# CoT-MAE v2：用于段落检索的多视图建模上下文掩码自编码器

**吴星**[1,2,3]**，马光远**[1, 2 *]**，王鹏**[1,2]**林萌**[1,2]**，林子佳**[3]**，张福正**[3] 和 **胡松林**[1, 2†][1] 信息工程研究所，中国科学院[2] 网络空间安全学院，中国科学院大学[3] 快手科技 {wuxing, maguangyuan, wangpeng, linmeng,husonglin}@iie.ac.cn linzijia07@tsinghua.org.cn, zhangfuzheng@kuaishou.com

arXiv:2304.03158v1 [cs.CL] 5 Apr 2023

## 摘要

不断涌现的技术旨在提升段落检索的性能。作为一种有效的表示瓶颈预训练技术，上下文掩码自编码器利用上下文嵌入来辅助段落重建。然而，它仅使用单一的自编码预任务进行稠密表示预训练。本研究将多视图建模引入上下文掩码自编码器。首先，多视图表示同时利用稠密向量和稀疏向量作为多视图表示，旨在从不同方面捕捉句子语义。此外，多视图解码范式在表示瓶颈预训练中同时利用自编码和自回归解码器，旨在提供重建信号和生成信号，从而实现更好的上下文表示预训练。我们将这种多视图预训练方法称为 CoT-MAEv2。通过广泛的实验，我们表明，CoT-MAEv2 在大规模段落检索基准和域外零样本基准上均表现出色且具有稳健性。

## 1 引言

段落检索是在大型语料库中搜索与特定查询相关的段落，对于网络搜索（Fan 等人，2021；Guo 等人，2022年；Lin 等人，2021a）、问答（Karpukhin 等人，2020；Lee 等人，2020；Zhu 等人，2021年）和对话系统（Gao 等人，2022年；Yu 等人，2021年）等多种应用都至关重要。BM25 等稀疏检索方法是实用且占主导地位的方案。然而，基于预训练语言模型（PLM）（Devlin 等人，2018；Liu 等人，2019）的检索方法近年来日益流行。预训练语言模型被用于改进：（1）依赖词汇匹配的稀疏检索模型；（2）对查询与段落在潜在语义空间中的语义交互进行建模的稠密检索模型。

为了改进稀疏检索，预训练语言模型通过将查询和段落中的每个词项投射到词汇表大小的权重向量，缓解词汇不匹配问题。权重向量中的每个维度代表预训练语言模型词汇表中一个词项的权重，而查询与段落之间的相关性取决于词汇匹配（Formal 等人，2021b，a；Shen 等人，2022年）。基于预训练语言模型的稀疏向量对段落在每个词汇词项上的特征进行建模（记为预训练语言模型的逻辑值输出），因此能够自然地捕捉词汇层面的语义。SPLADE（Formal 等人，2021b，a）使用可学习的预训练语言模型（例如 BERT）生成稀疏向量，并通过对比学习损失微调检索器。基于预训练语言模型的 SPLADE 优于 BM25 等传统稀疏检索方法。LexMAE（Shen 等人，2022年）进一步拓展了 SPLADE，提出预训练一种具有词汇瓶颈的掩码自编码器，以学习感知重要性的稀疏表示。它通过连续词袋表示瓶颈，以无监督的方式学习词汇重要性分布。

基于预训练语言模型的密集检索通常采用孪生或双编码器架构，将查询和文档映射到低维向量空间（Hofstätter 等人，2021年；Humeau 等人，2019；Xiong 等人，2020；Zhan 等人，2021年，2020）。查询与文档之间的相关性通过余弦相似度或点积计算。这种低维向量称为密集向量，经过训练后可捕捉段落级别的句子语义段落级别。近期研究尝试通过在预训练中加入辅助任务来提升稠密检索性能，例如上下文监督学习（Gao 和 Callan，2021b；Wu 等人，2022年）。上下文监督学习旨在通过考虑周围段落的上下文来改进文本表示。一种近期方法 CoT-MAE

*前两位作者贡献相同。
†通讯作者。

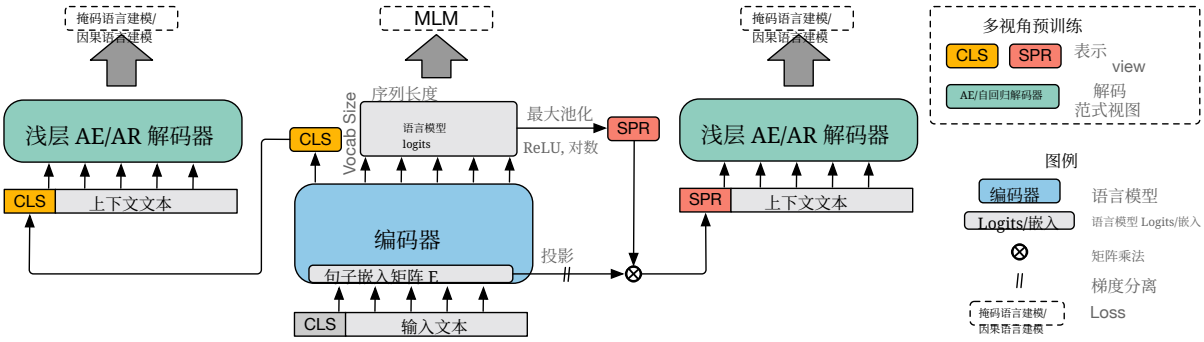



图 1：CoT-MAEv2 的预训练设计。CoT-MAE v2利用稠密（CLS）和稀疏（SPR）向量作为多视图表示。作为一种多视图解码范式，自编码解码器（AE-Dec）和自回归解码器（AR-Dec）被整合到上下文掩码自编码器预训练中，为表示预训练提供掩码语言建模重建信号和因果语言建模生成信号。

（Wu 等人，2022年）在此基础上进一步引入上下文掩码自编码，从而提升稠密检索性能。CoT-MAE 从文档中选取两个段落作为上下文段落，旨在借助上下文信息学习更好的表示。它使用一个浅层编码器（通常包含一两层），借助上下文段落嵌入重构掩码段落。

基于 PLM 的稀疏检索（Bai 等人，2020；Formal 等人，2021b，a；Shen 等人，2022年）和稠密检索（Wu 等人，2022年；Wang 等人，2022年；Gao 和 Callan，2021a）表明，稀疏表示和稠密表示都能有效捕捉文本含义。在专为稠密检索设计的上下文掩码自编码器（CoT-MAE）的基础上，一个自然的想法是将稀疏表示预训练纳入 CoT-MAE，即在上下文掩码自编码器预训练和微调过程中探索多视图表示。从理论上讲，稀疏表示侧重于词汇，而稠密表示侧重于句子语义，能够从不同角度呈现文本。这两种表示可能彼此兼容、相辅相成。

除了表示差异之外，先前方法中解码器的结构也存在区别。有些（Wu 等人，2022年；Shen 等人，2022年）选择使用自编码解码器执行预训练任务，而另一些（Lu 等人，2021年；Liu 和 Shao，2022年）更倾向于使用自回归解码器。尽管存在差异，这些解码器都能够利用编码器生成的文本表示进行预训练，这促使我们研究使用多个解码器进行预训练。这两不同类型的解码器具有不同的归纳偏置，对文本表示的使用方式也各不相同。我们相信，如果文本表示足够好，就应当能够适应使用方式的差异，从而使任何类型的解码器都能有效地完成预训练任务。

基于上述两个假设，本文扩展了 CoT-MAE，将多视图建模引入上下文掩码自编码器预训练。

- **多视图表示** 提出在预训练、微调和推理阶段使用稠密和稀疏表示，旨在从段落和词汇两个层面联合捕捉句子语义。通过进一步引入上下文表示瓶颈预训练，多视图表示能够实现强大的检索性能。

- **多视图解码范式** 提出为上下文表示预训练提供无偏的自监督信号。在上下文掩码自编码器中，自编码和自回归解码器均用于通过掩码语言建模（MLM）重建信号和因果语言建模（CLM）生成信号，辅助上下文嵌入的预训练。联合使用这两种解码信号有助于稳步提升检索性能。

融合这些视图的改进方法称为 **CoT-MAEv2**，如图1所示。为了验证我们提出的多视图上下文掩码自动编码器预训练的有效性，我们在大规模网络搜索基准上开展实验：MS-MARCO 段落排序（Nguyen 等人，2016年）、TREC

深度学习（DL）赛道 2019（Craswell 等，2020a）和2020 赛道（Craswell 等，2020b）。我们还在 BEIR（Thakur 等，2021年）基准测试中，使用了一组规模庞大的域外数据集进行评估。实验结果表明，CoT-MAE v2 相较于竞争基线取得了显著提升，并在这些基准数据集上达到了新的最先进性能。

我们的贡献总结如下：

• 我们提出将多视图预训练技术融入上下文掩码自编码器预训练。

• 多视图表示和多视图解码范式为表示学习提供了更出色的语义捕捉能力和丰富的自监督信号。

• 实验表明，在段落检索任务中，CoT-MAE v2 相较于具有竞争力的基线模型带来了显著的性能提升。

## 2 相关工作

基于预训练语言模型（预训练语言模型）（Devlin 等人，2018；Liu 等人，2019）的检索方法近年来日益受到关注。

### 2.1 预训练语言模型用于稠密检索

预训练语言模型在预训练阶段和微调阶段均能提升稠密检索效果。

预训练阶段预训练阶段的尝试分为两类。第一类聚焦于用于段落检索的段落预测任务（Chang等人，2020；Gao 和 Callan，2021b；Ma 等，2022年；Wu 等，2022年；Zhou等，2022年）。例如，（Chang 等人，2020）通过逆向完形任务（ICT）、正文优先选择（BFS）和维基链接预测（WLP）任务进行预训练。（Gao 和 Callan，2021b）基于文档邻近性，通过对比跨度预测进行预训练。（Ma 等，2022年）将对比跨度预测扩展到多个粒度。（Gao 和Callan，2021b）基于文档邻近性，通过对比跨度预测进行预训练。（Wu 等，2022年）采用结合混合相关跨度选择策略的生成式跨度预测进行预训练。另一类聚焦于使用辅助自编码任务增强编码器（Lu 等，2021；Gao和 Callan，2021a；Liu 和Shao，2022年；Wu 等，2022年）。例如，（Lu 等，2021；Gao 和Callan，2021a）旨在通过使用容量和注意力灵活性有限的弱解码器进行自编码，提升文本表示。（Liu 和 Shao，2022年；Wu 等人，2022年；Zhou 等人，2022年）通过对编码器和解码器采用非对称掩码比例来改进文本表示，然后借助其嵌入或上下文嵌入重建被大幅掩码的文本。最相关的方法是（Wu 等人，2022年），我们的工作基于（Wu 等人，2022年），并将多视角预训练技术融入（Wu 等人，2022年）。

微调阶段 在微调阶段，人们采用了多种方法来提升性能，包括挖掘困难负例（Xiong 等人，2020；Zhan 等人，2021）、晚期交互（Khattab 和 Zaharia，2020）、知识蒸馏（Lin 等人，2021b；Santhanam 等人，2021）、查询聚类（Hofstätter 等人，2021）、数据增强（Qu 等人，2020）以及检索器与重排序器联合优化（Ren 等人，2021；Zhang 等人，2022年，2021）。例如，（Xiong 等人，2020）通过 ANN 更新生成困难负例；（Zhan 等人，2021）利用微调后的检索器改进负例；（Khattab 和 Zaharia，2020）基于编码器隐藏状态，使用 MaxSim 对相似度建模；（Lin 等人，2021b）从 MaxSim 算子或强大的重排序器（Santhanam 等人，2021）中蒸馏知识；（Hofstätter 等人，2021）引入主题感知查询和边际采样以提升效率；（Qu 等人，2020）结合跨批次负例、去噪负例和数据增强；（Ren 等人，2021）通过列表式蒸馏，自适应地改进检索器和重排序器；（Zhang 等人，2022年）利用 HLATR 融合检索与重排序特征；（Zhang 等人，2021）通过极小极大对抗目标优化检索器与重排序器。与我们最相关的方法是（Xiong 等人，2020；Zhan 等人，2021；Santhanam 等人，2021），我们的微调流程正是基于这三种方法展开的。

### 2.2 用于稀疏检索的预训练语言模型

利用预训练语言模型改进稀疏检索的方法大致可分为两类。

**术语加权** 术语加权过程旨在根据基于上下文的词元表示（Dai and Callan，2020b，a； Gao et al.， 2021年； Linand Ma， 2021年）。例如，（Dai andCallan， 2020b）使用 BERT 词元表示来确定段落中术语的上下文特定重要性。（Dai and Callan，2020a）通过将文档拆分为段落并汇总术语权重来扩展这一方法。它使用标题，

网页链接和伪相关反馈，以生成弱监督信号来学习术语权重。（Gao 等，2021年）通过计算重叠词项的点积，得出查询与文本编码之间的相关性得分。（Lin 和 Ma，2021年）提出了一个统一这些方法的框架，将Gao 等，2021年）的权重向量降为一维，并能在提高效率的同时保持其有效性。

**词项扩展** 词项扩展通过使用预训练语言模型扩展查询或文档来改善词汇匹配（Nogueira 和 Lin，2019；Formal 等人，2021b，a； Shen 等人，2022年）。文本或查询的最终表示通过组合词元权重向量获得，从而有效地将其扩展为包含未出现的词项。应用稀疏性正则化，以获得用于高效使用倒排索引的稀疏表示。例如，（Nogueira 和 Lin， 2019）预测与文档相关的查询。（Formal 等人，2021b）和（Formal 等人，2021a）将查询和文本中的词项投影到权重向量中，该向量由掩码语言模型的逻辑值估计。（Shen 等人， 2022年）在（Formal 等人，2021a）的基础上进一步扩展，提出预训练词汇瓶颈掩码自编码器，以学习重要性感知稀疏表示。

## 3 方法

本节将介绍详细的预训练、微调和推理设计，包括同时采用稠密向量和稀疏向量的多视图表示，以及用于预训练的多视图解码范式。

### 3.1 多视图表示

基于 Transformer 的预训练语言模型（PLM），例如 BERT 编码器（记为 $Enc$），用于生成表示整段文本的向量。采用稠密向量和稀疏向量的多视图表示，从不同方面对输入序列进行建模。稠密向量（例如 $[CLS]$）能够自然地表示序列级或段落级语义。相比之下，PLM 在词汇维度上生成的稀疏向量表示 logit 频率，因此从词汇层面对句子进行建模。形式化地，给定由 $N$ 个经过分词的子词组成的序列作为输入文本。

$$\mathbb{T} = \{CLS, t_1, ..., t_N, SEP\} \quad (1)$$

输入文本经过基于 $L$ 层 Transformer 的 BERT 编码器。对于 Transformer 层 $l \in \{1, ..., L\}$，输出隐藏状态为

$$\mathbf{H}^l = \{\mathbf{h}_0^l, \mathbf{h}_1^l, ..., \mathbf{h}_N^l\} \quad (2)$$

**稠密表示** 遵循常见做法，我们通常将最后一层隐藏状态中的第一个输出隐藏向量（CLS）作为稠密表示（**DEN**）。

$$\mathbf{DEN} = \mathbf{h}_0^{last} \quad (3)$$

**稀疏表示** 稀疏向量将BERT 逻辑值表示为频率。BERT 通过将原始掩码位置的逻辑值作为标签，并使用交叉熵（CE）损失进行预训练，成为掩码语言模型。因此，BERT 在词汇表大小维度上的输出（例如，BERT-base为30522）自然地对输入序列中每个逻辑值的频率进行建模。某个逻辑值上的高频率表示由BERT 模型表征的高出现频率，而该模型的参数是可学习的。对于稀疏向量，最后一层的隐藏状态首先通过转置的嵌入矩阵$\mathbf{E}^\top$投影到词汇维度的逻辑值上。

$$\mathbf{w}_j^i = \mathbf{h}_i^{last}\mathbf{E}_j + b_j \quad (4)$$

其中 $i \in \{1, ..., N\}$ 和 $N$ 是序列长度，$j \in \{0, ..., V-1\}$ 和 $V$ 是词汇表大小。$\mathbf{w}_j^i$ 表示词汇维度中的单个逻辑值。遵循 SPLADE（Formal 等人，2021b，一个），沿序列输入轴对每个逻辑值执行最大池化，以聚合最大逻辑值。

$$s_j = \max_{i \in \{1,...,N\}} \log\left(1 + ReLU(\mathbf{w}_j^i)\right) \quad (5)$$

ReLU 函数会移除负数，并将正数保留为逻辑值频率。单调递增的对数饱和函数 $\log(1+x)$ 则会压缩较高频率，以确保稀疏性。最后，稀疏向量表示如下。

$$\mathbf{SPR} = \{s_0, s_1, ..., s_j\}, j \in \{0, ..., V-1\} \quad (6)$$

**相似度测量** 句子相似度测量（表示为 $d(q,p)$）通过对应表示向量的点积或余弦相似度计算查询（q）和段落（p）的相似度。给定一个句子级密集向量 **DEN**

以及一个词元级稀疏向量**SPR**，这里我们将两种相似度度量之和记为我们的多视图表示相似度。

$$d(q,p) = \mathbf{SPR}_q \cdot \mathbf{SPR}_p + \mathbf{SPR}_q \cdot \mathbf{SPR}_p \quad (7)$$

由于稀疏向量的存储成本极高（例如，单个稀疏向量需要 BERT-base 30522 * 4 字节），直接将完整向量存储为连续数组是不可行的。SPLADE（Formal 等人，2021b，a）等以往工作将向量舍入为离散整数，并为稀疏检索（例如 BM25）构造虚拟文档，这不适合纳入多视图表示相似度测量。得益于稀疏性，我们应用 Top-$k$ 函数，仅在微调和推理阶段保留稀疏向 $k$ 量中频率最高的若干项。Top-$k$ 在保留向量空间中大多数高频逻辑值的同时，将存储开销控制在可接受范围内。

### 3.2 采用多视图解码范式的上下文掩码自编码器

使用上下文掩码自编码进行预训练能够为检索器模型提供良好的初始化。遵循 Cot-MAE（Wu 等人，2022年），我们采用非对称编码器-解码器，并结合上下文嵌入，在预训练期间为表示向量提供自监督信号。

**编码器** 对于 BERT 编码器，给定方程 $\mathbb{T}$ 中的输入文本1，我们将 $\mathbb{T}$ 中的一定比例的词元替换为指定的掩码词元，例如$[MASK]$。

$$mask(\mathbb{T}) = \{CLS, t_1, MASK, t_3, ..., t_N, SEP\} \quad (8)$$

我们将所有 $[MASK]$ 位置的索引集合记为 $\mathbb{M}$。BERT 编码器（Enc）的预训练使用交叉熵（CE）损失来优化以下损失函数，即掩码语言建模（MLM）目标。

$$\mathcal{L}_{enc} = -\sum_{i \in \mathbb{M}} \log p(t_i | Enc(mask(\mathbb{T}))) \quad (9)$$

**解码器** 对于解码器端，我们采用两个不同的 Transformer，即自编码解码器（AE-Dec）和自回归解码器（AR-Dec），为稠密向量和稀疏向量提供多视图预任务信号。对于使用密集向量中，我们直接将密集向量与上下文句子嵌入拼接，并将其输入 AE-Dec和AR-Dec。对于使用稀疏向量进行预训练，我们首先通过与梯度分离的 [1] 嵌入矩阵 $\mathbf{E}^\top$的投影，将 **SPR** 向量投影到隐藏维度维度。然后，我们将稀疏向量进行拼接，并将其输入解码器。解码器的输入（$\mathbb{T}_{Dec}$）表示为分别使用的两个序列 [2]。

$$\mathbb{T}_{Dec-DEN} = \{\mathbf{DEN}, t_1, ..., t_N, SEP\} \quad (10)$$

$$\mathbb{T}_{Dec-SPR} = \{\mathbf{SPR} \cdot \mathbf{E}^\top, t_1, ..., t_N, SEP\} \quad (11)$$

AE-Dec 将 Transformer 编码器层作为自编码解码器，并利用掩码语言建模作为重建信号。AE-Dec 的损失函数表示为

$$\mathcal{L}_{ae-dec} = -\sum_{i \in \mathbb{M}} \log p(t_i | Dec(mask(\mathbb{T}_{Dec}))) \quad (12)$$

AR-Dec将 Transformer 解码器层作为自回归解码器，其利用因果语言模型（因果语言建模）作为生成信号。AR-Dec 的损失定义为

$$\mathcal{L}_{ar-dec} = -\sum_{i \in \mathbb{T}} \log p(t_{i+1} | Dec(\mathbb{T}_{Dec})) \quad (13)$$

结合 BERT 编码器的 MLM 损失，AE-Dec 的 MLM 重建信号和 AR-Dec 的 CLM 生成信号共同用于多视图有效预训练。

$$\mathcal{L}_{total} = \mathcal{L}_{enc} + \mathcal{L}_{ae-dec} + \mathcal{L}_{ar-dec} \quad (14)$$

正如 CoT-MAE 中所论证的，解码器侧的上下文嵌入能够有效促进 BERT 编码器表示的学习。因此，我们使用上下文段落作为解码器输入 $\mathbb{T}_{Dec}$。有关上下文段落采样方法的详细信息，请参阅 CoT-MAE（Wu 等人，2022年）。

[1]这样可以确保梯度仅通过**SPR**向量反向传播，从而使参数更新保持稳定。[2]为便于说明，在后续公式中，我们将稠密（或稀疏）向量与词元拼接成一个序列。在实际实现中，需要将稠密（或稀疏）向量与相应的词元嵌入进行整合。

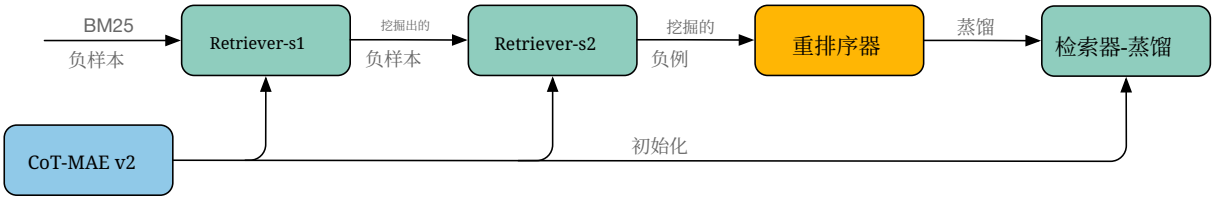



图 2：多阶段微调流程示意图。在第一阶段（s1），使用 BM25 负例训练检索器。在第二阶段（s2），使用 Retriever-s1 挖掘有效的困难负例，以微调 Retriever-s2。在 Retriever-s2 上测试得到的分数作为非蒸馏结果。在第三阶段（重排序器），训练一个更强大的单塔重排序器。在最后阶段（蒸馏），重排序器对 Retriever-distill 进行知识蒸馏。

### 3.3 微调与推理

**微调**微调在下游检索任务上进行，以验证预训练的有效性。我们使用多视图表示相似度函数 $\mathbf{d}(\mathbf{q},\mathbf{p})$ 作为距离度量。遵循 DPR 和 SPLADE（Formal 等人，2021b，a），我们使用对比学习损失函数训练稠密和稀疏检索器。

$$\mathcal{L}_{\text{ft}} = -\log \frac{\exp(d(q,p^+))/\tau)}{\sum \exp((d(q,p^+)+d(q,p^-))/\tau)} \tag{15}$$

其中 $\tau$ 是固定为 1 的温度超参数。 $\mathbf{p}^+$ 是人工标注的正例段落，而$\mathbf{p}^-$ 是通过 BM25 或稠密检索提供的困难负例集合。在微调和推理阶段使用 Top-$k$ 稀疏向量。遵循 SPLADE（Formal 等人，2021b，a），我们还引入了 FLOPs 正则项，以抑制稀疏检索器微调中的主导逻辑值频率。有关 FLOPs 的详细信息，请参阅 SPLADE 论文。

继承 CoT-MAE（Wu 等人，2022年），我们通过多阶段流程在 MS-MARCO 段落排序任务上训练检索器，如图 2所示。在第一阶段（s1）中，使用 MS-MARCO 训练集提供的 BM25 负例训练检索器。在第二阶段（s2）中，我们使用 Retriever-s1 挖掘有效的困难负例，以对 Retriever-s2 进行微调。Retriever-s2 的性能即为我们的**非蒸馏**结果。在第三阶段（重排序器）中，训练功能更强的单塔重排序器。我们直接使用 CoT-MAE 重排序器，无需重新训练。在最终阶段（蒸馏）中，通过使用知识蒸馏将重排序器的能力迁移至 Retriever-distill

KL 损失。

$$\mathcal{L}_{\text{ft}-\text{distill}} = \sum_{q} KL(d(q,p|retriever), d(q,p|reranker)) \tag{16}$$

**推理** 在推理阶段，我们将查询和段落语料库输入检索器，以获得 CLS 稠密向量和Top-$k$ 稀疏向量。我们使用公式（7）来衡量多视图查询-段落相似度。我们使用 Faiss 进行稠密检索，并使用 PyTorch 稀疏矩阵乘法进行稀疏检索。

遵循 BEIR（Thakur 等人，2021年），我们使用MS-MARCO 负例训练检索器，以在 BEIR 基准上进行域外评估。

## 4 实验设置

本节介绍我们的实验设置，包括预训练、微调、推理和基线方法。

### 4.1 预训练

**数据集** 遵循CoT-MAE（Wu 等人，2022年）和 BERT（Devlin 等人，2018），MS-MARCO 文档和 Wiki+BookCorpus 用作预训练语料库。

- **MS-MARCO**文档包含320万份文档。
- **Wiki+BookCorpus** 包含 560 万篇文档。

NLTK 句子分词器用于拆分每个文档，并动态填充至目标最大长度。MS-MARCO 的最大长度为144，Wiki+BookCorpus为 512，以便与 MS-MARCO 和 BEIR 基准段落对齐。上下文段落按照（Wu 等人，2022年）中所述的方式选取，用于上下文掩码自编码器预训练。

**实现** 继承 CoT-MAE（Wu 等人，2022年），编码器使用预训练的 12 层 BERT-base模型进行初始化，而解码器（包括 AE-Dec 和 AR-Dec）则从头开始初始化为单层。我们使用AdamW优化器对模型进行预训练，最多训练 50k步，学习率为4e-4，批量大小为 16k，并采用预热比例为 0.1 的线性调度。我们使用 16 块 Tesla A100GPU 训练模型 49 小时，随后丢弃解码器，仅保留编码器进行微调。我们将广泛使用的随机种子设置为 42，以确保结果可复现。我们还进行了统计显著性分析，以证明我们方法的鲁棒性和有效性。

### 4.2 微调与推理

**数据集与评估** 我们在 MS-MARCO 段落数据集上，对基于预训练编码器初始化的检索器进行微调。随后，我们在 MS-MARCO 段落排序（Nguyen 等人，2016年）、2019 年 TREC 深度学习（DL）赛道（Craswell 等人，2020a）和 2020 年（Craswell 等人，2020b）任务上进行评估。我们还在 BEIR 基准上进行领域外评估。

MS-MARCO（Nguyen 等人，2016年）是一个基准数据集，包含 880 万个段落以及从 Bing 搜索中收集的真实用户查询。训练集包含 50 万组由人工标注的查询-段落对，而开发集包含 6,980 条已标注查询。由于测试集未公开，因此使用开发集进行评估。遵循先前工作（Wu 等人，2022年），我们使用 MRR@10、Recall@50 和 Recall@1K 评估模型在 MS-MARCO 上的表现。

TREC 深度学习（DL）（Craswell 等人，2020a，b）赛道提供了带有更详尽标注的测试集，用于评估排序模型的实际能力。我们评估了 2019 年和 2020 年的测试集。2019 年测试集包含 200 个带标注的查询，2020 年测试集也包含 200 个带标注的查询。遵循先前的工作（Wu 等人，2022年），我们使用 NDCG@10 评估模型在 TREC 上的性能。

BEIR（Thakur 等人，2021年）是一个异构基准，包含涵盖九个任务领域的18个数据集。遵循其微调设置，我们使用 MS-MARCO 硬负样本对检索器进行微调[3]。我们在14个公开可用的用于域外 BEIR 评估的可用数据集 on.

**实现** 微调使用 MS-MARCO 数据集进行训练，并在 MS-MARCO 开发集和 TREC DL 2019/2020 测试集上进行评估。按照（Wu 等人，2022 年）的方法，我们使用 AdamW 优化器在 8 张 Tesla A100 GPU 上进行训练，学习率为 2e-5，全局批次大小为 64，训练轮数为 3，权重衰减为 0.01。查询长度为 32，段落长度为 144，负段落数量为 15，负样本深度为 200。为确保可复现性，微调的随机种子也设为 42。对于稀疏检索器的微调，稀疏向量的 Top-$k$ 设为 768，FLOPs 系数设为 0.002。

用于 BEIR 评测的微调同样使用 AdamW 优化器，在 8 块 A100 GPU 上进行训练。遵循（Thakur 等人，2021年），学习率设为 1e-5，全局批次大小为 1k，训练轮数为 10，负段落数量为 1。其他超参数与上述设置相同。

### 4.3 基线方法

我们的基线方法包括非蒸馏检索方法和蒸馏检索方法，如表1所示。我们将非蒸馏检索器与最新的先进基线方法进行比较，包括稀疏检索方法 BM25（Robertson 等人，2009）、SPLADE（Formal 等人，2021b）、稠密检索器 SEED（Lu 等人，2021年）、ColBERT（Khattab 和 Zaharia，2020）、RocketQA（Qu 等人，2020）、coCondenser（Gao 和 Callan，2021b）、SimLM（Wang 等人，2022年）、RetroMAE（Liu 和 Shao，2022年）以及 CoT-MAE（Wu 等人，2022年）。

我们将蒸馏后的检索器与多个强大的蒸馏基线进行比较，包括稀疏检索器 SPLADE v2（Formal 等人，2021a）、稠密检索器 RocketQA 第二版（Ren 等人，2021年）、AR2（Zhang 等人，2021年）、ColBERT v2（Santhanam 等人，2021年）、SimLM（Wang 等人，2022年）、MASTER（Zhou 等人，2022年）和 LexMAE（Shen 等人，2022年）。

## 5 实验结果

在本节中，我们分析实验结果，以证明所提出的 CoT-MAE v2 方法的有效性。

[3] https://sbert.net/datasets/msmarco-hard-negatives.jsonl.gz

| 模型 | MS-MARCO MRR@10 | MS-MARCO R@50 | MS-MARCO R@1k | TREC DL19 nDCG@10 | TREC DL20 nDCG@10 |
|---|---|---|---|---|---|
| BM25 † | 18.7 | 59.2 | 85.7 | 51.2 | 47.7 |
| SPLADE † (Formal 等人, 2021b) | 32.2 | - | 95.5 | 66.5 | - |
| SEED（Lu 等人，2021年） | 33.9 | - | 96.1 | - | - |
| ColBERT（Khattab 和 Zaharia，2020） | 36.0 | 82.9 | 96.8 | - | - |
| RocketQA (Qu 等人, 2020) | 37.0 | 85.5 | 97.9 | - | - |
| coCondenser (Gao 和 Callan, 2021b) | 38.2 | 86.5 | 98.4 | 71.7 | 68.4 |
| SimLM（Wang 等人，2022年） | 39.1 | - | 98.6 | - | - |
| RetroMAE (Liu 和 Shao, 2022年) | 39.3 | - | 98.5 | - | - |
| CoT-MAE（Wu 等人，2022年） | 39.4 | 87.0 | 98.7 | 70.9 | 70.4 |
| LexMAE †(Shen 等人, 2022年) | 40.8 | - | 98.5 | - | - |
| **CoT-MAE v2** | **41.4** | **89.4** | **98.7** | **74.5** | **70.8** |

表 1：非蒸馏检索器在 MS-MARCO 段落排序和 TREC DL 数据集上的主要结果。最佳分数以**粗体**标示。标有 † 的模型为稀疏检索模型。

| 模型 | MS-MARCO MRR@10 | MS-MARCO R@50 | MS-MARCO R@1k | TREC DL19 nDCG@10 | TREC DL20 nDCG@10 |
|---|---|---|---|---|---|
| SPLADE v2 † (Formal 等人, 2021a) | 36.8 | - | 97.9 | 72.9 | - |
| RocketQA 第二版 (Ren 等人, 2021年) | 38.8 | 86.2 | 98.1 | - | - |
| AR2 (Zhang 等人, 2021年) | 39.5 | 87.8 | 98.6 | - | - |
| ColBERT v2(Santhanam 等人, 2021年) | 39.7 | 86.8 | 98.4 | - | - |
| ERNIE-SEARCH(Lu 等人,2022年) | 40.1 | 87.7 | 98.2 | - | - |
| CoT-MAE (Wu 等人, 2022年) | 40.4 | 88.5 | 98.7 | - | - |
| SimLM（Wang 等人，2022年） | 41.1 | 87.8 | 98.7 | 71.2 | 69.7 |
| MASTER (Zhou等，2022年) | 41.5 | 88.6 | 98.8 | 72.7 | 71.7 |
| LexMAE †(Shen 等人, 2022年) | 42.6 | - | **98.8** | 73.7 | **72.8** |
| **CoT-MAE v2** | **43.1** | **90.2** | 98.7 | **75.6** | 72.5 |

表 2：MS-MARCO 段落排序和 TREC DL数据集上经过蒸馏的 检索器的主要结果。最佳分数以**粗体**标记。标有 † 的模型是稀疏检索模型。

### 5.1 主要结果

**非蒸馏结果** 表中的主要结果1表明，非蒸馏的 CoT-MAEv2优于多个最新的强基线方法。非蒸馏检索器在BM25 负样本或检索器挖掘的负样本上进行微调。对于非蒸馏检索器的性能，CoT-MAE v2 在 MRR@10 上优于之前的基线方法 LexMAE +0.6 个点，在 MS-MARCO 段落排序任务的 R@50上优于 CoT-MAE 2.4 个点，并且在+3TREC-DL19 任务的 nDCG @10 上优于 CoT-MAE .6 个点。

**蒸馏结果**表2 中的主要结果表明，与最新的强大重排序器蒸馏基线相比，蒸馏后的 CoT-MAE v2 达到了业界领先的性能。例如，在对比参赛的蒸馏检索器时，CoT-MAE v2 在 MRR@10 上领先 LexMAE +0.5 个百分点，在 MS-MARCO 段落排序任务的 R@50 上领先 MASTER +1.6 个百分点，在 +1.9 个百分点，在 TREC-DL 19 任务上。蒸馏后的 CoT-MAE v2 在 MS-MARCO MRR@10 上达到 43.1，甚至超过了许多强大的重排序器，并且接近其教师 CoT-MAE 重排序器的性能（MRR@10 为 43.9）。这些性能提升清楚地表明，相较于以往的基线方法，其具备领先优势。

### 5.2 BEIR基准上的域外评估

遵循 BEIR（Thakur 等人，2021年），我们使用提供的 MS-MARCO 负例训练检索器，并在 14 个公开可用数据集上评估其域外表现。如表3所示，CoT-MAE v2 的表现优于先前的基线方法[4]，平均高出 +3 分，并在 14 个数据集中的 11 个上取得最佳表现。CoT-MAE v2 在 BEIR 基准上的领先表现，展现了其在异构检索任务设置中的有效性和适应性。

总体而言，将 CoT-MAE v2 与之前的有效方法在最常见的

[4]我们没有与 LexMAE 进行比较，因为它并未报告所有数据集上的结果。

| 数据集 | BM25 | DPR | ANCE | ColBERT | Contriever | RetroMAE | MASTER | CoT-MAE v2 |
|---|---|---|---|---|---|---|---|---|
| TREC-COVID | 65.6 | 33.2 | 65.4 | 67.7 | 59.6 | 75.6 | 62.0 | **77.1** |
| NFCorpus | 32.5 | 18.9 | 23.7 | 30.5 | 32.8 | 30.1 | 33.0 | **33.5** |
| NQ | 32.9 | 47.4 | 44.6 | 52.4 | 49.8 | 49.0 | 51.6 | **53.9** |
| HotpotQA | 60.3 | 39.1 | 45.6 | 59.3 | 63.8 | 63.8 | 58.9 | **67.2** |
| FiQA-2018 | 23.6 | 11.2 | 29.5 | 31.7 | 32.9 | 30.1 | 32.8 | **33.1** |
| ArguAna | 31.5 | 17.5 | 41.5 | 23.3 | 44.6 | 48.1 | 39.5 | **48.2** |
| Touché-2020 | **36.7** | 13.1 | 24.0 | 20.2 | 23.0 | 24.3 | 32.0 | 30.3 |
| CQADupStack | 29.9 | 15.3 | 29.6 | 35.0 | 34.5 | **38.2** | 32.7 | 32.2 |
| Quora | 78.9 | 24.8 | 85.2 | 85.4 | **86.5** | 85.6 | 79.1 | 86.1 |
| DBPedia | 31.3 | 26.3 | 28.1 | 39.2 | 41.3 | 38.5 | 39.9 | **42.6** |
| SCIDOCS | 15.8 | 7.7 | 12.2 | 14.5 | **16.5** | 15.0 | 14.1 | **16.5** |
| FEVER | 75.3 | 56.2 | 66.9 | 77.1 | 75.8 | 71.9 | 69.2 | **81.2** |
| Climate-FEVER | 21.3 | 14.8 | 19.8 | 18.4 | 23.7 | 21.4 | 21.5 | **27.5** |
| SciFact | 66.5 | 31.8 | 50.7 | 67.1 | 67.7 | 64.8 | 63.7 | **69.2** |
| 平均值 | 43.0 | 25.5 | 40.5 | 44.4 | 46.6 | 46.9 | 45.0 | **49.9** |

表 3：Ou 在 BEIR 基准上的域外评估。对比中表现更好的分数以**粗体**标记。 **d**.

所使用的基准数据集表明，CoT-MAE v2 预训练过程能够有效提升稠密检索。该提升源于两个方面。

- 一方面，CoT-MAEv2 整合了密集表示和稀疏表示，其中密集表示自然地捕获段落级别的特征，而稀疏表示捕获词汇层面的语义，如第 3.1 节所述。

- 另一方面，CoT-MAE v2 使用自编码解码器（AE-Dec）和自回归解码器（AR-Dec）进行表示预训练，其中 AE-Dec 专注于掩码语言建模重建任务，而 AR-Dec 专注于因果语言建模生成任务，为表示学习提供丰富的自监督信号。

CoT-MAE v2 的多视图特性有助于高效预训练。

## 6 分析

鉴于 CoT-MAE v2 的领先性能，本节将讨论并分析若干科学问题（**问题**）。

**Q1** 多视图表示预训练如何助力取得这些结果?

**Q2** 多视图解码范式如何促成这些结果?

**Q3** 仅使用一种表示进行微调和推理时，性能如何变化?

| 模型 | MRR@10 | R@50 | R@1k |
|---|---|---|---|
| CoT-MAE v2 | **41.4** | **89.4** | **98.7** |
| **使用单视图表示瓶颈进行预训练** | | | |
| 仅使用 DEN | 40.1 (-1.3) | 88.0 (-1.4) | 98.6 (-0.1) |
| 仅使用 SPR | 40.9 (-0.5) | 89.0 (-0.4) | 98.1 (-0.6) |
| **采用单视图解码范式进行预训练** | | | |
| 仅使用 AE-Dec | 41.0 (-0.4) | 89.4 | 98.5 (-0.2) |
| 仅使用 AR-Dec | 40.9 (-0.5) | 89.2 (-0.2) | 98.4 (-0.3) |
| **使用单视图表示进行微调** | | | |
| 仅使用 DEN | 39.4 (-2.0) | 87.7 (-1.7) | 98.7 |
| 仅使用 SPR | 40.2 (-1.2) | 88.2 (-1.2) | 98.4 (-0.3) |

表4：多视图表示和多视图解码范式预训练的消融研究。得分在非蒸馏检索器上进行测试。

**Q4** 多视图表示的总维度大小是多少？缩小表示的维度时，性能如何变化?

**Q5** 多视图预训练的鲁棒性如何?

以下将逐一分析相应的答案（**A**），并提供详细的消融数据。所有分析均基于非蒸馏检索器。

### 6.1 多视图表示与多视图解码范式的影响

表4展示了预训练和微调阶段多视图表示的消融实验结果。

| DEN 和 SPR 的维度 | MRR@10 | R@50 | R@1k |
|---|---|---|---|
| 768 | 41.4 | 89.4 | 98.7 |
| 384 | 40.9 | 89.1 | 98.4 |

表5：降低稠密向量和稀疏向量维度大小的消融研究。这些得分在非蒸馏检索器上进行测试。

**A1** 正如第3.1节所述，多视图表示旨在捕捉不同层次的句子特征，因此同时使用它们能够取得更好的性能。仅使用稠密表示或稀疏表示进行预训练时，MS-MARCO 上的 MRR@10 分别降至 40.1 和 40.9。这表明，我们提出的预训练架构中的多视图表示有助于更好地进行表示学习。

**A2** 如第 3.2 节所述，多视图解码范式为上下文掩码自编码器预训练同时提供了重建信号和生成信号。这些多视图解码信号有助于更好地学习表示。仅使用 AE-Dec 或 AR-Dec 进行预训练时，MS-MARCO 上的 MRR@10 性能会降至 41.0 或 40.9。因此，多视图解码范式有助于提升表示学习效果。

**A3** 由于我们采用多视图表示，因此在微调和推理阶段都会使用稠密向量和稀疏向量。因此，我们需要探究仅使用一种表示类型时性能会如何变化。我们尝试仅使用单视图表示进行微调。在 MS-MARCO 上，MRR@10 性能降至 39.4 和 40.2。这进一步表明，多视图表示有助于提升段落检索性能。

### 6.2 多视图表示降维的性能

尽管 CoT-MAE v2 取得了显著的性能提升，但多视角表示也带来了额外的表示维度需求。

**A4** 对于 DPR 或 CoT-MAE 等传统稠密检索方法，单个稠密表示的维度为 768。相比之下，CoT-MAE v2 进一步引入了一个 top-768 稀疏向量，使表示的总维度增加了一倍。在此，我们尝试将稠密向量和稀疏向量的维度都缩减至 768/2=384 ，

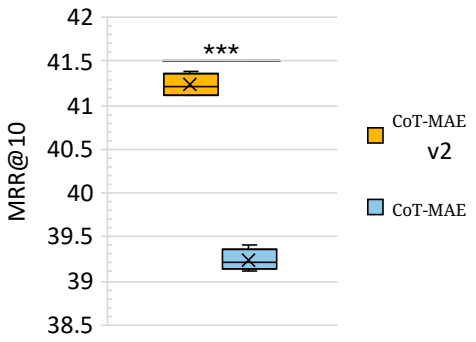



图 3：多视角预训练与 CoT-MAE 基线的性能比较。通过更换随机种子重复进行预训练。分数来自在 MS-MARCO段落排序任务上测试的非蒸馏检索器。学生t检验表明，结果相较于此前的基线具有统计显著性。

从而使总维度与 DPR 或 CoT-MAE 相同。如表 5 所示，将向量维度减半后，MS-MARCO 上的 MRR@10 降至 40.9。但这一分数仍远胜于 CoT-MAE(39.4) 等先前基线。因此，多视图 CoT-MAE v2 在降维场景下表现出显著的鲁棒性。

### 6.3 多视角预训练的鲁棒性。

**A5** 为了测试我们提出的多视角预训练方法的鲁棒性，我们使用不同的随机种子（43、2022年、2023）额外重复预训练 3 次。如图3所示，CoT-MAE v2 明显优于此前的基线，且性能稳定性良好，展现出其鲁棒性和有效性。

### 6.4 案例研究

为了定性地理解我们提出的预训练方法所带来的收益，我们从 CoT-MAE v2 首次召回的段落中选取了一些示例，并在表6中与其基线方法 CoT-MAE 进行比较。

- 第一个查询与“用于”（以棕色标记）的用法相关。CoT-MAE 基线模型未能找到相关答案，却返回了一个段落，其中包含关于“由……制成”和“被认为是绿色的”（以红色标记）的内容。相比之下，CoT-MAE v2 检索出了准确答案（以蓝色标记）。

- 第二个查询询问“图书”（标记为棕色）的作者。CoT-MAE 基线返回了关于“电影”（标记为红色）的段落。相比之下，CoT-MAE v2 检索到了“幻影忍者图书”

| 相关 | 模型 | 排名第1的段落 |
|---|---|---|
| | | **查询:** 什么是绿色皂用于 |
| ✗ | CoT-MAE | **段落:** 如何使用绿色皂。绿色皂有许多不同的用途。它由制成<br>纯植物油和甘油。虽然这种皂不是绿色的，但它被认为是绿色皂<br>因为它是有机的、可生物降解且对环境安全。 |
| ✓ | CoT-MAE v2 | **段落:** 博客——绿色皂用于纹身和穿孔工作室，可用于从准备<br>纹身或穿孔部位到清洁纹身机等各种用途。它能很好地去除污垢、血液和<br>纹身墨水。由于它不具腐蚀性，因此也可以用来安全浸泡设备，使污垢松脱<br>清洁和消毒前已干燥的碎屑。 |
| | | **查询:** 是谁写的幻影忍者这本书 |
| ✗ | CoT-MAE | **段落:** 主条目：《乐高幻影忍者大电影》。哈格曼兄弟丹和凯文<br>哈格曼创作了《乐高幻影忍者：旋风忍者大师》，并参与撰写了《乐高大电影》的故事。他将为这部<br>电影改编作品执笔，该片将采用一种偏离原作的新演绎方式，<br>电视剧。 |
| ✓ | CoT-MAE v2 | **段落:** 《寇：地之忍者》是一本 幻影忍者图书 ，于2012年出版， 由格雷格·法什蒂撰写。<br>这本书讲述了寇为掌握土元素和旋风术而经历的冒险。 |
| | | **查询:** 恐龙战队何时在欧洲发行 |
| ✗ | CoT-MAE | **段落:** 主文：最初，《恐龙战队》原定于2016年7月22日上映，<br>也就是暑期电影档期的后半段。然而，四月时，<br>制片厂决定推迟该项目，直到2017年1月13日——影片的制作计划于<br>2016年1月开始制作。 |
| ✓ | CoT-MAE v2 | **段落:** 《恐龙战队：超级传奇》的发行日期为10月23日（美国），以及2日<br>欧洲的十一月适用于 Wii、DS、PS2、GameCube（作为最后一款 GameCube 游戏！）和 PC。该<br>游戏将通过集结可选的恐龙战队成员来庆祝恐龙战队成立15周年<br>涵盖该系列的十五季，从《恐龙战队：初代》到《恐龙战队》：<br>《超能战队：极速行动》不仅会包括角色，还会包括佐德。 |

表 6：不同模型在MS-MARCO段落排序数据集开发集上召回的排名第 1 段落示例。查询的核心语义以棕色标注。我们将基线方法CoT-MAE中不匹配的含义标为红色，将 CoT-MAE v2 中匹配的含义标为蓝色。

其作者姓名（以蓝色标记）。

- 第三个查询要求获取“欧洲恐龙战队的发行日期”（标为棕色）。尽管CoT-MAE 基线检索到了与恐龙战队相关的段落，但该段落只告知“原始发行日期、延期安排和电影制作日期”（标为红色）。这些内容都没有告知欧洲的发行日期。相比之下，我们的 CoT-MAE v2准确检索到了真正的答案“欧洲*11*月*2*日”（标为蓝色）。

这些示例表明，CoT-MAEv2 能够捕捉段落中更精确的语义，证明 CoT-MAE v2 预训练方法比以往的预训练方法更有效。

## 7 结论

本文提出了一种多视图上下文掩码自编码预训练架构，以实现更好的段落检索。实验结果表明，多视图表示和多视图解-码范式显著有助于提高效-检索性能。我们的方法也表现出良好的鲁棒性和稳定性。未来，我们将进一步探索融入新的预训练范式，以获得更有效、更鲁棒的检索器。

## 参考文献


Yang Bai、Xiaoguang Li、Gang Wang、Chaoliang Zhang、Lifeng Shang、Jun Xu、Zhaowei Wang、Fang-shan Wang 和 Qun Liu。2020。Sparterm：学习基于术语的稀疏表示，实现快速文本检索。arXiv 预印本，arXiv:2010.00768。

Wei-Cheng Chang、Felix X Yu、Yin-Wen Chang、Yiming Yang 和 Sanjiv Kumar。2020。用于基于嵌入的大规模检索的预训练任务。arXiv 预印本 arXiv:2002.03932。

Nick Craswell、Bhaskar Mitra、Emine Yilmaz、Daniel Campos 和 Ellen M Voorhees。2020a。TREC 2019 深度学习赛道概述。arXiv 预印本 arXiv:2003.07820。

Nick Craswell、Bhaskar Mitra、Emine Yilmaz、Daniel FernandoCampos 和 Ellen M.Voorhees。2020b。TREC 2020 深度学习赛道概述。arXiv，abs/2003.07820。

ZhuyunDai和JamieCallan。2020a。面向即席检索的上下文感知的文档术语加权。载于《2020年万维网大会》论文集，第1897–1907页。

Zhuyun Dai 和 Jamie Callan。2020b。面向第一阶段段落检索的上下文感知术语加权。载于《第43届国际 ACM SIGIR 信息检索研究与发展会议论文集》，第 1533–1536 页。

Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。2018。BERT：用于语言理解的深度双向变换器预训练。arXiv 预印本 arXiv:1810.04805。

Yixing Fan、Xiaohui Xie、Yinqiong Cai、Jia Chen、Xinyu Ma、Xiangsheng Li、Ruqing Zhang、Jiafeng Guo 和 Yiqun Liu。2021年。信息检索中的预训练方法。arXiv 预印本 arXiv:2111.13853。

Thibault Formal、Carlos Lassance、Benjamin Piwowarski 和 Stéphane Clinchant。2021a。SPLADE v2：用于信息检索的稀疏词汇扩展模型。arXiv 预印本 arXiv:2109.10086。

Thibault Formal、Benjamin Piwowarski 和 Stéphane Clinchant。2021b。SPLADE：用于第一阶段排序的稀疏词汇扩展模型。载于《第44届国际 ACM SIGIR 信息检索研究与发展会议论文集》，第 2288–2292 页。

Jianfeng Gao、Chenyan Xiong、Paul Bennett 和 Nick Craswell。2022年。面向对话式信息检索的神经方法。arXiv 预印本，arXiv:2201.05176。

Luyu Gao 和 Jamie Callan。2021a。Condenser：用于稠密检索的预训练架构。arXiv 预印本 arXiv:2104.08253。

Luyu Gao 和 Jamie Callan。2021b。用于稠密段落检索的无监督语料库感知语言模型预训练。arXiv 预印本 arXiv:2108.05540。Luyu Gao、Zhuyun Dai 和 Jamie Callan。2021年。Coil：在信息检索中重新审视精确词汇匹配与上下文化倒排列表。arXiv 预印本 arXiv:2104.07186。Jiafeng Guo、Yinqiong Cai、Yixing Fan、Fei Sun、Ruqing Zhang 和 Xueqi Cheng。2022年。第一阶段检索的语义模型：全面综述。《ACM 信息系统汇刊》（TOIS），40(4):1–42。

Sebastian Hofstätter、Sheng-Chieh Lin、Jheng-Hong Yang、Jimmy Lin 和 Allan Hanbury。2021年。通过均衡的主题感知采样，高效地训练有效的稠密检索器。载于《第44届国际 ACM SIGIR 信息检索研究与发展会议论文集》，第 113–122 页。

Samuel Humeau、Kurt Shuster、Marie-Anne Lachaux 和 Jason Weston。2019年。Poly-Encoder：用于快速、准确多句评分的 Transformer 架构与预训练策略。arXiv 预印本，arXiv:1905.01969。

VladimirKarpukhin、Barlas Oˇguz、SewonMin、Patrick Lewis、Ledell Wu、Sergey Edunov、Danqi Chen 和 Wen-tau Yih。2020年。用于开放域问答的稠密段落检索。arXiv 预印本 arXiv:2004.04906。

Omar Khattab 和 Matei Zaharia。2020。ColBERT：通过在 BERT 上进行上下文化晚期交互实现高效且有效的段落搜索。载于《第43届国际 ACM SIGIR 信息检索研究与发展会议论文集》，第 39–48页。

Jinhyuk Lee、Mujeen Sung、Jaewoo Kang 和 Danqi Chen。2020年。大规模学习短语的稠密表示。arXiv 预印本 arXiv:2012.12624。

Jimmy Lin 和 Xueguang Ma。2021年。关于 DeepImpact、COIL 以及信息检索技术概念框架的几点简要说明。arXiv 预印本 arXiv:2106.14807。

Jimmy Lin、Rodrigo Nogueira 和 Andrew Yates。2021a。《用于文本排序的预训练 Transformer：BERT 及其后续发展》。《人类语言技术综合讲座》，14(4):1–325。

Sheng-Chieh Lin、Jheng-Hong Yang 和 Jimmy Lin。2021b。《利用紧密耦合教师进行稠密检索的知识蒸馏中的批内负例》。《第六届 NLP 表示学习研讨会论文集》（RepL4NLP-2021），第 163–173 页。

Yinhan Liu、Myle Ott、NamanGoyal、JingfeiDu、Mandar Joshi、Danqi Chen、Omer Levy、Mike Lewis、Luke Zettlemoyer 和 Veselin Stoyanov。2019 年。

RoBERTa：一种稳健优化的 BERT 预训练方法。arXiv 预印本 arXiv:1907.11692。

Zheng Liu 和 Yingxia Shao。2022年。RetroMAE：通过掩码自编码器预训练面向检索的 Transformer。arXiv 预印本 arXiv:2205.12035。

Shuqi Lu、Di He、Chenyan Xiong、Guolin Ke、Waleed Malik、Zhicheng Dou、Paul Bennett、Tie-Yan Liu 和 Arnold Overwijk。2021年。少即是多：利用弱解码器预训练用于稠密文本检索的强大孪生编码器。载于《2021年自然语言处理实证方法会议论文集》，第2780–2791页。

Yuxiang Lu、Yiding Liu、JiaxiangLiu、Yunsheng Shi、Zhengjie Huang、Shikun Feng Yu Sun、Hao Tian、Hua Wu、Shuaiqiang Wang、DaweiYin 等。2022年。Ernie-search：通过即时自蒸馏连接跨编码器与双编码器，用于密集段落检索。arXiv 预印本 arXiv:2205.09153。

Xinyu Ma、Jiafeng Guo、Ruqing Zhang、Yixing Fan 和 XueqiCheng。2022年。通过对比式跨度预测预训练判别性文本编码器以进行稠密检索。arXiv 预印本 arXiv:2204.10641。

Tri Nguyen、Mir Rosenberg、Xia Song、Jianfeng Gao、Saurabh Tiwary、Rangan Majumder 和 Li Deng。2016年。Ms marco：一个由人工生成的机器阅读理解数据集。载于 CoCo@ NIPs。

RodrigoNogueira 和 Jimmy Lin。2019。从 doc2query 到doctttttquery。在线预印本，6。YingqiQu、Yuchen Ding、Jing Liu、KaiLiu、RuiyangRen、Wayne Xin Zhao、DaxiangDong、Hua Wu 和HaifengWang。2020年。Rocketqa：一种面向开放域问答的密集段落检索优化训练方法。arXiv 预印本 arXiv:2010.08191。Ruiyang Ren、Yingqi Qu、JingLiu、Wayne Xin Zhao、Qiaoqiao She、Hua Wu、HaifengWang 和 Ji-RongWen。2021年。Rocketqav2：一种密集段落检索与段落重排序的联合训练方法。arXiv 预印本 arXiv:2110.07367。Stephen Robertson、Hugo Zaragoza 等。2009。概率相关性框架：BM25 及其他方法。《信息检索领域的基础与趋势》，3(4)：333–389。Keshav Santhanam、Omar Khattab、Jon Saad-Falcon、Christopher Potts和 MateiZaharia。2021年。Colbertv2：通过轻量级晚期交互实现有效且高效的检索。arXiv 预印本 arXiv:2112.01488。Tao Shen、Xiubo Geng、Chongyang Tao、CanXu、Xiaolong Huang、Binxing Jiao、LinjunYang 和DaxinJiang。2022年。Lexmae：面向大规模检索的词汇瓶颈预训练。arXiv 预印本 arXiv:2208.14754。

Nandan Thakur、Nils Reimers、Andreas Rücklé、Ab- hishekSrivastava和Iryna Gurevych。2021年。BEIR：用于零样本评估信息检索模型的异构基准。arXiv 预印本 arXiv:2104.08663。

Liang Wang、Nan Yang、XiaolongHuang、Binxing Jiao、Linjun Yang、Daxin Jiang、RanganMajumder和 Furu Wei。2022年。Simlm：基于表征瓶颈的稠密段落检索预训练。arXiv 预印本 arXiv:2207.02578。

吴星、马光远、林萌、林子佳、Zhongyuan Wang 和胡松林。2022年。用于密集段落检索的上下文掩码自动编码器。arXiv预印本 arXiv:2208.07670。

Lee Xiong、Chenyan Xiong、Ye Li、Kwok-Fung Tang、Jialin Liu、Paul Bennett、Junaid Ahmed 和 Arnold Overwijk。2020年。用于稠密文本检索的近似最近邻负对比学习。arXiv 预印本 arXiv:2007.00808。

Shi Yu、Zhenghao Liu、Chenyan Xiong、Tao Feng 和 Zhiyuan Liu。2021年。少样本对话式稠密检索。载于《第44届国际 ACM SIGIR 信息检索研究与发展会议论文集》，第829–838页。

Jingtao Zhan、Jiaxin Mao、Yiqun Liu、Jiafeng Guo、Min Zhang 和 Shaoping Ma。2021年。利用困难负例优化稠密检索模型训练。载于《第44届国际 ACM SIGIR 信息检索研究与发展会议论文集》，第1503–1512页。Jingtao Zhan、Jiaxin Mao、Yiqun Liu、Min Zhang 和 Shaoping Ma。2020年。Repbert：用于第一阶段检索的上下文化文本嵌入。arXiv 预印本 arXiv:2006.15498。Hang Zhang、Yeyun Gong、Yelong Shen、Jiancheng Lv、Nan Duan 和 Weizhu Chen。2021年。用于稠密文本检索的对抗式检索器-排序器。arXiv 预印本 arXiv:2110.03611。Yanzhao Zhang、Dingkun Long、Guangwei Xu 和 Pengjun Xie。2022年。HLATR：利用混合列表感知 Transformer 重排序增强多阶段文本检索。arXiv 预印本 arXiv:2205.10569。Kun Zhou、Xiao Liu、Yeyun Gong、Wayne Xin Zhao、Daxin Jiang、Nan Duan 和 Ji-Rong Wen。2022年。MASTER：多任务预训练瓶颈掩码自编码器能够实现更优的稠密检索。arXiv 预印本 arXiv:2212.07841。Yunchang Zhu、Liang Pang、Yanyan Lan、Huawei Shen 和 Xueqi Cheng。2021年。面向开放域问答的自适应信息搜寻。arXiv 预印本 arXiv:2109.06747。