

赞同 36

分享

## 华为2024：运用LLM优化双塔召回系统的生成式推荐



SmartMindAI

专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注我

已关注

36 人赞同了该文章

### introduction

推荐系统<sup>+</sup>被广泛应用于解决信息过载的问题，旨在从大量商品集合中为用户识别出感兴趣的商品。现代推荐系统通常会集成表示学习和构建搜索索引，以优化匹配过程。首先，使用例如双塔架构和序列推荐模型等模型，用户和商品在共享潜在空间中被编码为潜在表示。随后，为了高效地为用户检索top-k个商品，例如使用Faiss和SCANN等工具构建了近邻搜索索引。尽管取得了显著的进展，表示学习和索引构建的两个阶段通常独立运行，这限制了实现端到端优化的能力，从而影响了推荐系统的整体效果。

为了克服这一局限性，许多研究已经做出了很多尝试。一个突出的方向是构建基于树的匹配索引，这不仅可以优化内容匹配模型，还可以优化树结构索引。然而，这些方法往往面临着树结构导致的推理效率低和内容语义信息利用有限的挑战。最近，生成检索已经作为一种有希望的新范式出现在信息检索领域，并且最近被应用于生成推荐中。

与传统的基于表示的用户-内容匹配方法不同，这种范式采用了一种端到端的生成模型，以自回归<sup>+</sup>的方式直接预测候选内容标识符。具体来说，这些方法首先将每个内容 $x$ 分词为一组离散的语义代码，然后使用编码器-解码器模型（例如Transformer）作为检索的端到端索引。在这个设置中，编码器编码用户与内容之间的交互历史，而解码器预测下一个内容 $x_t$ 的代码序列 $C_t$ 。整个框架的示意图<sup>+</sup>见图

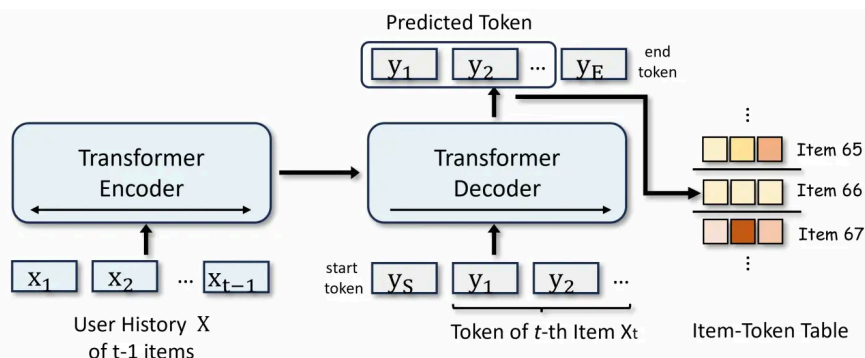


Figure 1: A framework of generative recommendation.

然而，现有的生成推荐方法在利用内容信息时存在一个显著的缺点，往往只专注于行为或语义的某一方面，忽略了行为与语义之间的互补关系。一方面，预训练模态编码器的进步，如BERT和ViT，促进了多模态特征的整合，增强了先验知识<sup>+</sup>，并在多模态推荐模型中找到了广泛的应用。另一方面，行为数据<sup>+</sup>通过交互序列捕捉了用户特定的偏好，使其在推荐场景中特别有效。相反，语义信息<sup>+</sup>提供了更广泛的、不受偏见的见解，促进了不同领域内的更好泛化。

在本文中，我们提出了一种新的两流生成推荐器模型--EAGER，它结合了行为和语义的合作。我们从以下几个方面分析了在统一生成框架中建模行为和语义的挑战，并提出了相应的解决策略：

- 多模态特征的利用：利用预训练的多模态编码器来处理文本和视觉信息，以增强推荐系统的多模态能力。
- 用户偏好与内容特性的平衡：设计算法以平衡用户偏好和内容特性，确保推荐既满足个性化需求，又具有广泛的适用性。

通过这些方面的研究，我们旨在开发一个既能考虑用户行为又能整合内容语义信息的推荐系统，以提供更加精确和全面的推荐结果。首先，构建一个联合生成架构来处理两种不同类型的内容至关重要。考虑到行为和语义特征空间的固有差异，直接在编码器端通过特征融合来整合它们存在挑战，这在之前的双塔模型中已经得到证明。因此，我们的方法在解码器端分别构建了独立的行动代码和语义代码，采用了一种两路生成架构，其中每种类型的信息都作为独立的监督信号。这个架构包括一个共享编码器用于编码用户交互历史，以及两个独立的解码器分别预测行为和语义代码，从而避免了过早的特征交互。在推断阶段，我们通过利用每个流的预测熵作为内容排名的置信度度量，来增强两个流的结果融合，确保有效的预测。其次，确保充足的独立学习对于充分利用每种类型信息的潜力至关重要。之前的工作通常采用自回归方法，逐个学习每个token，专注于离散和局部信息，而未能捕捉到全局洞察。

在EAGER中，我们引入了一个全局对比任务，使用摘要token。这个模块从两个主要来源：（1）传统的双塔模型使用对比学习来获取区分性内容特征。我们旨在让解码器模型不仅具备自回归生成能力，还能够掌握全局的区分性能力，从而在对比学习的框架下增强内容特征的提取；（2）Transformer模型利用特殊token来封装全局信息，促使我们在token序列末尾添加一个摘要token。这个token以单向的方式总结积累的知识，作为知识提炼的焦点。最后，虽然单独的解码和预测重新排序已经显示出有效性，但整合微妙的互动可以增强知识流的共享。正如前面提到的，直接在特征级别进行交互往往导致不理想的结果。因此，我们引入了一个精心设计的语义引导的转移学习任务，以促进隐式知识的交流。具体来说，我们提出语义信息可以指导行为方面，并设计了一个辅助Transformer，具有重建与识别双目标：重建目标涉及使用全局语义特征预测被掩码的行为token，而识别目标旨在判断行为token是否与指定的全局语义特征相匹配。通过这些目标，这个模块间接地通过Transformer模型优化了行为和语义特征之间的交互。

我们的主要贡献包括：1. 构建了联合生成架构，以独立处理行为和语义信息。2. 引入了全局对比任务与摘要token，以增强全局信息的捕捉和局部信息的整合。3. 设计了语义引导的转移学习任务，通过Transformer模型优化了行为和语义特征之间的交互。4. 在推断阶段，通过预测熵的利用，增强了行动和语义代码的融合，确保了有效的预测。

## Method

### Problem Formulation

给定用户的所有内容集合 $\mathcal{X}$ 和交互历史

$$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{t-1}\} \in \mathcal{X}$$

序列推荐系统返回一个候选内容列表，用于预测下一内容 $\mathbf{x}_t$ 。在生成式推荐中，每个内容 $\mathbf{x}$ 的标识符表示为一个序列代码

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_l] \in \mathcal{Y}$$

其中 $l$ 是代码的长度。生成模型的目标是学习一个映射

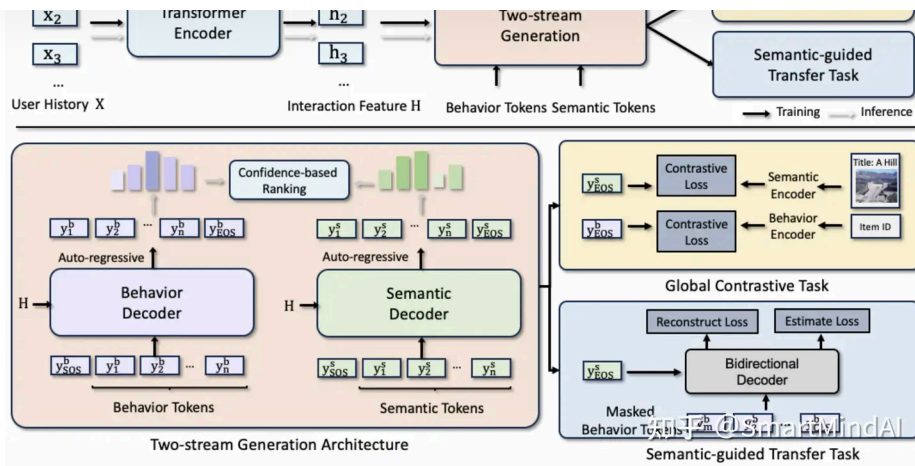
$$f: \mathcal{X} \rightarrow \mathcal{Y}$$

它接受用户交互过的内容序列作为输入，并生成内容代码（候选标识符）。在训练过程中，模型首先将用户的历史交互数据 $\mathbf{X}$ 输入编码器，然后利用自回归解码器逐步生成内容代码 $\mathbf{Y}$ 。交互的概率可以通过以下公式计算得出：

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{i=1}^l p(\mathbf{y}_i|\mathbf{X}, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{i-1})$$

在推理阶段，解码器选取top-n候选时，会对序列代码进行束搜索。

### Overall Pipeline



我们的整体EAGER框架如图。EAGER包括：（1）一个双流生成架构，用于统一行为和语义信息的内容推荐，（2）一个带有摘要标记的全局对比任务，用于捕获全局知识以提高自回归生成质量（3）一个语义引导的转移任务，以实现跨信息和跨解码器的交互。

首先，在我们的双流生成架构中，我们使用编码器建模用户交互记录并获取交互特征。然后，我们提取行为和语义特征以构建两种代码，并利用两个解码器分别以自回归的方式预测它们。同时，我们在全局对比任务中优化一个摘要标记，并利用它在语义引导的转移任务中提高跨解码器的交互。经过训练后，我们采用基于信心的排名策略，将两个不同预测的结果合并。

### Two-stream Generation Architecture

处理两种不同的信息，即行为和语义，我们利用了转换器模型的强大建模能力，并设计了一个两流生成架构。这个框架由一个共享编码器、两个独立的代码和解码器组成。

用户互动历史序列  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots\}$  的序列建模\*基于堆叠的多头自注意力层与前向传播层，这是转换器模型提出的方法。值得注意的是，我们只采用了共享编码器，而不是两个编码器，这足以后续的独立解码生成丰富的表示。我们通过编码器表示了历史互动特征 $\mathbf{H}$ 。

首先，我们使用两个预训练模型\*提取行为和语义内容嵌入 $\mathbf{E}^b$ 和 $\mathbf{E}^s$ 。行为编码器是一个双塔模型（例如DIN），仅用于推荐的ID序列，而语义编码器是一个通用模态表示模型（例如Sentence-T5）。通过两个提取的嵌入，我们分别应用了层次聚类\*算法，每个聚类均匀地分为K个子聚类，直到每个子聚类仅包含单一内容。如图所示，我们可以获得两种编码 $\mathbf{Y}^b$ 和 $\mathbf{Y}^s$ ，分别对应行为和语义。

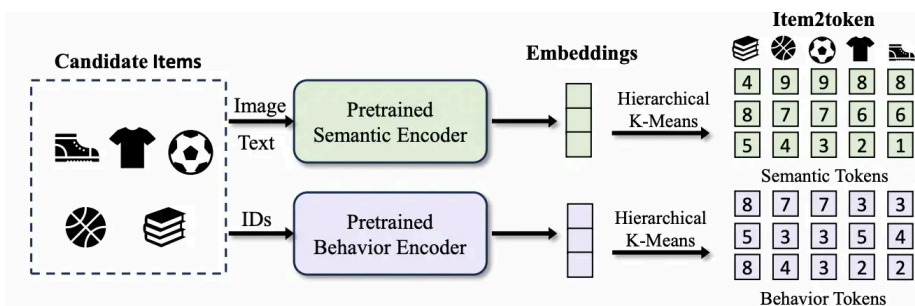


Figure 3: The illustration of dual codes.

为了处理两种不同的编码，我们采用两个单独的解码器分别对每种编码进行解码和生成预测，使得每个解码器专注于处理一种编码。这样的设计降低了监督差异，提高了并行生成的效能。在训练过程中，我们在编码序列 $\mathbf{Y}$ 的开头添加一个开始标记 $\mathbf{y}_{\text{SOS}}$ ，构建解码器输入

$$\bar{\mathbf{Y}} = \{\mathbf{y}_{\text{SOS}}, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t\}$$

并使用交叉熵损失\*进行预测。总的生成损失 $\mathcal{L}_{\text{gen}}$ 等于基于 $b$ 和 $s$ 的两个生成损失 $\mathcal{L}_{\text{gen}}^b$ 和 $\mathcal{L}_{\text{gen}}^s$ 的总和。假设 $\mathcal{L}_{\text{gen}}$ 是总的生成损失 $\mathcal{L}_{\text{gen}}^b$ 和 $\mathcal{L}_{\text{gen}}^s$ 分别对应基于 $b$ 和 $s$ 的生成损失。

## 知乎

为了赋予每个生成解码器足够的区分能力，我们设计了一个带有摘要标记的全局对比任务，以提炼全局知识。对于每个解码器的输入 $\hat{\mathbf{Y}}$ ，我们考虑自回归生成的从左到右顺序，并在序列末尾插入一个可学习的结束标记 $\mathbf{y}_{[\text{EOS}]}$ ，构建修改后的输入

$$\hat{\mathbf{Y}} = \{\mathbf{y}_{\text{SOS}}, \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_l, \mathbf{y}_{[\text{EOS}]}\}$$

这种设计鼓励前一个标记学习更全面和区分能力更强的知识，从而使最终的标记能够进行总结。在更新过程中，摘要标记上的梯度可以反向传播到前一个标记。

为了使摘要标记捕获全局信息，我们采用对比学习范式来从预训练的编码器中提取内容嵌入 $\mathbf{E}^b$ 和 $\mathbf{E}^s$ 并传递给当前训练的模型。这里我们采用仅正样本的对比度量，而不是通常使用的Info-NCE，以实现这一目标。全局损失 $\mathcal{L}_{\text{con}}$ 由两个损失 $\mathcal{L}_{\text{con}}^b$ 和 $\mathcal{L}_{\text{con}}^s$ 的总和给出，其中每个损失都是：

$$\mathcal{L}_{\text{con}} = \mathcal{L}_{\text{con}}^b + \mathcal{L}_{\text{con}}^s$$

$$\mathcal{L}_{\text{con}}^t = \mathcal{F}(\mathbf{y}_{[\text{EOS}]}^t, \mathbf{E}^t), \quad t \in \{b, s\}$$

$\mathbf{y}_{[\text{EOS}]}^t$ 表示摘要标记的嵌入 $\mathbf{y}_{[\text{EOS}]}^t$ ，而 $\mathcal{F}(\cdot, \cdot)$ 是度量函数，用于度量两个元素之间距离或相似度，例如平滑的 $\ell_1$ 范数。

### Semantic-guided Transfer Task

通过上述组件，我们的模型能够有效地利用两种类型的信息进行预测。然而，我们不采用完全独立的解码方式，而是进一步提出了一种语义引导的转移任务，利用语义知识来指导行为学习。

为了在双方之间建立知识流动的同时避免直接交互，我们构建了一个独立的双向Transformer解码器作为辅助模块。我们首先在行为代码的开头添加一个标记行为代码的 $\mathbf{CLS}$ 标记，以获得序列行为代码的 $\mathbf{CLS}$ 标记，行为代码的1，行为代码的2，，行为代码的 $l$ 作为解码器的输入。然后，我们将语义摘要标记的嵌入语义摘要标记输入到交叉注意力中，使解码器中的每个行为标记都能够关注全局的语义特征。我们表示输出特征为行为代码的 $\mathbf{CLS}$ 标记，行为代码的1，行为代码的2，，行为代码的 $l$ 。为了进行我们的转移训练，我们设计了两个后续的目标：重建和识别。

**重建。**我们通过语义全局特征来重建被遮罩的行为代码，旨在使每个行为标记都能从语义中受益。在重建训练中，我们随机遮罩行为代码中的 $m\%$ 的标记，以获得遮罩代码行为代码的 $\mathbf{CLS}$ 标记，行为代码的1，行为代码的遮罩标记，，行为代码的 $l$ ，其中行为代码的遮罩标记表示被遮罩的标记。然后，我们获取对应的输出特征行为代码的 $\mathbf{CLS}$ 标记，行为代码的1，行为代码的遮罩标记，，行为代码的 $l$ ，并应用对比损失来促进重建：

$$\mathcal{L}_i = \log \frac{\exp(\mathbf{r}_{[m_i]}^{b+} \cdot \mathbf{y}_i)}{\exp(\mathbf{r}_{[m_i]}^{b+} \cdot \mathbf{y}_i) + \sum_{j=1}^J \exp(\mathbf{r}_{[m_i]}^{b+} \cdot \mathbf{y}_j)},$$

$$\mathcal{L}_{\text{recon}} = -\frac{2}{N} \sum_{i=1}^{N/2} \mathcal{L}_i$$

其中 $\mathbf{y}_i$ 是第 $i$ 个被遮罩的真实token特征 $\mathbf{y}_j$ 是采样的真实token特征。

**识别。**此外，我们还构建了一个二元分类器来判断行为代码是否与语义全局特征相关或无关。在识别训练中，我们通过随机替换行为代码中的 $m\%$ 的token为采样的无关token来构造负样本，例如，将 $23, 123, 32$ 中的123替换为145，得到 $23, 145, 32$ 。我们对输出 $\mathbf{CLS}$ 的token添加一个线性层，并使用具有sigmoid激活的线性层来计算正样本与负样本的得分 $s^+/s^-$ 。识别的二元交叉熵损失被用于计算：

$$\text{Loss} = -(y \log(s) + (1 - y) \log(1 - s))$$

其中， $y$ 是真实的标签， $s$ 是分类器的输出。

$$\mathcal{L}_{\text{recog}} = -\log(s^+) + \log(1 - s^-)$$

### Training and Inference



$$\mathcal{L}_{\text{EAGER}} = \mathcal{L}_{\text{gen}} + \lambda_1 \mathcal{L}_{\text{con}} + \lambda_2 (\mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{recog}})$$

其中 $\lambda_1$ 和 $\lambda_2$ 是损失系数。

由于我们从行为流和语义流中获得了两个结果，我们首先从每个流中通过beam search获取前 $k$ 个预测。使用 $2k$ 个预测代码，我们计算代码的对数概率作为每个预测的信心分数，这类似于语言模型中使用的困惑度<sup>+</sup>，较低的值表示更高的信心。最后，我们按照信心分数对这些预测进行排序，得到前 $k$ 个预测结果，这对应 $k$ 个预测内容。

Experimental Setting

数据集。我们在这四个开源数据集上进行实验，这些数据集常用于序列推荐任务。对于所有数据集，我们按照用户对交互记录进行分组，并按照交互时间戳的升序进行排序。之后，我们仅保留了核心数据集中的5个，这些数据集过滤了交互记录少于五条的不受欢迎的内容以及不活跃用户。表中列出了这些数据集的统计信息。

评估指标。在匹配阶段，我们采用了两种被广泛使用的评估标准，即召回率<sup>+</sup>和归一化累计增益（NDCG）。我们报告了在推荐候选者的前5/10/20个中计算的指标。遵循标准评估流程，我们采用留一法进行评估。对于每个内容序列，最后的内容用于测试，倒数第二个内容用于验证，其余的用于训练。在训练过程中，我们将每个用户的内容历史限制在20个以内。

实施细节。对于双流生成架构，我们设置了编码器层的数量为1，解码器层的数量为4。遵循先前的研究工作，我们采用预训练的DIN作为行为编码器，Sentence-T5作为语义编码器，并将隐藏大小设置为128，如在中报告的那样。在层次聚类K-means算法中，我们设置聚类数量 $k$ 为256。对于全局对比性任务，我们采用平滑 $l_1$ 距离作为提炼损失。对于语义引导的转移任务，我们随机屏蔽50%的行为代码用于重建，并随机替换50%的行为代码为采样代码，以构建识别的负对。为了训练我们的模型，我们采用学习率0.001的Adam优化器，并采用温升策略以实现稳定的训练。因为GCT和STT任务收敛速度很快，在超参数设置上，EAGER对这些任务不敏感。因此，损失系数 $\lambda_1, \lambda_2$ 都被设置为1。

Performance Comparison (RQ1)

Dataset	Metric	Traditional			Transformer-based			Tree-based		Generative		Improv.
		GRU4REC	Caser	HGN	SASRec	Bert4Rec	S^3-Rec	TDM	Recforest	TIGER	EAGER	
Beauty	Recall@5	0.0164	0.0205	0.0325	0.0387	0.0203	0.0387	0.0442	0.0470*	0.0454	<b>0.0618</b>	31.49%
	Recall@10	0.0283	0.0347	0.0512	0.0605	0.0347	0.0647	0.0638	0.0664*	0.0648	<b>0.0836</b>	25.90%
	Recall@20	0.0479	0.0556	0.0773	0.0902	0.0599	0.0994	0.0876	0.0915*	-	<b>0.1124</b>	13.08%
	NDCG@5	0.0099	0.0131	0.0206	0.0249	0.0124	0.0244	0.0323	0.0341*	0.0321	<b>0.0451</b>	32.26%
	NDCG@10	0.0137	0.0176	0.0266	0.0318	0.0170	0.0327	0.0376	0.0400*	0.0384	<b>0.0525</b>	31.25%
	NDCG@20	0.0187	0.0229	0.0332	0.0394	0.0233	0.0414	0.0438	0.0464*	-	<b>0.0599</b>	29.09%
Sports	Recall@5	0.0129	0.0116	0.0189	0.0233	0.0115	0.0251	0.0127	0.0149*	0.0264	<b>0.0281</b>	6.44%
	Recall@10	0.0204	0.0194	0.0313	0.0350	0.0191	0.0385	0.0221	0.0247*	0.0400	<b>0.0441</b>	10.25%
	Recall@20	0.0333	0.0314	0.0477	0.0507	0.0315	0.0607	0.0349	0.0375*	-	<b>0.0659</b>	8.57%
	NDCG@5	0.0086	0.0072	0.0120	0.0154	0.0075	0.0161	0.0096	0.0101*	0.0181	<b>0.0184</b>	1.66%
	NDCG@10	0.0110	0.0097	0.0159	0.0192	0.0099	0.0204	0.0110	0.0133*	0.0225	<b>0.0236</b>	4.89%
	NDCG@20	0.0142	0.0126	0.0201	0.0231	0.0130	0.0260	0.0141	0.0164*	-	<b>0.0291</b>	11.92%
Toys	Recall@5	0.0097	0.0166	0.0321	0.0463	0.0116	0.0443	0.0305	0.0313*	0.0521	<b>0.0584</b>	12.09%
	Recall@10	0.0176	0.0270	0.0497	0.0675	0.0203	0.0700	0.0359	0.0383*	0.0712	<b>0.0714</b>	0.28%
	Recall@20	0.0301	0.0420	0.0716	0.0941	0.0358	<b>0.1065</b>	0.0442	0.0483*	-	0.1024	-3.85%
	NDCG@5	0.0059	0.0107	0.0221	0.0306	0.0071	0.0294	0.0214	0.0260*	0.0371	<b>0.0464</b>	25.07%
	NDCG@10	0.0084	0.0141	0.0277	0.0374	0.0099	0.0376	0.0230	0.0285*	0.0432	<b>0.0505</b>	16.90%
	NDCG@20	0.0116	0.0179	0.0332	0.0441	0.0138	0.0468	0.0284	0.0310*	-	<b>0.0538</b>	14.96%
Yelp	Recall@5	0.0152	0.0151	0.0186	0.0162	0.0051	0.0201	0.0181	0.0220*	0.0212*	<b>0.0265</b>	20.45%
	Recall@10	0.0263	0.0253	0.0326	0.0274	0.0090	0.0341	0.0287	0.0302*	0.0367*	<b>0.0453</b>	12.69%
	Recall@20	0.0439	0.0422	0.0535	0.0457	0.0161	0.0573	0.0422	0.0449*	0.0552*	<b>0.0724</b>	11.56%
	NDCG@5	0.0099	0.0096	0.0115	0.0100	0.0033	0.0123	0.0121	0.0119*	0.0146*	<b>0.0177</b>	3.51%
	NDCG@10	0.0134	0.0129	0.0159	0.0136	0.0045	0.0168	0.0154	0.0154	0.0214	<b>0.0241</b>	10.41%
	NDCG@20	0.0178	0.0171	0.0212	0.0182	0.0063	0.0226	0.0208	0.0210*	0.0230*	<b>0.0311</b>	19.62%

表列出了四个数据集的整体性能。

- EAGER几乎在不同的数据集上达到比基础模型更好的结果。特别是，在Beauty基准测试中，EAGER的表现比第二好的基线模型高出31.49%，在召回率@5方面提高了高达20.45%，在NDCG@5方面提高了高达3.51%与TIGER相比。
- EAGER在大多数数据集上超越了先前的生成模型。它与现有的模型不同，通过双流解码器架构和多任务训练，实现了对行为与语义信息的整合，从而更深入地理解行为-语义关系，并捕获跨代码的关键全局信息。这些显著的提升验证了我们设计的有效性，并强调了行为与语义信息整合的必要性。

此外，在实际场景中，构建ANN索引主要侧重于实现快速匹配，这可能导致由于优化目标不一致而导致的额外性能下降。然而，通过利用束搜索策略来直接预测内容代码，生成方法能够克服这一挑战，从而增强模型的鲁棒性<sup>+</sup>，使其能够更好地应对这种情况。

原文《EAGER: Two-Stream Generative Recommender with Behavior-Semantic Collaboration》

发布于 2024-08-02 16:33 · IP 属地北京

华为 LLM 推荐系统



理性发言，友善互动

1 条评论

默认 最新



瀚海星云

华为会发这种质量的论文？第一单位肯定不是华为  
08-14 · 安徽

回复 喜欢

推荐阅读

RAG+微调：企业LLM大模型的护城河

围绕LLM的炒作是前所未有的，但这是有道理的，生成式 AI 有潜力改变我们所知道的社会。在很多方面，LLM将使数据工程师变得更有价值——这令人兴奋！不过，向老板展示数据发现工具或文本到...

汇智网

llm 是泡沫？

这篇文章不谈技术，只想聊聊自己这两年从事 llm 工作的一些感悟。我个人对 llm 是一种很复杂的态度。毕竟，它真的击碎了我 2023 年之前所有的技术积累，以前在 nlp 苦心钻研的训练经验、模...

ybq

Figure 2: Overall framework of CTRLA. It contains two steps: Step 1 trains *honesty* and *confidence* probes. Step 2 directly utilizes the learned probes to control and monitor LLM behavior at inference. The *honesty* probe manipulates the representation of LLM to make it more honest, while *confidence*...

对LLM如何平衡内外部知识的一些思考-RAG，LLM，LLM...

Initialll

Leaderboards



来自 AI Secure 实验室的安全排行榜简介

Hugging Face