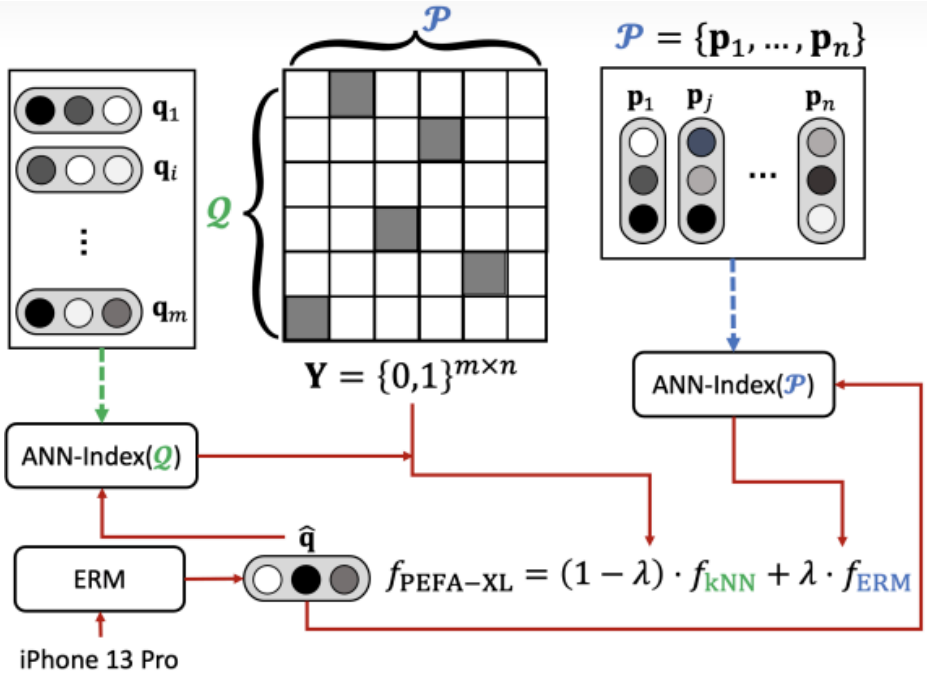
知乎



亚马逊2023创新技术：PEFA框架——向量召回中针对GPT模型的高效快速黑盒微调策略

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

原文《PEFA: Parameter-Free Adapters for Large-scale Embedding-based Retrieval Models》

Introduction

在搜索引擎+中，使用大规模文本库检索相关文档。通常使用双向编码器等基于向量嵌入的模型，这些模型将Query和文档映射到一个语义嵌入空间，使相关的Query和文档彼此接近。推理时，通过找到与Query最相似的文档，搜索过程转化为最大内积检索问题。借助适当的索引数据结构，如Faiss、ScaNN或HNSWLIB，可以高效地解决MIPS问题，其时间复杂度+为线性。

为了适应下游检索任务，通常使用全参数微调+的方法，该方法涉及计算梯度并更新Transformer编码器的参数。然而，在工业环境中实现全参数微调面临着挑战，因为计算量巨大。例如，在现代电子商务商店中，与Query相关的(Query，产品)对可能有数十亿或更多。对于这样的大规模全参数微调，BERT模型可能需要数千个GPU小时的时间，因为它涉及到多个步骤：预训练、第1阶段的随机负样本和BM25候选人的微调、第2阶段的硬挖掘的负样本的微调以及第3阶段的从昂贵的跨注意力模型中提取的知识的微调。 此外，这些微调方法还需要访问模型的梯度信息，而许多黑盒语言模型，如GPT-3+和更远，无法访问这些信息。

针对这一问题，本文提出了一种名为PEFA的框架，即参数自由的适配器，用于快速微调黑盒BERT模型，无需访问任何梯度信息。PEFA框架中的得分函数是BERT模型和新的非参数化+的最近邻模型（KNN）之间的凸组合。通过学习新KNN模型，可以减少为构建ann索引所需的过程，该索引存储Query向量及其学习信号的键值对。在推理时间，KNN模型将在邻域中查找与训练Query相似的Query，并将相关的文档作为其评分函数。在PEFA框架下，我们引入了两种KNN模型：一种是本地KNN，另一种是全局KNN。这两种模型分别用于不同的场景。

对于文档检索任务，PEFA可以提高TriviaQA中预训练的ERMs在Recall\@100上的性能13.2%，以及提高NQ中微调的ERMs在Recall\@100上的性能5.5%。 对于NQ数据集，将PEFA应用于微调的GTR模型可以达到新的最先进水平（SOTA），在相同模型大小的情况下，Recall\@10为**88.71%**超过了之前的SOTA基于Seq2Seq的NCI中的Recall\@10的**85.20%**。 对于包含十亿级数据的产品搜索任务，PEFA可以提高微调的ERMs在Recall\@100上的性能5.3%和14.5%。

知乎

Dense Text Retrieval

密集文本检索通常采用Embedding-based Retrieval Model（ERM）架构，也称为双编码器。为了让文章更简洁，我们将passage和document互换使用。 给定Query$q \in \mathcal{X}$和一个段落$p \in \mathcal{X}$ ， ERM的关联得分函数$f_{\mathrm{ERM}}(q,p)$是

$$f_{\mathrm{ERM}}(q,p) = E_{p\sim q}(f(p)) = E_p[\log P(q|p)]$$

$$f_{\mathrm{ERM}\backslash\mathrm{xspace}}(q,p;\theta) = \langle E(q;\theta), E(p;\theta)\rangle$$

定义$\mathcal{D}$为一个样本集，其中每个元素是一个Queryq_i和对应的段落p_i。 编码器参数θ通常通过最大化似然损失函数学习，即最大化

$$\max_\theta \sum_{(q,p)\in\mathcal{D}} \log p_\theta(p|q)$$

这个最大化的任务通常通过反向传播算法来实现，其中 Softmax函数被用来计算条件概率$p_\theta(p|q)$。

$$p_\theta(p|q) = \frac{\exp\left(f_{\mathrm{ERM}\backslash\mathrm{xspace}}(q,p;\theta)\right)}{\sum_{p'\in\mathcal{D}}\exp\left(f_{\mathrm{ERM}\backslash\mathrm{xspace}}(q,p';\theta)\right)}$$

在实际应用中，为了减少计算复杂度，可以使用负采样技术来近似昂贵的条件Softmax部分函数。

Problem Statement

我们提出了一种名为PEFA的参数自由适配器，用于使用ERM。它通过采用非参数化的KNN组件来实现。这种模型的学习是无约束的，避免了对ERM进行任何优化步骤以调整参数。PEFA主要在推理阶段构建ANN索引，这使得它可用于预训练和微调的ERM，甚至是那些初始化为黑盒LLMs的ERM。

需要注意的是，虽然PEFA与其他旨在在学习阶段获得更好预先训练或微调的ERM的研究文献相关，但它们之间互不依赖且具有互补性，包括最近关于ERM参数效率微调的研究。最后，尽管我们假设从ERM获取的嵌入是单位范数（即ℓ_2归一化），但我们提出的PEFA技术可以轻松地扩展到非单位范数的情况，只需更改用于KNN的距离度量即可。

Proposed Framework

$$f_{\mathrm{PEFA}\backslash\mathrm{xspace}}(\hat{q},p_j) = \lambda\cdot f_{\mathrm{ERM}\backslash\mathrm{xspace}}(\hat{q},p_j) + (1-\lambda)\cdot f_{\mathrm{kNN}\backslash\mathrm{xspace}}(\hat{q},p_j)$$

$\lambda \in [0,1]$是插值超参数，用于平衡ERM和KNN模型的重要性。PEFA是一种学习自由的框架，这意味着θ不会被改变，而公式只在推理时应用。

$$f_{\mathrm{kNN}\backslash\mathrm{xspace}}(\hat{q},p_j) = \langle \hat{q}, \mathbf{Q}^\top \mathbf{D}(\hat{q},\mathbf{Q})\mathbf{Y}_{:,j}\rangle$$

其中

$$\mathbf{D}(\hat{q},\mathbf{Q}) \in \mathbb{R}^{n\times n}$$

的归一化对角矩阵，它像一个门机制，控制当前测试Query$\hat{q}$应该关注的训练Query集。在给定这个矩阵的情况下，我们可以将其代入公式 中，得到PEFA的具体评分函数。

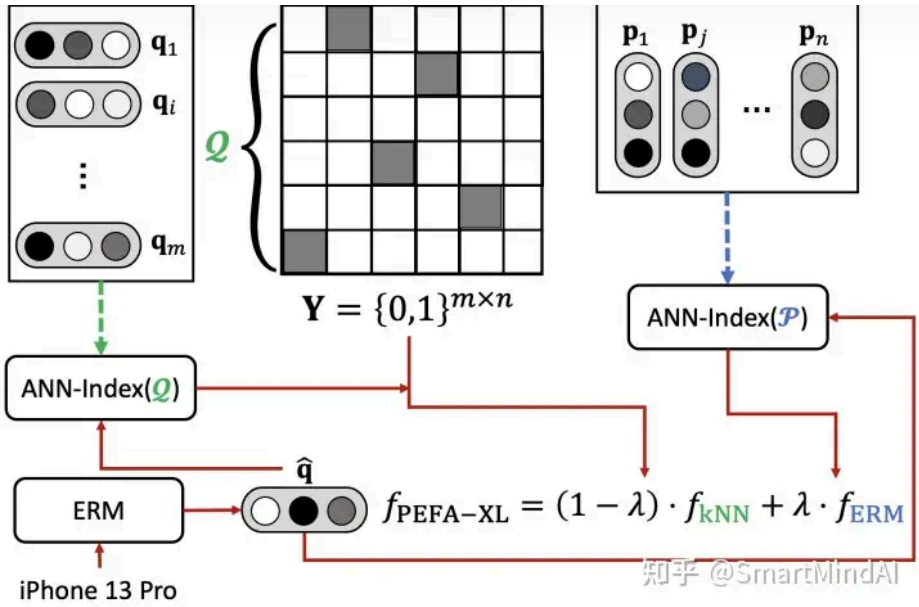
$$f_{\mathrm{PEFA}\backslash\mathrm{xspace}}(\hat{q},p_j) = \lambda\langle \hat{q},p_j\rangle + (1-\lambda)\langle \hat{q}, \mathbf{Q}^\top \mathbf{D}(\hat{q},\mathbf{Q})\mathbf{Y}_{:,j}\rangle$$

其中，基于设计的对角矩阵 $\mathbf{D}(\hat{q},\mathbf{Q})$

这种实现方式有两种，分别是PEFA-XL（第3节）和PEFA-XS（第4节）。

PEFA-XL

知乎



KNN模型的一个标准实现是，测试查询$\hat{\boldsymbol{q}}$只关注$\mathcal{Q}$中最相似的前k个训练查询。 具体来说，如果某个训练Query$i$在

$$\mathrm{NN}(\hat{\boldsymbol{q}}, \mathbf{Q}; k)$$

中，则其在对角矩阵 $\mathbf{D}(\hat{\boldsymbol{q}}, \mathbf{Q})$

中的元素为1，否则为0。因此，可以由上述定义得到式 中的KNN模型。

$$f_{\text{kNN}\backslash\text{xspace}}(\hat{\boldsymbol{q}}, \boldsymbol{p}_j) = \left\langle \hat{\boldsymbol{q}}, \sum_{i=1}^{n} (\mathbf{D}_{i,i} \mathbf{Y}_{i,j}) \cdot \boldsymbol{q}_i \right\rangle = \sum_{i \in \mathrm{NN}(\hat{\boldsymbol{q}}, \mathbf{Q}; k)} \left\langle \hat{\boldsymbol{q}}, \boldsymbol{q}_i \right\rangle \cdot \mathbf{Y}_{i,j}$$

通过将式 中带有查询感知的KNN模型插值[+]到式 中，

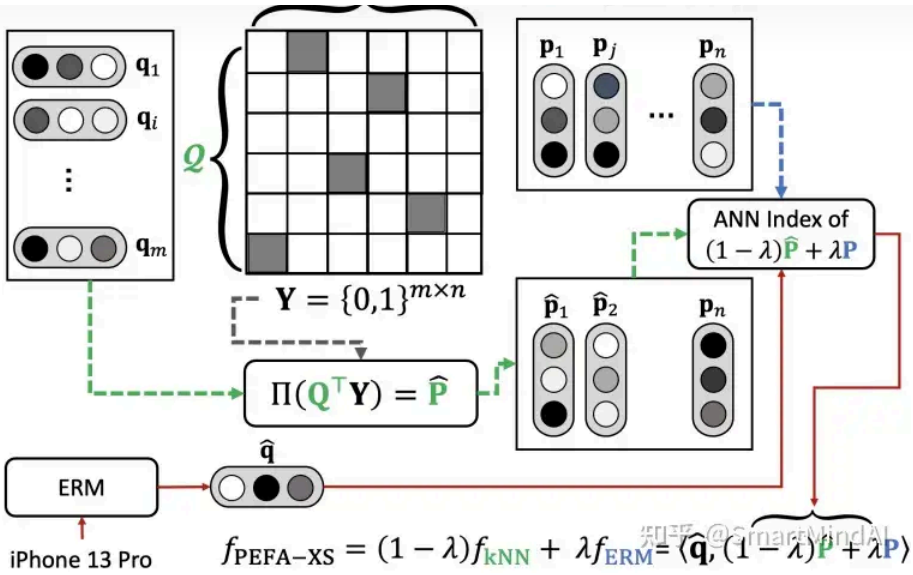
$$f_{\text{PEFA-XL}\backslash\text{xspace}}(\hat{\boldsymbol{q}}, \boldsymbol{p}_j) = \lambda \left\langle \hat{\boldsymbol{q}}, \boldsymbol{p}_j \right\rangle + (1-\lambda) \sum_{i \in \mathrm{NN}(\hat{\boldsymbol{q}}, \mathbf{Q}; k)} \left\langle \hat{\boldsymbol{q}}, \boldsymbol{q}_i \right\rangle \cdot \mathbf{Y}_{i,j}$$

在这个过程中，，KNN模型通过聚合训练Query的相关段落来生成匹配集。需要注意的是，式 中$f_{\text{ERM}\backslash\text{xspace}}$的评分函数在内积两个单位形式嵌入时被限制在$[-1, 1]$范围内，这是因为余弦相似性[+]的内积范围是有限的。另一方面，式 中$f_{\text{kNN}\backslash\text{xspace}}$的评分函数需要额外的归一化以便其分数能够调整为$f_{\text{ERM}\backslash\text{xspace}}$。

为了达到这个目的，建议将$f_{\text{kNN}\backslash\text{xspace}}$归一化为$k$，即将某些点的权值设置为$k/1$。这样可以使$f_{\text{kNN}\backslash\text{xspace}}$的值调整为与$f_{\text{ERM}\backslash\text{xspace}}$相同，但不会超出$[0, 1]$的范围。

PEFA-XS

知乎



PEFAxl通过近似地使用目标段落 $\boldsymbol{p_j} \in \mathcal{P}$

的相应Query集 $\mathcal{I}(\boldsymbol{p_j}, \mathbf{Y})$

来代替 $\mathbf{NN}(\hat{\boldsymbol{q}}, \mathcal{Q}; \boldsymbol{k})$

结果的对角矩阵$\mathbf{D}_{i,i} = 1$如果 $i \in \mathcal{I}(\boldsymbol{p_j}, \mathbf{Y})$

其他$\mathbf{D}_{i,i} = 0$。

$$\boldsymbol{f}_{\mathbf{kNN}\backslash\mathbf{xspace}}(\hat{\boldsymbol{q}}, \boldsymbol{p_j}) = \langle\hat{\boldsymbol{q}}, \textstyle\sum_{i\in\mathcal{I}(\boldsymbol{p_j},\mathbf{Y})} \mathbf{Y}_{i,j}\cdot\boldsymbol{q_i}\rangle = \langle\hat{\boldsymbol{q}}, \mathbf{Q}^\top\mathbf{Y}_{:,j}\rangle$$

通过近似地使用Query独立的kNN模型插入到PEFAxl评分函数中，提出了PEFAxs的评价函数。

$$\boldsymbol{f}_{\mathbf{PEFA\text{-}XS}\backslash\mathbf{xspace}}(\hat{\boldsymbol{q}}, \boldsymbol{p_j}) = \lambda\langle\hat{\boldsymbol{q}}, \boldsymbol{p_j}\rangle + (1-\lambda)\langle\hat{\boldsymbol{q}}, \mathbf{Q}^\top\mathbf{Y}_{:,j}\rangle$$

在实现时，图形显示了PEFAxs的方法。与PEFAxl的实现相似，需要对评分函数$\boldsymbol{f}_{\mathbf{kNN}\backslash\mathbf{xspace}}$进行归一化，使其得分为1。为此，引入了$\boldsymbol{\ell_2}$归一化操作器

$$\boldsymbol{\Pi}(\boldsymbol{x}) = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$$

它将嵌入映射[+]回单位球上。然后，可以将PEFAxs的评分函数重写为

$$\tilde{\boldsymbol{f}}_{\mathbf{kNN}\backslash\mathbf{xspace}} = \boldsymbol{\Pi}(\boldsymbol{f}_{\mathbf{kNN}\backslash\mathbf{xspace}} + \tfrac{1}{n}\textstyle\sum_i^n(\boldsymbol{\Pi}^{-1}\mathbf{e}_i)^T\mathbf{x})$$

$$\boldsymbol{f}_{\mathbf{PEFA\text{-}XS}\backslash\mathbf{xspace}}(\hat{\boldsymbol{q}}, \boldsymbol{p_j}) = \langle\hat{\boldsymbol{q}}, \\ \boldsymbol{\lambda}\cdot\boldsymbol{p_j} + (1-\boldsymbol{\lambda})\cdot\boldsymbol{\Pi}(\mathbf{Q}^\top\mathbf{Y}_{:,j})\rangle$$

Experiments on Document Retrieval

Datasets & Evaluation Protocols

本节将讨论两个用于文档检索的公共基准数据集，分别是Natural Questions和TriviaQA。

- Natural Questions：这是开放领域问答数据集，有320k个Query-文档对，其中包含答案的文档是从维基百科收集的，而Query是用自然语言提出的。我们使用的是通常称为NQ 的数据集。
- TriviaQA：这是一个阅读理解数据集，有来自维基百科[+]域的78k个Query-文档对。我们使用与相同版本的数据集。

是通过召回率*来衡量性能，这是在检索社区中广泛使用的。具体来说，对于一个预测得分向量 $\hat{\boldsymbol{y}} \in \mathbb{R}^n$

和一个真实的标签向量$\boldsymbol{y} \in \{0,1\}^n$，Recall\@k定义为

$$\textbf{Recall@}k = \frac{1}{|\boldsymbol{y}|} \sum_{j \in \text{top}_k(\hat{\boldsymbol{y}})} \boldsymbol{y}_j$$

其中 $\text{top}_k(\hat{\boldsymbol{y}})$

表示具有第k大预测得分的标签。

Implementation Details

我们的PEFA框架可以应用于任何黑箱无监督学习方法。我们将PEFA应用到各种竞争性无监督学习模型，如超大规模预训练语言模型BERT、大规模自注意力机制*模型DPR、微调网络模型MPNet、搜索五点模型STfive以及不同寻常的搜索（Graph Transformer）模型GTR。此外，本段话还讨论了超参数设置。PEFA有两个超参数，即插值系数λ和近邻数k。我们在第4部分进行了超参数选择的分析，在

$$\lambda = \{0.1, 0.3, 0.5, 0.7, 0.9\}$$

和$k = \{16, 32, 64\}$的情况下进行了比较。当$\lambda = 1.0$时，PEFA退化回其基础无监督学习模型。对于无监督索引搜索，我们考虑HNSW作为搜索算法*，并根据现有工作进行超参数设置。在索引构建阶段，最大边每节点的数量为$M = 32$，优先队列*大小为构造图的最大优先级为$efC = 500$。在线服务阶段，图搜索的beam宽度为$efS = 300$。

Main Results

Methods	Recall@10	Recall@100
BM-25	32.48	50.54
DSI (base) [53]	56.60	-
NCI (base) [56]	85.20	92.42
SEAL (large) [2]	81.24	90.93
$\text{Sent-BERT}_{\text{distill}}$ [45]	67.08	81.40
+PEFA-XS (ours)	80.52	92.22
+PEFA-XL (ours)	85.26	92.53
DPR_{base} [29]	70.68	85.19
+PEFA-XS (ours)	83.45	92.22
+PEFA-XL (ours)	84.65	92.07
$\text{MPNet}_{\text{base}}$ [50]	80.82	92.39
+PEFA-XS (ours)	86.67	<u>94.53</u>
+PEFA-XL (ours)	**88.72**	**95.13**
$\text{Sentence-T5}_{\text{base}}$ [41]	73.63	88.16
+PEFA-XS (ours)	82.52	92.18
+PEFA-XL (ours)	83.69	92.55
GTR_{base} [42]	79.74	90.91
+PEFA-XS (ours)	84.90	93.28
+PEFA-XL (ours)	<u>88.71</u>	94.36
Avg. Gain of PEFA-XS over ERM	+9.22	+5.28
Avg. Gain of PEFA-XL over ERM	+11.32	+5.72

、$\mathbf{DPR}_{\mathbf{base}}$、$\mathbf{MPNet}_{\mathbf{base}}$、$\mathbf{GTR}_{\mathbf{base}}$）进行微调，并在。。10和召回\@100方面的平均增益达9.22%和5.29%，而%和5.20%。对于竞争性强的ERMs，如$\mathbf{GTR}_{\mathbf{base}}$，2Seq方法------NCI。当添加，使用10和召回\@100分别为88.72%和94.13%，远高于之前最先进的方法------NCI。

在表中，对比使用-文档对进行微调的情况。通过这种方法，我们评估了。结果显示，在召回\@20的情况下，，其中$\mathbf{18.67\%}$的平均增益，而$\mathbf{17.07\%}$的平均增益。需要注意的是，这两种模型都没有针对下游任务进行微调。

Ablation Studies

ERM	PEFA	Recall@100 of various λ				
		0.1	0.3	0.5	0.7	0.9
$\mathrm{DPR}_{\mathrm{base}}$	PEFA-XS	91.48	92.22	91.71	89.87	87.08
	PEFA-XL (k=16)	91.98	90.66	89.72	88.54	87.62
	PEFA-XL (k=32)	92.07	90.50	89.20	88.62	87.46
	PEFA-XL (k=64)	91.93	89.89	88.95	88.39	87.04
$\mathrm{Sentence\text{-}T5}_{\mathrm{base}}$	PEFA-XS	91.23	92.16	92.20	91.61	89.72
	PEFA-XL (k=16)	92.53	91.25	90.82	90.69	90.24
	PEFA-XL (k=32)	92.34	91.20	90.96	90.77	90.11
	PEFA-XL (k=64)	92.22	91.26	91.03	90.70	89.90
$\mathrm{GTR}_{\mathrm{base}}$	PEFA-XS	92.11	93.07	93.31	92.85	91.74
	PEFA-XL (k=16)	94.36	93.32	92.81	92.53	91.93
	PEFA-XL (k=32)	94.32	93.23	92.82	92.44	91.79
	PEFA-XL (k=64)	93.93	93.14	92.76	92.29	91.62

在表中，对：$\boldsymbol{\lambda}$是平衡$\boldsymbol{f}_{\backslash \mathbf{erm}}$和$\boldsymbol{f}_{\backslash \mathbf{knn}}$的系数$\boldsymbol{k}$是用于控制$\boldsymbol{f}_{\backslash \mathbf{knn}}$中的最近邻数量。研究发现，当$\mathbf{0.0} < \boldsymbol{\lambda} < \mathbf{1.0}$时，无论$\mathbf{1.0}$为，100始终较高。对于PEFA-XL，当$\boldsymbol{\lambda} = \mathbf{0.5}$和$\boldsymbol{\lambda} = \mathbf{0.1}$时，平均增益大多达到最大值。此外，线性插值*，因此不会增加任何推理延迟开销相对于。而对于PEFA-XL，除了超参数*$\boldsymbol{\lambda}$之外，还有一个超参数$\boldsymbol{k}$，当$\boldsymbol{k} = \mathbf{32}$时，性能通常达到饱和。

Experiments on Product Search

我们在大规模产品搜索系统上做了实验，证明了，不仅能提供多种预训练的，还能提供全参数微调的。

Datasets & Evaluation Protocols

根据目录大小$\boldsymbol{n} = |\mathcal{P}$，我们构造三个子集。

- ProdSearch-5M：大约包含3千万条相关Query-产品对，涵盖了约1千万个Query和5百万个产品。
- ProdSearch-15M：大约包含1.5亿条相关Query-产品对，涵盖了约4千万个Query和15百万个产品。
- ProdSearch-30M：大约包含5亿条相关Query-产品对，涵盖了约1亿个Query和30百万个产品。

对于所有的ProdSearch数据集，数据统计并不反映实际的电子商务系统的流量，因为涉及隐私问题。所有相关的Query-产品对是匿名汇总搜索日志中的随机样例。我们进一步将这些对分为训练集和测试集*，通过时间范围进行划分，在这里，我们使用前十二个*月的搜索日志作为训练集，最后一个月的搜索日志作为评估测试集。

为了消除对我们，所有的测试Query在训练集中都是未见的。为了避免透露生产系统的确切性能，我们报告了在基线。我们还报告了离线索引阶段的ANN索引大小（GiB）和索引构建时间（小时）。在线推理阶段，按照ANN基准协议，我们考虑单线程设置，并报告了Query延迟（毫秒/Query）。

知乎

Methods	ProdSearch-5M		ProdSearch-15M		ProdSearch-30M	
	Recall@100	Recall@1000	Recall@100	Recall@1000	Recall@100	Recall@1000
$MPNet_{base}$ [50]	0.00	0.00	0.00	0.00	0.00	0.00
+PEFA-XS (ours)	11.23	13.14	5.05	11.79	9.67	17.47
+PEFA-XL (ours)	22.83	12.31	23.48	21.56	27.22	18.96
$Sentence\text{-}T5_{base}$ [41]	0.44	3.42	1.32	3.44	1.89	5.17
+PEFA-XS (ours)	13.63	17.13	10.39	16.34	13.18	21.28
+PEFA-XL (ours)	23.09	13.43	23.91	23.72	30.10	21.25
GTR_{base} [42]	7.85	9.23	6.75	10.33	8.35	9.83
+PEFA-XS (ours)	17.32	19.55	16.83	25.00	18.49	24.38
+PEFA-XL (ours)	27.79	19.23	27.87	28.75	31.71	24.28
$E5_{base}$ [54]	9.93	9.75	9.98	12.98	12.01	12.61
+PEFA-XS (ours)	19.23	19.18	17.21	27.78	20.11	26.08
+PEFA-XL (ours)	26.83	17.75	30.48	31.07	31.91	25.49
FT-ERM [40]	21.32	20.87	21.74	30.04	18.49	24.11
+PEFA-XS (ours)	23.42	22.17	26.34	34.84	23.79	29.61
+PEFA-XL (ours)	**29.32**	**22.87**	**36.54**	**37.24**	**32.99**	**30.01**
Avg. Gain of PEFA-XS	16.97	18.23	15.16	23.15	17.05	25.76
Avg. Gain of PEFA-XL	25.97	17.12	28.46	28.47	30.79	24.00

在表中，我们将ERMs（例如$MPNet_{base}$、

$Sentence\text{-}T5_{base}$

、$E5_{base}$）和微调后的ERMs（FT-ERM）。由于对专有产品搜索数据集的隐私考虑，我们只报告了与其他基线（即$MPNet_{base}$）相比绝对召回率的提高。 如果没有PEFA，预训练的FT-ERM，后者经过精心预训练和微调。添加，使其与甚至优于微调后的。以最大的数据集。向

$Sentence\text{-}T5_{base}$

、100的30.10%、31.71%和31.91%。这些召回率\@100的提升已经超过了微调后的FT-ERM。另一方面，。只有$E5_{base}$+

100比微调后的。类似于，我们也发现。例如，在最大的数据集，100分别提高了

5.3\%和14.50\%

发布于 2024-03-26 12:37 · IP 属地北京

GPT 亚马逊 (Amazon.com) 检索