缓解这种数据泄露，LongRetriever 采用了一种上下文训练策略。上下文训练基本上涉及重新组织每个小批量中的样本，使用缓存机制确保同一小批量内的子序列 S_u 和候选项目 v 只共享一个兴趣类别。

在上下文训练之前，EBR 的目标等同于判断用户是否会购买整个样本空间中的“食品”类别下的商品，给定特定类别的序列 S_u。经过上下文训练后，EBR 的学习目标变为判断用户是否会购买候选商品，而非整个“食品”类别样本空间中的商品，给定特定类别的序列 S_u。

2.4 多上下文检索

上下文训练允许用户和候选者在训练期间进行交互。而在服务阶段，由于用户塔和项目塔的分离推理，推断出几个用户可能感兴趣的类别以基于用户的超长行为序列实现多上下文检索具有重要意义。在这里，设计了两个必要的阶段来估计用户兴趣和多兴趣检索。

兴趣选择与表示。LongRetriever 根据用户的历史行为选择特定兴趣，并多次执行用户编码器以生成每个兴趣的不同用户表示。由于计算限制，每次请求所选的兴趣数量不应过多。为此，我们提供了一个简单的自动兴趣选择策略，随机在顶部。它通过使用时间加权求和汇总跨兴趣的历史行为数据来计算每个兴趣 c_i 的用户参与度分数：

$$Score_{c_i} = \sum_{b_i^u \in B_u} \frac{sign(c_i = c_{b_i^u})}{t_{b_i^u}} \tag{3}$$

其中 $c_{b_i^u}$ 表示与行为 b_i 相关的兴趣类别，而 $t_{b_i^u}$ 代表行为 b_i 发生时间戳与当前请求时间戳之间的时间差（以天为单位）。随机在顶部首先根据参与度分数从整个兴趣集合中选择 $top - M$ 个兴趣，然后对每个用户请求从这个子集中随机选择 N 个兴趣。

多兴趣检索。在上下文训练之后，用户向量专门用于在一个特定兴趣领域内挖掘相关项目。因此，LongRetriever 根据类别将整个项目库划分为多个独立的子库，并为每个子库构建索引。在接收到用户请求时，LongRetriever 跨与选定兴趣相关的子库执行多次检索，并使用预定义策略合并结果。

3 实验

3.1 实验设置

我们在购物商城中评估了 LongRetriever，这是一个真实的、大规模的工业推荐场景。与目标排序阶段不同，工业匹配系统非常复杂，通常包括数十种不同的检索策略。因此，仅靠离线指标不足以评估检索效果。我们首先在工业数据集上训练模型，然后进行为期 7 天的真实用户在线 A/B 测试以进行评估。训练集包含来自我们的购物商城的用户交互日志，涵盖 9 亿用户、1.5 亿商品以及超过 100 亿个样本，这些样本跨越了连续 400 天的时间段。每个样本包括用户的特征（例如，用户 ID、性别）、候选商品、从用户原始终身行为序列中根据候选商品类别过滤出的 50 个行为子序列——该原始序列平均包含 2 万多个交互，并且有一个标签指示是否存在交互。我们将 LongRetriever 与广泛采用的多兴趣检索策略 MIND[9] 进行基准测试，该策略已在我们的生产环境中部署。LongRetriever 采用 Top 20 中的随机 5 个策略来获得 5 个兴趣。

表 1: 在线 A/B 测试结果对业务指标的影响

PV	UV CTR	UV CVR	Orders Per User	Exposed Categories Per User	Clicked Categories Per User
+0.62%	+0.17%	**+1.33%**	**+1.70%**	**+0.14%**	**+1.39%**

表 2: 在线 A/B 测试结果在中间指标上

Model	AER	UER	CTR*CVR
MIND	**31.82%**	4.58%	6.89%
LongRetriever	18.72%	**9.29%**	**7.92%**

表 3: 随机兴趣的消融研究

Model	AER	UER	CTR*CVR
Random5 in Top20	18.72%	**9.29%**	7.92%
Top5	**30.43%**	7.23%	**8.29%**

表 4: 终身行为序列的消融研究

Model	AER	UER	CTR*CVR
LongRetriever	**18.72%**	**9.29%**	**7.92%**
w/o Long Sequence	11.45%	5.92%	5.33%

3.2 实验结果

表 1 总结了 LongRetriever 与 MIND 在我们的购物中心中的性能对比。由于商业政策，我们仅报告业务指标的相对增长，并对具有统计显著性结果的项目加粗（$p < 0.05$）。就转换效率而言，LongRetriever 相比 MIND 提高了 0.62%的商品页面浏览量 (PV)，0.17%的独特访客 (UV) 点击率 (CTR)，以及 1.33%的独特访客转化率 (CVR)。这共同导致了每位用户平均订单数增长了 1.7%。关于生态系统发展，LongRetriever 通过提高每位用户的平均曝光类别数量 0.14%，和每位用户的平均点击类别数量 1.39%，增强了推荐的多样性。

表 2 展示了检索项目的中间指标，揭示了表格 1 中业务指标改进的潜在原因。所有暴露比率（AER）用于衡量推荐器对模型检索的所有结果的偏好程度。另一方面，唯一暴露比率（UER）用于衡量推荐系统对于只能通过此模型检索到的项目偏好的程度。LongRetriever 的 AER 显著低于 MIND，主要是因为 MIND 会根据每个用户向量从完整的项目库中检索，而 LongRetriever 仅使用 5 个类别进行多上下文检索，严重限制了其候选范围。然而，相比 MIND，LongRetriever 实现了 4.71%更高的 UER。这表明 LongRetriever 检索到的项目与现有其他策略检索到的项目高度不同。此外，通过 CTR * CVR 衡量的 LongRetriever 的独特转化效率比 MIND 提高了 1.03%，从而促进了业务指标的增长。

3.3 消融研究

随机兴趣。表 3 比较了两种不同的兴趣选择策略。用户兴趣分布通常表现出长尾特征。在仅保留前 5 个兴趣的选择中，与 Top 中的随机相比，AER 增加了 11.71%。此外，转化效率提高了 0.37%。然而，UER 减少了 2.06%。因此，整体表现劣于 Top 中的随机。此外，Top 中的随机固有的显式多样性结构对生态系统发展有积极贡献。

终身行为序列。表 4 进行了关于终身行为序列影响的消融研究，同时保留了上下文训练和多上下文检索。在没有有效终身行为序列的情况下，LongRetriever 难以有效地完成所有负例都是硬负例的上下文训练范式，导致所有后验度量标准都有显著下降。

上下文训练和上下文检索。表 5 分析了上下文训练和上下文检索的影响，同时保留了终身行为序列。移除上下文组件会在 LongRetriever 的训练中引入数据泄漏。尽管这些组件的缺失会导致所有曝光比率和唯一曝光比率显著增加，但检索到的候选对象在转化效率方面的表现平平。

表 5: 消融研究在上下文组件中的应用

Model	AER	UER	CTR*CVR
LongRetriever	18.72%	9.29%	**7.92%**
w/o In-Context	**38.92%**	**13.48%**	5.61%

4 结论

本文提出了 LongRetriever，以解决在实际工业中建模更可控和准确的多样化用户兴趣的挑战。LongRetriever 采用独立的上下文训练和多上下文检索来处理每个兴趣。LongRetriever 通过基于搜索的方法整合了用户的长期行为序列。这将单个固定长度的用户向量转换为其各自兴趣领域内的多个精确向量，且几乎不需要额外的训练开销。在我们的业务场景中进行的广泛在线 A/B 测试验证了 LongRetriever 的有效性。该框架现已大规模部署，影响数十亿用户。

在未来的工作中，用户信息可以通过大型语言模型进行处理，生成优化的分类分配，从而实现更准确和多样化的推荐。

References

[1] Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. 2020. ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Information Systems* 87 (2020), 101374.

[2] Weijie Bian, Kailun Wu, Lejian Ren, Qi Pi, Yujing Zhang, Can Xiao, Xiang-Rong Sheng, Yong-Nan Zhu, Zhangming Chan, Na Mou, et al. 2022. CAN: feature co-action network for click-through rate prediction. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 57–65.

[3] Jon Nicolas Bondevik, Kwabena Ebo Bennin, Önder Babur, and Carsten Ersch. 2024. A systematic review on food recommender systems. *Expert Systems with Applications* 238 (2024), 122166.

[4] Zheng Chai, Zhihong Chen, Chenliang Li, Rong Xiao, Houyi Li, Jiawei Wu, Jingxu Chen, and Haihong Tang. 2022. User-aware multi-interest learning for candidate matching in recommenders. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1326–1335.

[5] Zheng Chai, Qin Ren, Xijun Xiao, Huizhi Yang, Bo Han, Sijun Zhang, Di Chen, Hui Lu, Wenlin Zhao, Lele Yu, et al. 2025. LONGER: Scaling Up Long Sequence Modeling in Industrial Recommenders. *arXiv preprint arXiv:2505.04421* (2025).

[6] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[7] Jiri Hron, Karl Krauth, Michael Jordan, and Niki Kilbertus. 2021. On component interactions in two-stage recommender systems. *Advances in neural information processing systems* 34 (2021), 2744–2757.

[8] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2553–2561.

[9] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 2615–2623.

[10] Sen Li, Fuyu Lv, Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng, Xiao-Ming Wu, and Qianli Ma. 2021. Embedding-based product retrieval in taobao search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3181–3189.

[11] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence* 42, 4 (2018), 824–836.

[12] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2685–2692.

[13] Deepjyoti Roy and Mala Dutta. 2022. A systematic review and research perspective on recommender systems. *Journal of Big Data* 9, 1 (2022), 59.

[14] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.

[15] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.