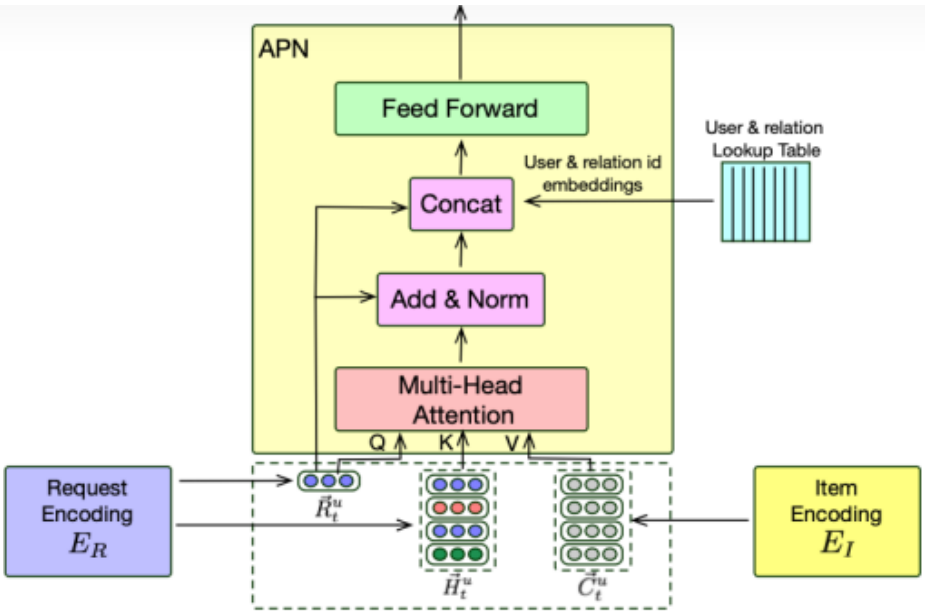
知乎



一探究竟：基于APN架构如何革新通用信息检索，打造更智能的搜索体验

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

1 人赞同了该文章

Introduction

本文提出了一种新的通用信息检索框架-该框架采用双编码器架构，第一个编码器学习给定信息访问请求的信息访问功能的稠密向量；第二个编码器学习每个信息项集合中的稠密表示。通过将非个性化的预训练和个性化的微调策略相结合，以及从不同来源获取硬负例样本的策略，我们旨在实现在各种信息访问场景下的更好性能。首先，利用文本描述实现信息访问功能，构建通用且可扩展的框架；其次，在真实世界的电商网站用户与多种信息访问系统交互的数据集上评估模型，并对亚马逊ESCII数据集进行进一步测试，证明该模型在所有三种信息访问功能上明显优于竞争对手。同时，我们还展示了对该信息访问功能进行单独建模时，最多可以将NDCG提升至45%\@10。

- 提出一个通用且可扩展的信息检索框架。
- 研究一个单一模型在执行信息访问功能方面的可能性。
- 一种两阶段训练与负采样结合的注意力个性化网络，用于提升个性化检索性能。
- 评估模型实际效果并对比竞争基准，验证增益显著。

我们开源了我们的UIA框架实现，旨在促进研究者开发通用的信息访问。

Related Works

Personalized Information Access

本文讨论了个性化在信息检索和推荐系统中的重要性。对于信息检索，HEM和ZAM是常用的个性化模型，它们使用doc2vec和注意力机制来建模用户、查询和项的表示。对于推荐系统，早期的模型如协同过滤忽略了用户历史交互的顺序，后来的深度学习模型如GRU4Rec、SASRec和BERT4Rec可以捕捉用户历史偏好并预测下一个项目的预测。

Dense Retrieval

大规模预训练语言模型与近邻搜索结合，发展出密集检索模型，通过微调预训练模型并优化对比损失，在下游信息检索任务上表现优于BM25等词汇匹配方法。优化密集检索模型的方法有：DPR使用BM25负样本进行硬负采样；ANCE应用自采样策略进行负采样；RocketQA和Condenser使用

知乎

Joint Search and Recommendation

最新研究显示，联合建模搜索和推荐能取得更好结果。JSR是基于共享表示学习网络上添加任务特定层的框架，通过多任务学习进行优化。最近，SRJGraph扩展了类似方法，使用图卷积网络捕捉用户-查询-项高阶交互。这些模型对特定任务高度专业，不能简单应用于其他信息访问功能。相比之下，我们提出了一种密集检索模型(UIA)，统一了各种信息访问功能，包括搜索和推荐，可以在不引入额外参数的情况下训练多种任务。此外，（APN），能捕获用户序列交互历史并显著优于JSR和SRJGraph。

Methodology

Task Formulation

我们提出一种新的评分函数，该函数结合了这三个输入变量，旨在提供最有效的信息访问结果。我们的方法包括建模用户需求，优化搜索策略，并利用推荐系统提高信息检索效率。我们将通过实验验证我们的模型的有效性。

1. 信息访问请求（$\mathcal{R}$）：包含搜索查询、情境上下文（如位置和时间）和短期上下文（如会话数据）。可以为空（零查询检索）。
2. 用户历史：$\mathcal{H}$包含用户发出$\mathcal{R}$的用户信息，例如个人信息或长期交互历史。
3. $\mathcal{I}$：候选检索项信息，包括文本内容、作者和来源。

注意：信息访问功能可扩展至多模态信息（如图像查询与视频项）、上下文请求（如会话与对话）及零查询检索/推荐。

1. 给定一个简短的文本查询，检索相关项目。
2. 求解用户指定样例的相似项
3. 参考物品推荐：配合指定物品，推荐相关联的其他物品。

将表中的描述视为一个函数f，其中四个输入变量分别是$\mathcal{F}_t^u$、$\mathcal{R}_t^u$、$\mathcal{H}_t^u$和$\mathcal{I}_i$，对应的输出是模型参数θ。其中$\mathcal{F}_t^u$是用户在时刻t正在使用的信息访问功能$\mathcal{R}_t^u$是用户的查询请求或给定锚点项的文本内容$\mathcal{H}_t^u$是用户过去的历史交互记录，而$\mathcal{I}_i$则是候选物品的信息，只包含其文本内容。$\mathcal{R}_t^u$分为两种类型

Overview of the UIA Framework

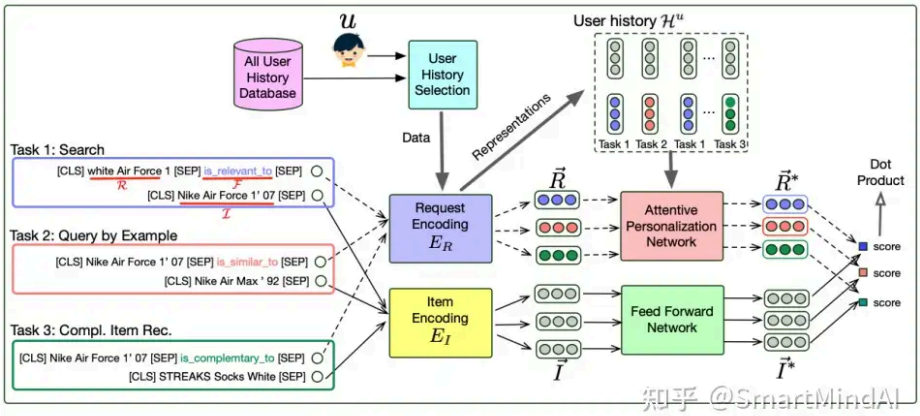


Figure 1: A high-level overview of the UIA framework.

该框架采用双编码架构，每个用户都有一套编码器和解码器，用于生成和处理信息访问请求。编码器通过用户的交互历史数据生成个性化请求矢量，解码器则通过内积计算编码请求和候选项内容之间的相似度。该框架在后续章节中进行了实现和优化。

UIA Architecture

入token的一维密集向量表示。在本文中，我们使用BERT-base+对每个信息访问功能($\mathcal{F}_t^u$)和请求($\mathcal{R}_t^u$)进行编码，公式如下：

$$\mathbf{E}_{\mathcal{F}_t^u}, \mathbf{E}_{\mathcal{R}_t^u} = \mathbf{E}(input)$$

$$\vec{R}_t^u = \mathbf{E}_{\mathcal{R}}(\texttt{[CLS]}\ \mathcal{R}_t^u\ \texttt{[SEP]}\ \mathcal{F}_t^u\ \texttt{[SEP]})$$

图示为一个请求编码输入的例子。候选项目编码采用BERT-base模型预训练的方式，表示每个候选项$\mathcal{I}_i$。编码过程如下：

$$\vec{I}_i = \mathbf{E}_{\mathcal{I}}(\texttt{[CLS]}\ \mathcal{I}_i\ \texttt{[SEP]})$$

$$Q_j = \vec{R}_t^u . \theta_j^Q, \quad K_j = H_t^u . \theta_j^K, \quad V_j = C_t^u . \theta_j^V$$

$$\mathbf{Attn}(Q_j, K_j, V_j) = \mathbf{softmax}(\frac{Q_j K_j^T}{\sqrt{l}})V_j$$

注意力函数输出被concatenate，输入Add & Norm层，接着与用户和功能向量concatenate，并经过ReLU激活层得到当前请求的个人化表示$\vec{R}_t^{u*}$。最后，通过前馈网络调整候选项向量的表示，得到最终结果$\vec{I}_i^*$。

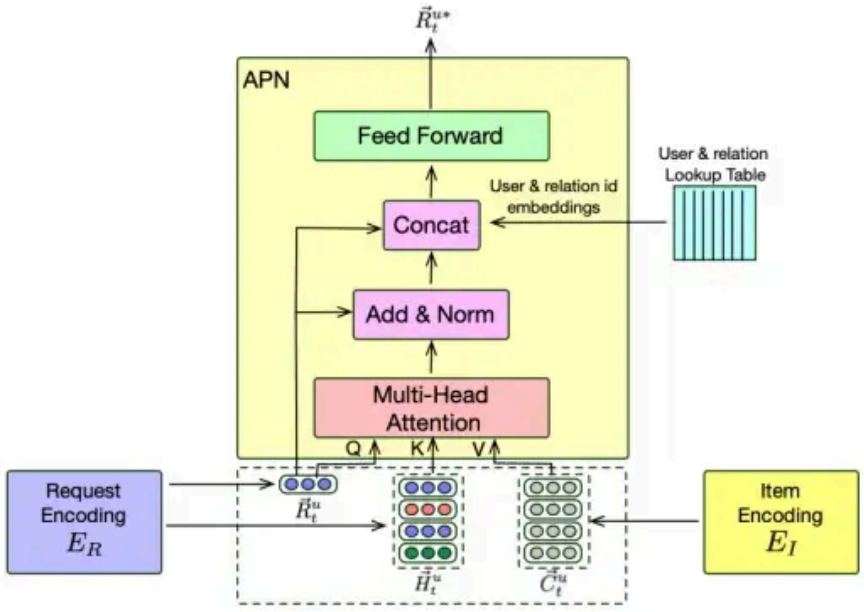


Figure 2: The Attentive Personalization Network (APN).

UIA Optimization

我们提出了一种两阶段优化流程：(1) 非个性化的预训练，(2) 个性化的微调。因为现实世界系统常常需要处理大量新用户和冷启动用户，这些用户缺乏历史交互数据，因此无法进行个性化。但我们可以先在这些用户的原始数据上进行非个性化的预训练。 通过聚合所有用户的训练数据构建非个性化训练集。对于每个训练实例

$$(\mathcal{F}_k, \mathcal{R}_k, \mathcal{I}_k, \mathcal{Y}_k)$$

其中$\mathcal{Y}_k$是地面真相标签。获取请求编码器和候选项目编码器的输出向量，即$\vec{R}_k$和$\vec{I}_k$。使用点积计算非个性化匹配请求和候选项目：$\vec{R}_k \cdot \vec{I}_k$。 训练数据只包含正例，需要适当的负采样。采用两阶段的负采样和训练策略。

第一阶段：对训练数据中的每个请求，都随机从排序后的前200个items中抽取负面样本。将负样本与正训练实例的比例设置为1。使用交叉熵损失函数+来训练模型。

数据中的请求和随机负样本。这种自否定采样的策略已在一些密集检索模型，如ANCE和RANCE中成功应用。将负样本与正训练实例的比例设置为1。再次使用交叉熵损失函数来训练模型，同时使用in-batch负样本。

在个性化预训练中，我们只调整$\mathbf{E}_{\mathcal{R}}$和$\mathbf{E}_{\mathcal{I}}$的参数。然后，我们添加框架的个性化部分并重新创建训练数据，包括用户的个人信息和他们过去的交互。接下来，我们使用每个请求和候选物品的个性化表示，即$\vec{R}_k^*$和$\overrightarrow{I_k^*}$（如图所示）。我们通过点积计算它们的匹配分数：

$$\vec{R}_k^* \cdot \overrightarrow{I_k^*}$$

为了负采样，我们利用了BM25结果以及在批量内选择的负样本（与非个性化预训练的第一阶段类似）。最后，我们使用交叉熵损失函数进行训练。

Experiment Settings

Dataset

1 精确匹配（E），表示该项目与查询相关联；(2) 替代（S），表示该项目与查询相关但不完全匹配（部分相关）；(3) 补充（C），表示该项目与查询无关，但可以补充相关项的标签E；(4) 不相关（I）。让$I_E(q)$ $I_S(q)$和$I_C(q)$分别表示所有具有标签E，标签S和标签C的查询q在Q中的所有项目。 我们使用以下方法构造了三个数据集：

(1)关键词搜索：

$$D_{KE} = \{(q, i)|q \in Q, i \in I_E(q)\}$$

(2) 按示例查询数据集：

$$D_{KS} = \{(i_1, i_2)|q \in Q, i_1 \in I_E(q), i_2 \in I_S(q)\}$$

(3) 互补推荐商品数据集：

$$D_{CC} = \{(i)|q \in Q, i \in I_C(q)\}$$

Evaluation Protocols

我们使用留一出数据分割策略，对Lowe's数据集进行了处理。在这个过程中，我们将最近的交互作为测试项，次近的交互作为验证项，其余的交互用于训练。这样，在每个测试、验证集中，都会有超过890,000个交互。这种方法可以让我们评估模型在未来可能面临的实际应用情况下的表现。然而，Amazon ESCI数据集中没有提供用户标识符或时间戳，所以我们随机选择了80%的请求作为训练项，10%作为验证项，剩下的10%作为测试项。 此外，我们在评估模型时，还报告了许多不同的性能指标。

Implementation Details

我们使用了HuggingFace的BERT-base模型作为预训练语言模型。对于Lowe's数据集，我们加载了从huggingface.co/bert-bas...检查点获得的预训练权重。对于Amazon ESCI数据集，我们获取了从huggingface.co/sentence...检查点获取的预训练权重。 我们在验证集上以NDCG衡量性能，然后选择训练迭代次数为$[8, 12, 16, 24, 48]$。我们设置了用户的交互历史记录数量为5，批量大小为384。预训练和个性化微调的学习率为$7e^{-6}$和$7e^{-5}$，分别。对于个性化微调任务，我们仅保留有至少10个交互的用户和至少5个交互的互补商品推荐任务的用户。隐藏维度为768，头部数为12，每个头中的关键值和值的维度为$l = l_v = 64$。用户嵌入维度为$l_u = 128$，功能嵌入维度为$l_f = 64$。我们使用Adam作为优化器。

Baselines

实验使用的基准线包括术语匹配模型、密集检索模型以及协同推荐和序列推荐等。如果适用，还可以考虑其他相关基准线。[1]引文格式：你需要将你的引文按照APA或其他适当的引用风格来呈现。

- NCF: 这是一种有效的协同过滤模型，它结合了广义矩阵分解和多层感知器的方法来进行推荐。它只从项项交互中学习，不能应用于关键词搜索任务。
- DPR 是一个密集检索模型，它通过在 BM25 中检索的项目中采样负文档，以及在批处理中进行负采样来实现。此外，DPR 只使用最后请求的信息，并且不进行个性化操作。
- 情境感知DPR模型：将DPR模型扩展到考虑用户历史。使用$< code > SEP < /code >$将当前用户请求与过去交互分开，然后输入查询编码器。
- ANCE: ANCE是一种有效且稠密的检索模型，它使用自身的模型来挖掘硬负样本。与DPR类似，ANCE无法实现个性化，因此我们还包含了一种名为**上下文感知ANCE**的模型，其方法与用于上下文感知DPR的方法相似。
- 火箭问答：是基于大规模数据集和去噪负样本进行对比学习的先进检索模型。该模型与DPR、ANCE类似，不具有个性化特性。为解决这一问题，我们引入了上下文感知的火箭问答技术。
- 本研究提出了一种名为BERT4Rec++的序列推荐模型，该模型使用BERT来表示用户的交互历史，并预测接下来的物品。BERT4Rec原始模型通过接收物品ID并预测序列中的下一个物品ID来实现此功能。为了改进BERT4Rec模型，我们通过BERT对物品内容进行编码，将这种编码称为BERT4Rec++.
- 本文介绍SASRec++模型，一种基于自注意力机制的序列推荐模型。它能够识别与用户历史交互中相关的项目，并用于未来商品预测。然而，该模型不适用于关键词搜索任务。我们将使用BERT进行内容嵌入，并将此模型命名为SASRec++。
- JSR: 这是一个联合学习搜索和推荐任务的神经框架。每个任务在其基底共享网络之上都有一个特定的任务层。
- JSR+BERT4Rec++: 使用BERT4Rec++改进JSR，将用户上下文编码。
- SRJGraph： 这是一个基于神经图卷积的联合搜索和推荐任务的新框架。

使用公共代码实现BM25、NCF、DPR、ANCE、RocketQA、SASRec++和BERT4Rec。实现JSR和SRJGraph未公开模型。将模型训练分为两类：任务特定训练和联合训练。任务特定训练只在目标任务上训练，联合训练可以访问所有任务的数据。使用BERT作为基础架构和相同的预训练权重训练所有密集检索基准。所有上下文感知变体消耗历史数据反向时间顺序。SASRec++和BERT4Rec++添加多层Transformer，每层包括12个头部，总隐藏维度为64。JSR和SRJGraph任务特定层为一个密集层，隐藏维度为768，由相同的BERT模型初始化。选择学习率从$1e-4, 7e-5, 1e-5, 7e-6$中，根据开发集结果确定。

Experimental Results

Main Results

Lowe's数据集上，三种信息访问功能（包括BM25、NCF和深度学习模型）的性能均被报告。结果显示，深度学习模型在性能上优于BM25和NCF。值得注意的是，NCF不适用于关键词搜索任务，因为它是基于协同过滤的方法。同时，作者观察到，对稠密检索模型进行上下文感知变异显著优于原始DPR和ANCE模型。这显示了电子商务中信息访问的重要性。

	Keyword Search			Query by Example			Complementary Item Rec.		
Model	MRR	NDCG	Recall	MRR	NDCG	Recall	MRR	NDCG	Recall
BM25	0.089	0.095	0.367	0.153	0.167	0.584	0.016	0.014	0.111
Task-Specific Training									
NCF	-	-	-	0.132	0.147	0.351	0.117	0.118	0.236
DPR	0.188	0.192	0.578	0.171	0.180	0.598	0.153	0.156	0.487
ANCE	0.193	0.199	0.582	0.176	0.188	0.601	0.159	0.158	0.494
RocketQA	0.201	0.207	0.595	0.189	0.204	0.613	0.174	0.176	0.507
Context-Aware DPR	0.324	0.377	0.848	0.311	0.356	0.860	0.278	0.283	0.707
Context-Aware ANCE	0.332	0.385	0.856	0.317	0.361	0.866	0.289	0.292	0.714
Context-Aware RocketQA	0.335	0.389	0.861	0.326	0.369	0.874	0.300	0.304	0.723
SASRec++	-	-	-	0.305	0.347	0.836	0.271	0.264	0.695
BERT4Rec++	-	-	-	0.314	0.354	0.851	0.283	0.279	0.703
Joint Training									
JSR	0.324	0.379	0.853	0.349	0.380	0.878	0.325	0.317	0.760
JSR+BERT4Rec++	0.337	0.394	0.871	0.415	0.479	0.919	0.421	0.419	0.820
SRJGraph	0.336	0.392	0.874	0.416	0.478	0.921	0.423	0.420	0.822
UIA	**0.340**$^{\triangle}$	**0.399**$^{\blacktriangle}$	**0.880**$^{\blacktriangle}$	**0.433**$^{\blacktriangle}$	**0.495**$^{\blacktriangle}$	**0.945**$^{\blacktriangle}$	**0.438**$^{\blacktriangle}$	**0.432**$^{\blacktriangle}$	**0.836**$^{\blacktriangle}$

此外，BERT4Rec++相较于SASRec++有更好的效果，表明堆叠多个Transformer层的BERT能够更好地捕捉用户历史行为。在训练于相应单个任务的模型中，Context-Aware ANCE达到最佳性能。表显示，联合训练模型比任务特定模型在查询示例和互补项推荐等任务上的表现更好，而关键词搜索任务则是例外。 总体而言，。除了关键词搜索任务中的MRR值，几乎所有的改进都有统计学意义。

常较小，因为数据集缺乏对。我们使用已发布的预训练模型进行了零度评估，并在Amazon ESCI数据集上进行了测试，但结果低于BM25。例如，在互补推荐任务中，MRR\@10为0.223，而BM25为0.230。我们推测这是因为该方法在优先考虑语义文本相似性且忽视互补或替代项目匹配信号的数据集上进行训练。

Model	Keyword Search			Query by Example			Complementary Item Rec.		
	MRR	NDCG	Recall	MRR	NDCG	Recall@50	MRR	NDCG	Recall
BM25	0.513	0.351	0.494	0.017	0.011	0.084	0.030	0.032	0.165
Task-Specific Training									
DPR	0.505	0.347	0.511	0.235	0.174	0.527	0.434	0.450	0.838
ANCE	0.522	0.354	0.519	0.237	0.178	0.531	0.431	0.443	0.825
RocketQA	0.526	0.357	0.525	0.244	0.185	0.538	0.445	0.458	0.847
Joint Training									
JSR	0.528	0.355	0.527	0.243	0.192	0.536	0.477	0.484	0.853
SRJGraph	0.526	0.351	0.522	0.241	0.187	0.540	0.477	0.488	0.855
UIA	**0.532**▲	**0.360**▲	**0.533**▲	**0.251**▲	**0.199**▲	**0.543**▲	**0.490**▲	**0.493**▲	**0.868**▲

Conclusions

本文提出了一种名为，用于实现普遍信息访问系统(UIAS)。该框架将信息访问功能编码到用户的请求中，通过近邻搜索提高检索和推荐效率。此外，还引入了一个注意力个人化网络，使得。 通过使用两个不同来源的大规模真实世界数据集和Amazon ESCI数据集，我们评估了。为了进一步验证模型的各方面性能，我们进行了大量的实验，包括删除模型的研究。 根据本文的结论，我们相信通用信息访问在未来充满希望，不仅短期内有可能实现，长远来看也将成为可能。

原文《A Personalized Dense Retrieval Framework for Unified Information Access》

发布于 2024-02-05 13:57 · IP 属地北京

信息检索 搜索 Transformer