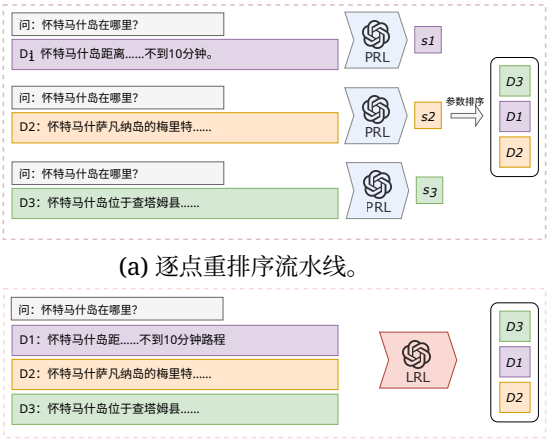
arXiv:2305.02156v1 [cs.IR] 3 May 2023

使用大语言模型进行零样本列表式文档重排序

Xueguang Ma, Xinyu Zhang,Ronak Pradeep,Jimmy Lin加拿大滑铁卢大学 David R. Cheriton 计算机科学学院
{x93ma, x978zhang, rpradeep, jimmylin}@uwaterloo.ca

摘要

基于双编码器或交叉编码器架构的有监督排序方法在多阶段文本排序任务中取得了成功，但它们需要大量相关性判断作为训练数据。在这项工作中，我们提出了基于大语言模型的**列表式重排序器**（LRL），无需使用任何任务特定训练数据，即可实现出色的重排序效果。与现有的逐点排序方法不同，后者独立为文档打分并根据分数进行排序，而 LRL 则直接根据候选文档生成重新排序后的文档标识符列表。在三个 TREC 网页搜索数据集上的实验表明，在对第一阶段检索结果进行重排序时，LRL 不仅优于零样本逐点方法，还可以作为最终阶段重排序器，对逐点方法排名靠前的结果进行优化，从而提高效率。此外，我们将该方法应用于近期多语言检索数据集 MIRACL 的若干子集，结果显示该方法有望推广至不同语言。[1]



(a) 逐点重排序流水线。

(b) 列表式重排序流水线。

图 1：点式重排序流程与列表式重排序流水线的比较。

1 引言

文本检索是许多开放域自然语言理解能力的重要上游任务，例如问答和事实核查。它旨在针对给定的查询，查找并排序最相关的文档或文本片段（Chen 等人，2017；Petroni 等人，2021年）。有监督检索器和重排序器已成功用于构建多阶段排序流程，从而在多个领域和语言中实现文本排序（Nogueira 和 Cho，2019年；Thakur 等人，2021年；Bonifacio 等人，2021年）。然而，这些方法依赖大量标注数据，例如 MS MARCO 数据集（Bajaj 等人，2016），用于训练模型。一个尚未解决的问题是，如何构建一种无需使用任何人工相关性判断也能表现良好的文本排序流程。

在当前多阶段排序范式中，重排序器通常以查询-文档对为输入，并输出相关性分数。通常，候选文档列表中的文档会被独立处理，彼此之间的信息不会纳入考虑。以往关于有监督重排序的研究通过在训练过程中基于成对损失或列表式损失（例如，对比学习）更新模型来解决这一不足；这些损失根据与同一查询相关的一对或一组候选文档计算得出（Gaoet al.，2021年；Zhuang et al.，2022年）。这样做能促使模型学习文档之间的相对相关性。然而，未经任何任务特定训练的零样本重排序器无法利用这种方法。

受大语言模型强大零样本效果的启发（Ouyang 等人，2022年；Sanh 等人，2022年；Wei 等人，2022年），我们提出了基于大语言模型的零样本列表式重排序器（LRL）。具体而言，我们使用 GPT-3 作为一种零样本

[1]本研究于2022年末开展，并被 ACL 2023 拒稿。

列表式重排序器，它以一个查询和一组候选文档为输入，并根据文档与查询的相关性直接生成重新排序后的文档标识符列表。我们假设，通过同时考虑多个文档，零样本列表式重排序器可以比逐点重排序器更有效。图 1 对比了逐点和列表式流程。

我们在多种数据集上开展实验，以评估 LRL 的有效性。这些数据集包括历时三年的 TREC DL 共享任务（DL19、DL20 和 DL21）（Craswell 等人，2020年，2021年，2022年），以及来自 MIRACL 的三个非英语数据集（Zhang 等人，2022b年）。结果表明，在 TREC DL 数据集上，LRL 的表现优于逐点重排序器，平均 nDCG@10 比当前已知最佳的零样本重排序器 UPR 高约 6 个百分点。LRL 的表现也优于 BM25，在 MIRACL 的一个子集（Zhang 等人，2022b年）上，nDCG@10 平均提升了 15 个百分点。MIRACL 是一个较新的多语言文本检索数据集。尽管有人担忧大语言模型中的训练数据与测试数据相互混杂，但解读这些结果时应考虑以下缓解因素：首先，我们的核心论点关乎逐点方法与成对方法的比较，这与“数据集污染”问题正交。其次，MIRACL 是一个相对较新的数据集，因此较不可能受到这些问题的影响（至少目前如此）。

2 背景与相关工作

给定一个语料库 $\mathcal{C} = \{D_1, D_2, ..., D_n\}$，其中包含一组文档和一个查询 q，**检索器** 的任务是返回一个 k 来自 $\mathcal{C}$ 的文档列表，这些文档与查询 q最相关，相关性根据某项指标衡量，其中 $k \ll |\mathcal{C}|$。**重排序器** 的任务是根据相同或不同的指标，进一步提升排序列表的质量。检索器与重排序器共同构成文本排序的多阶段排序流程（Nogueira 等人，2019年；Gao 等人，2021年）。尽管近期有监督神经检索器的研究在文本排序任务中取得了很大成功，但它们都需要大量针对特定任务、由人工标注的查询-文档对来训练有监督模型。在这项工作中，我们关注零样本设置，即模型不接受任何针对特定任务的监督训练。

BM25（Robertson 和 Zaragoza，2021年）是一种基于概率相关性模型的传统词汇检索器，根据词频和逆文档频率等统计信息估计查询-文档相关性。它不需要人工标注数据，因此天然适用于零样本约束。Contriever（Izacard 等人，2021年）是一种通过使用从原始文本中截取的伪查询—文档对进行动量对比学习训练的无监督稠密检索器。它同样不需要特定任务的有监督训练。

在 BM25 和 Contriever 的基础上，UPR（Sachan 等人，2022年）采用零样本问题生成模型，根据查询似然对检索到的文档进行重排序。具体而言，UPR 使用预训练语言模型计算以文档为条件的查询词元的平均对数似然，从而估计文档 D 与查询 q 相关的对数概率。根据我们的实验结果，{BM25, Contriever} +UPR 重排序构成了最先进水平（SoTA）的零样本多阶段排序流程（Sachan 等人，2022年）。

近期工作如 InPars（Bonifacio 等人，2022年；Boytsov 等人，2023）以及 Promptagator（Dai 等人，2022年）探索使用大语言模型为文档生成合成查询，以构造正向的查询–文档对，作为检索器或重排序器的训练数据。类似地，HyDE（Gao 等人，2022年）使用大语言模型生成假设文档来扩充查询，以用于无监督稠密检索。在这项工作中，我们在多阶段排序流程中直接将大语言模型用作重排序器。

3 方法

3.1 列表式与逐点式重排序器

给定一个用户查询 q 以及来自语料库或上一阶段排序的候选文档$\{D_1, \ldots, D_n\}$，逐点式重排序器（例如 UPR）会接收该查询 q 和单个文档 D_i，并输出一个分数 $s_{q,\ D_i}$ 用于表示查询-文档相关性。随后，根据这些分数对候选文档重新排序。即使重排序器基于序列到序列（seq-to-seq）模型，这种方法仍然适用。以 monoT5（Nogueira 等人，2020年）为例：它以 seq-to-seq T5（Raffel 等人，2020年）为骨干网络，要求模型生成 `True` 或`False` 标记作为输出。该重排序器将这两个标记的逻辑值作为分类任务进行优化，并为每个查询-文档对输出一个分数。

与逐点重排序器不同，我们提出了 LRL，这是一种利用大语言模型的零样本列表式重排序器。具体而言，LRL 接收查询 q和文档列表 $\{D_1, ..., D_m\}$ 作为

输入并返回输入文档标识符的重新排序列表。关键区别在于，LRL 在重排序时会同时考虑多个文档的信息。在 Transformer模型的背景下，这意味着模型可以关注所有候选文档，以确定它们的相对排序位置。此外，列表式重排序器会直接输出一个有序列表（由文档标识符组成），这构成了一个“序列到序列”过程，非常适合大语言模型。点式重排序流程与列表式重排序流程之间的比较见图 1。

3.2 提示设计

我们使用一个简单的指令模板，提示大语言模型执行零样本列表式重排序：

段落1 = {passage_1}

... 段落10 ={passage_10} 查询 = {query} 段落 = [段落1，...，段落10]根据相关性对这些段落进行排序。已排序段落 = [

候选段落逐行列出，后面依次是目标查询和一条指令。末尾的方括号有意不闭合，以鼓励模型补全“已排序段落”，并给出一个段落 ID 列表，其中第一项应为最相关的段落。

为了公平比较列表式重排序和逐点评排序，我们还为零样本逐点评排序设计了一个变体提示。我们指示模型按如下方式预测查询—段落对的相关性：

段落：{passage} 查询：{query} 该段落与查询相关吗？请回答 True/False。回答：

这使第一个解码词元的概率分布集中于 True 和 False 这两个词元上，我们使用解码出 True 这一词元的概率作为相关性分数。我们将这一基线记为 PRL（**P**点式 **R**重排序器，使用 **LLM**）。

3.3 渐进式重排序

受运行时间和内存开销的限制，基于 Transformer 的模型会限制输入长度；例如，GPT-3 将最大输入长度限制为 4,000 个 token。我们的列表式重排序器可能需要对长度超过模型处理能力的文档列表进行重排序。我们提出使用滑动窗口策略来解决这一问题。

具体而言，假设模型一次能够对 m 篇文档进行重排序，为了对长度为 $n > m$ 的列表排序，我们先对列表末尾的 m 篇文档进行排序。然后将排序窗口向列表开头移动 $m/2$，并再次进行重排序。重复此操作，直到窗口到达列表开头。理想情况下，这一策略应能在一轮操作中将最相关的 $m/2$ 篇文档“提升”到列表开头。请注意，这一策略无法完全重新排列整个候选列表，但我们证明了它在改善排名靠前的结果方面的有效性。

4 实验设置

为了展示 LRL 的有效性，我们将其与现有的代表性无监督排序流程进行比较：BM25、Contriever、UPR，以及我们的基线 PRL。

我们在 TREC 深度学习（DL）段落检索任务上评估我们的方法：DL19、DL20 和 DL21（Craswell 等人，2020年，2021年，2022年）。DL19 和 DL20 使用 MS MARCO v1 段落语料库，其中包含 880 万个段落；DL21 使用 MS MARCO v2 段落语料库，其中包含 1.38 亿个段落。

我们还在 MIRACL（Zhang 等人，2022b年）上进行了评估。MIRACL 是一个多语言数据集，专注于 18 种不同语言的临时检索。受成本限制，我们仅在三种语言上开展实验：中文、斯瓦希里语和约鲁巴语，以验证我们的方法在非英语语言（包括高资源和低资源语言）上的有效性。

我们的实现基于最新的[2] GPT- 3（Brown 等人，2020年）API 版本。对于 TREC DL 数据集，我们对 PRL 和 LRL 均使用`text-davinci-003`。由于输入序列长度限制，LRL 使用 10 个候选段落（即$m = 10$），并在必要时截断每个段落的长度以满足长度限制。若要对大小超过 10 的候选列表进行重排序（例如，对排名前 20 的段落进行排序），我们使用第 3.3 节所述的渐进式重排序策略。

对于 MIRACL 数据集的重排序，我们使用`code-davinci-002`，它可接受的输入词元数量是`text-davinci-003`的两倍。我们选择支持更长输入长度的模型，因为我们观察到 GPT-3 往往会对许多非英语语言进行过度分词。

[2]在我们进行实验时。

	来源		DL19		DL20		DL21	
	前一项。	前-k	nDCG@10	MRR@10	nDCG@10	MRR@10	nDCG@10	MRR@10
			零样本					
(1) BM25	None	$\|\mathcal{C}\|$	0.5058	0.7024	0.4796	0.6533	0.4458	0.4981
(2) Contriever	None	$\|\mathcal{C}\|$	0.4454	0.5928	0.4213	0.5408	–	–
(3) UPR	BM25	100	0.5910	0.6494	0.5958	0.7247	0.5621	0.6956
(4) PRL	BM25	100	0.5975	0.7347	0.6088	0.7699	0.5678	0.7148
(5) LRL	BM25	100	0.6580	0.8517	0.6224	0.8230	0.5996	**0.8113**
(6) LRL	UPR	10	0.6382	0.8320	0.6357	**0.8256**	0.5867	0.7543
(7) LRL	UPR	20	0.6561	**0.8659**	**0.6364**	0.8129	0.6035	0.7464
(8) LRL	PRL	10	0.6369	0.8085	0.6116	0.7841	0.5844	0.7315
(9) LRL	PRL	20	**0.6650**	0.8405	0.6349	0.8237	**0.6260**	0.7689
			有监督					
(a) DPR	None	$\|\mathcal{C}\|$	0.6297	0.7388	0.6480	0.8184	–	–
(b) TCT_ColBERT	None	$\|\mathcal{C}\|$	0.7210	**0.8864**	0.6854	0.8392	0.5001	0.6527
(c) MonoBERT	BM25	1000	0.7233	0.8566	0.7218	0.8530	0.6098	0.7278
(d) MonoELECTRA	DPR	1000	**0.7557**	0.8748	**0.7450**	**0.8650**	–	–

表1：三个 TREC 深度学习赛道上不同排序流程的 nDCG@10 和 MRR@10。在“来源”列中，“前阶”表示前一阶段，因此第一阶段检索器对应“无”。UPR 是此前的最先进水平。PRL 和 LRL 是本文提出的方法。所有有监督部分中的模型均在 MS MARCO（Bajaj等人，2016）上训练。最佳无监督和有监督结果以粗体标出。

	中文	斯瓦希里语	约鲁巴语
	零样本		
（1）BM25	0.180	0.383	0.406
(2) mContriever	0.212	0.368	0.187
(3) 混合方法（1 和 2）	0.307	0.510	0.422
(4) LRL	**0.344**	**0.542**	**0.533**
	有监督		
(a) mDPR (MS)	**0.512**	0.299	0.444
(b) mDPR（MS + Mr. TYDI）	0.358	**0.658**	**0.598**

表 2：MIRACL 三个数据集上的 nDCG@10。LRL 对第 (3) 行混合系统的前 20 个结果进行重新排序。在有监督设置下，mDPR (MS) 在 MS MARCO 上训练，而 mDPR (MS +Mr. TYDI) 则进一步使用 Mr. TYDI (Zhang 等人，2022a年）。

5 结果

表1 展示了三个 TREC DL 数据集上的结果，其中第（1–9）行比较了不同的零样本排序流程，第（a–d）行报告了使用在 MS MARCO 训练集上训练的有监督方法所得的结果（Bajaj等人，2016）。当然，我们也注意到 GPT 模型中训练数据与测试数据可能存在交叉混用，因此“零样本”的说法应谨慎看待——不过，我们的工作并非唯一存在这一潜在缺陷的研究。

第（1、2）行报告了使用两种第一阶段检索器 BM25 和 Contriever 得到的基线结果。第（3–5）行展示了不同大语言模型对 BM25 检索出的排名前 100 的候选文档进行重排序的结果，其中包括当前最先进水平 UPR、我们的基线 PRL，以及我们的列表式重排序器 LRL。我们使用 BM25 作为第一阶段检索器，因为它在 DL19 和 DL20 上的效果更好。结果表明，LRL 在全部三个数据集上的效果都优于逐点式重排序器：以 nDCG@10 为例，LRL 的平均表现分别比 UPR 和 PRL 高出约 6 个百分点和 3 个百分点。

第（6–9）行展示了将 LRL 作为额外的重排序阶段，对 UPR 和 PRL 返回的排名前 10/20 篇文档进行重排序的结果。这一设置的灵感来自 duoBERT 中提出的多阶段排序方法（Nogueira 等人，2019年）。在这种情况下，我们可以避免对过长的列表进行渐进式重排序，从而提高效率。结果表明，与逐点重排序器相比，我们的方法能够持续带来进一步提升。例如，如第（9）行所示，对 PRL 返回的排名前 20 篇文档进行重排序后，LRL 的 nDCG@10 和 MRR@10 平均分别提升了五分和七分。这进一步支持了我们的观点：基于文档列表进行重排序，确实比单独计算每篇文档的相关性分数能提供更好的信号。

将表现最佳的 LRL 流程与有监督排序流程进行比较后，我们很高兴地发现，LRL 无需任何任务特定训练，其效果甚至优于部分有监督稠密检索器，例如 DPR。DPR 使用大量领域内训练数据进行训练，如第 (a) 行所示。不过，与更复杂的有监督排序流程相比，效果差距依然存在，例如 TCT-ColBERT（Lin 等人，2021年）。

由于 GPT-3 具备多语言能力，我们提出的方法可以直接应用于非英语数据集。表2 展示了 MIRACL 中三个数据集上的结果。在第 (3) 行中，我们对第一阶段检索器检索结果前20名进行重排序；该检索器是 BM25 与 mContriever 的混合方法，结果见第 (1–2) 行。在这三个 MIRACL 数据集上，LRL 的表现优于混合方法基线，证明了我们的 LRL 重排序器在多语言检索中的有效性。

6 结论

在这项工作中，我们提出了 LRL，这是一种采用大语言模型的列表式零样本重排序器，它通过一种简单、新颖且有效的方法进行重排序。我们展示了它在网页搜索数据集上的表现优于最先进水平的零样本逐点重排序器，并证明了其在不同语言中的有效性。这项工作才刚刚触及大语言模型在文本检索领域巨大潜力的表层，我们对未来的发展前景充满期待。

致谢

本研究部分得到了加拿大自然科学与工程研究委员会（NSERC）的资助。

参考文献

Payal Bajaj、Daniel Campos、Nick Craswell、Li Deng、Jianfeng Gao、Xiaodong Liu、Rangan Majumder、Andrew McNamara、Bhaskar Mitra、Tri Nguyen、Mir Rosenberg、Xia Song、Alina Stoica、Saurabh Tiwary 和 Tong Wang。2016。MS MARCO：一个由人工生成的机器阅读理解数据集。*arXiv:arXiv:1611.09268v3*。

Luiz Bonifacio、Hugo Abonizio、Marzieh Fadaee 和 Rodrigo Nogueira。2022年。InPars：用于信息检索的无监督数据集生成。载于第*45*届 *ACM SIGIR* 信息检索研究与开发国际会议论文集，SIGIR '22，第2387–2392页。

Luiz Bonifacio、Vitor Jeronymo、Hugo Queiroz Abonizio、Israel Campiotti、Marzieh Fadaee、Roberto Lotufo 和 Rodrigo Nogueira。2021年。mMARCO：MS MARCO 段落排序数据集的多语言版本。*arXiv:2108.13897*。

Leonid Boytsov、Preksha Patel、Vivek Sourabh、Riddhi Nisar、Sayani Kundu、Ramya Ramanathan 和 Eric Nyberg。2023年。InPars-Light：以经济高效的方式对高效排序器进行无监督训练。*arXiv:2301.02998*。

Tom B. Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell、Sandhini Agarwal、Ariel Herbert-Voss、Gretchen Krueger、Tom Henighan、Rewon Child、Aditya Ramesh、Daniel M. Ziegler、Jeffrey Wu、Clemens Winter、Christopher Hesse、Mark Chen、Eric Sigler、Mateusz Litwin、Scott Gray、Benjamin Chess、Jack Clark、Christopher Berner、Sam McCandlish、Alec Radford、Ilya Sutskever 和 Dario Amodei。2020年。语言模型是少样本学习器。神经信息处理系统进展，33:1877–1901。

Danqi Chen、Adam Fisch、Jason Weston 和 Antoine Bordes。2017。《阅读维基百科以回答开放领域问题》。载于第*55*届计算语言学协会年会论文集（第*1*卷：长篇论文），第1870–1879页，加拿大温哥华。

Nick Craswell、Bhaskar Mitra、Emine Yilmaz 和 Daniel Campos。2021年。TREC 2020年深度学习赛道概述。*arXiv:2102.07662*。

Nick Craswell、BhaskarMitra、Emine Yilmaz、Daniel Campos 和 Jimmy Lin。2022年。TREC 2021 深度学习赛道概述。载于第三十届文本检索会议（*TREC 2021*）论文集。

Nick Craswell、Bhaskar Mitra、Emine Yilmaz、Daniel Campos 和 Ellen M. Voorhees。2020年。TREC 2019年深度学习赛道概述。*arXiv:2003.07820*。

Zhuyun Dai、Vincent Zhao、Ji Ma、Yi Luan、Jianmo Ni、Jing Lu、AntonBakalov、Kelvin Guu、Keith B. Hall 和 Ming-Wei Chang。2022年。Promptaga-tor：基于 8 个示例的少样本稠密检索。*arXiv:2209.11755*。

Luyu Gao、Zhuyun Dai 和 Jamie Callan。2021年。重新思考多阶段检索流程中 BERT 重排序器的训练。在第 *43* 届欧洲信息检索会议（*ECIR*）。

Luyu Gao、Xueguang Ma、Jimmy Lin 和 Jamie Callan。2022年。无需相关性标签的精确零样本稠密检索。*arXiv:2212.10496*。

GautierIzacard、MathildeCaron、Lucas Hosseini、Se- bastian Riedel、Piotr Bojanowski、Armand Joulin 和 Edouard Grave。2021年。基于对比学习的无监督稠密信息检索。*arXiv:2112.09118*。

Sheng-Chieh Lin、Jheng-Hong Yang 和 Jimmy Lin。2021年。用于稠密检索的紧密耦合教师知识蒸馏中的批内负例。载于第六届自然语言处理表示学习研讨会论文集（*RepL4NLP-2021*），第 163–173 页。

Rodrigo Nogueira 和 Kyunghyun Cho。2019年。使用 BERT 进行段落重排序。*arXiv:1901.04085*。

Rodrigo Nogueira、Zhiying Jiang、Ronak Pradeep 和 Jimmy Lin。2020年。使用预训练序列到序列模型进行文档排序。载于计算语言学协会论文集：*EMNLP 2020*，第708–718页。

Rodrigo Nogueira、Wei Yang、Kyunghyun Cho 和 Jimmy Lin。2019年。基于 BERT 的多阶段文档排序。*arXiv:1910.14424*。

Long Ouyang、Jeff Wu、Xu Jiang、Diogo Almeida、Carroll L. Wainwright、Pamela Mishkin、Chong Zhang、Sandhini Agarwal、Katarina Slama、Alex Ray、John Schulman、Jacob Hilton、Fraser Kelton、Luke E. Miller、Maddie Simens、Amanda Askell、PeterWelinder、Paul FrancisChristiano、Jan Leike 和 Ryan J. Lowe。2022年。训练语言模型遵循带有人类反馈的指令。*arXiv:2203.02155*。

Fabio Petroni、Aleksandra Piktus、Angela Fan、Patrick Lewis、Majid Yazdani、Nicola De Cao、James Thorne、Yacine Jernite、Vladimir Karpukhin、Jean Maillard、Vassilis Plachouras、Tim Rocktäschel 和 Sebastian Riedel。2021年。KILT：知识密集型语言任务的基准测试。载于*2021*年计算语言学协会北美分会会议论文集：人类语言技术，第2523–2544页。

Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、Yanqi Zhou、Wei Li 和 Peter J. Liu。2020年。探索统一文本到文本转换器的迁移学习极限。机器学习研究期刊，21(140):1–67。

Stephen E. Robertson 和 Hugo Zaragoza。2009年。概率相关性框架：BM25 及其扩展。信息检索基础与趋势，3(4):333–389。

Devendra Sachan、Mike Lewis、Mandar Joshi、Armen Aghajanyan、Wen-tau Yih、Joelle Pineau 和 Luke Zettlemoyer。2022年。通过零样本问题生成改进段落检索。载于*2022*年自然语言处理实证方法会议论文集，第 3781–3797 页，阿布扎比，阿拉伯联合酋长国。

Victor Sanh、Albert Webson、Colin Raffel、Stephen Bach、Lintang Sutawika、Zaid Alyafeai、Antoine Chaffin、Arnaud Stiegler、Arun Raja、Manan Dey、M Saiful Bari、Canwen Xu、Urmish Thakker、Shanya Sharma Sharma、Eliza Szczechla、Taewoon Kim、Gunjan Chhablani、Nihal V. Nayak、Debajyoti Datta、Jonathan Chang、Mike Tian-Jian Jiang、Han Wang、Matteo Manica、Sheng Shen、Zheng Xin Yong、Harshit Pandey、Rachel Bawden、Thomas Wang、Trishala Neeraj、Jos Rozen、Abheesht Sharma、Andrea Santilli、Thibault Févry、Jason Alan Fries、Ryan Teehan、Teven Le Scao、Stella Biderman、Leo Gao、Thomas Wolf 和 Alexander M. Rush。2022年。多任务提示训练实现零样本任务泛化。发表于第十届国际学习表征会议.

Nandan Thakur、Nils Reimers、Andreas Rücklé、Ab- hishek Srivastava 和 Iryna Gurevych。2021年。BEIR：用于零样本评估信息检索模型的异构基准。载于第三十五届神经信息处理系统大会数据集与基准赛道（第二轮）。

Jason Wei、Maarten Bosma、Vincent Y. Zhao、Kelvin Guu、Adams Wei Yu、Brian Lester、Nan Du、An- drew M. Dai 和 Quoc V. Le。2022年。微调后的语言模型是零样本学习器。载于第十届国际学习表征会议。

Xinyu Zhang、Kelechi Ogueji、Xueguang Ma 和 Jimmy Lin。2022a年。训练多语言稠密检索模型的最佳实践探索。*arXiv:2204.02363*。

Xinyu Zhang、Nandan Thakur、Odunayo Ogundepo、Ehsan Kamalloo、David Alfonso-Hermelo、Xi- aoguang Li、Qun Liu、Mehdi Rezagholizadeh 和 Jimmy Lin。2022b年。打造 MIRACL：跨越连续语言谱系的多语言信息检索。*arXiv:2210.09984*。

Honglei Zhuang、Zhen Qin、Rolf Jagerman、Kai Hui、Ji Ma、Jing Lu、Jianmo Ni、Xuanhui Wang 和 Michael Bendersky。2022年。RankT5：使用排序损失微调 T5 以进行文本排序。*arXiv:2210.10634*。