arXiv:2602.01116v1 [cs.CL] 1 Feb 2026

逻辑导向 通过对比学习增强检索器

Wenxuan Zhang[1], *Yuan-Hao Jiang*[1],*Changyong Qi*[1],*Rui Jia*[1],*Yonghe Wu*[2]

[1]上海人工智能教育研究院、华东师范大学，中国[2]教育技术，华东师范大学，中国

摘要

大语言模型（LLM）在知识密集型任务中表现吃力，因为检索器往往过度拟合表层相似度，难以处理涉及复杂逻辑关系的查询。逻辑分析能力内在于模型表征之中，但在标准训练中未得到充分利用。LORE（逻辑导向检索器增强）引入细粒度对比学习，以激活这种潜在能力，引导嵌入与逻辑结构所对应的证据对齐，而非仅关注浅层相似度。LORE 无需外部监督、资源或检索前分析，兼容索引，并能持续提升检索效用和下游生成效果，同时保持高效。数据集和代码已公开发布于 https://github.com/mazehart/Lore-RAG。

索引词—大语言模型（大语言模型），检索增强生成（检索增强生成），对比学习

1. 引言

大语言模型（LLM）在广泛的自然语言处理任务中取得了显著进展 [1, 2], ，但在知识密集型场景中仍存在局限，常出现事实性幻觉、知识陈旧和可追溯性较差等问题。这些局限降低了可靠性，使得大语言模型难以应用于对正确性和可验证性要求极高的高风险应用[3, 4]。因此，检索增强生成（RAG）已成为一种实用的解决方案 [5]。RAG 遵循“先检索，再生成”的范式，通过嵌入模型进行向量匹配，从外部知识源中检索信息，再将其传递给大模型，从而缓解幻觉问题，并增强可解释性和来源可追溯性 [6, 7]。

在实际应用中，始终存在一个明显的差距：嵌入模型在处理直接、简单的查询时表现良好，但在处理受到复杂逻辑表达式干扰的查询时，其效果会显著下降。如图所示，现有的开源嵌入模型即使参数量很大，也无法避免受到复杂逻辑表达式干扰的查询带来的困难。这

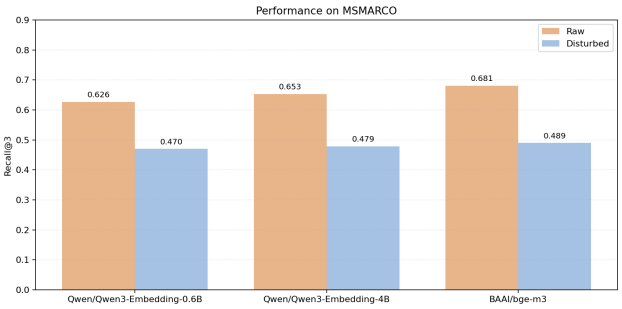

图 1。 复杂逻辑表达式对嵌入模型性能的影响。 MSMARCO 数据集上的结果显示，当查询中加入复杂逻辑表达式进行扰动时，不同嵌入模型的性能都会显著下降，这凸显了当前嵌入方法面对复杂逻辑结构时的脆弱性。

是因为它们由相似度驱动的特性使其往往更倾向于关注表层的词汇重叠，而非忠实地表示并满足查询的复杂逻辑要求，进而形成脆弱的证据集，无法满足查询要求 [8]。

一种自然的做法是在检索之前，使用强大的大语言模型分析或分解查询 [9, 10]。然而，这种“大语言模型在环”策略会带来不可忽视的成本和延迟，而且依赖于为数据集构建的知识图谱及其他结构。本文采取了不同的思路：我们不再增加大语言模型的计算量，而是提出一个问题——现有开源嵌入模型是否已经具备尚未充分发挥的能力，能够表示复杂的语言结构？针对性的对比学习能否激发这种能力，使检索器直接编码复杂逻辑结构和显著性，而不是仅仅依赖表层相似度？

本文提出了 LORE，一种通过对比学习增强嵌入的方法，用于改进检索器。LORE 基于现有数据集，在查询之后、检索之前不调用大模型来辅助嵌入，而是在预训练之后，利用大模型标注的细粒度对比知识重新训练嵌入模型，强调与复杂逻辑表达

通讯作者：yhwu@deit.ecnu.edu.cn

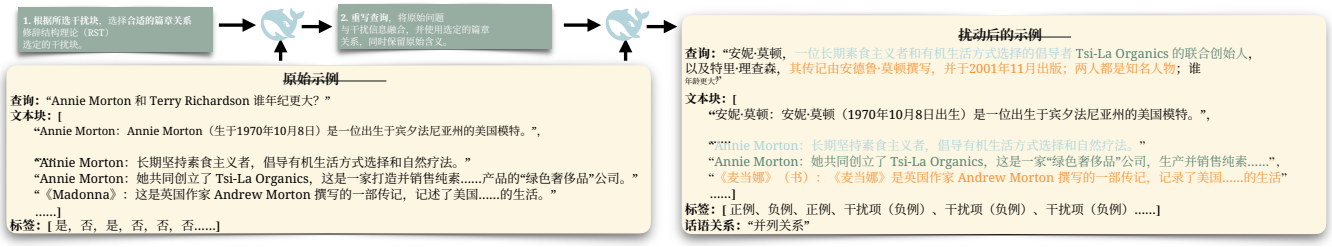

图 2。查询重写示意图。左侧展示了包含查询、文本块和标签的原始示例，右侧则展示了对应的修改后示例，其中包含经过扩充的查询、篇章关系和干扰信息标注。

相关的信号。

本文的贡献主要有三点：

- 本文提出 LORE，这是一种嵌入模型增强范式，在后预训练阶段融入大模型知识，并提供一种简单的对比微调方法。
- 本文基于现有开源资源构建了一个开源的细粒度对比学习数据集。
- 本文验证了 LORE 在多个包含复杂逻辑表达式干扰查询的任务上均能带来稳定提升；在无标签任务上的改进尤其证明了所提方法的有效性。

2. 相关工作

超越相似度。 近期方法开始着手解决检索增强生成依赖相似度所带来的局限。ME-TRAG 尝试将大语言模型在下游任务上的答案准确率变化作为监督信号，对检索嵌入模型进行微调，使其能够识别有助于提高大语言模型下游任务答案准确率的检索内容 [10]。与此同时，Mindful-RAG 将大语言模型的内部知识与外部知识图谱相结合：首先通过大语言模型分析查询意图，然后整合知识图谱，在检索过程中引入上下文对齐和结果验证，从而减少复杂推理中的错误和歧义 [9]。

局限性。 这些方法通过任务反馈、意图建模或外部知识结构提升效用，但往往会带来额外开销，并依赖外部监督或资源，从而阻碍其在不同任务间的迁移。尽管这些方法利用大模型知识来提升嵌入模型的性能，但这一过程并未增强嵌入模型的基本能力。因此，一个关键问题是：如何让检索器理解包含复杂逻辑表达式的查询，使其能够直接编码查询，并优先检索符合查询逻辑结构的证据，同时保持高效，并避免依赖检索前大语言模型分析、外部监督和外部知识结构 [8]。

定位。 本文提出 LORE，这是一种对比式后训练方法，旨在激活开源嵌入模型表示具有复杂逻辑表达式的查询时的潜在能力，使检索器能够直接编码逻辑结构，而非依赖表层相似度，从而从嵌入的基础能力层面提升检索性能。

3. 方法

3.1. 数据集构建

细粒度文本块类别 构建过程从一个基础语料库开始，其中包含查询 q、候选文本块 $\{c_k\}$ 和布尔标签 $y_k \in \{$True，False$\}$。对于每个 q，抽取一部分标记为 False 的文本块作为干扰项，为每个选中的干扰项分配其原始文本块索引作为源标识符，并提示大语言模型将 q 改写为更自然的查询，在保留原始含义的同时融入部分干扰项内容。随后将干扰项的使用情况映射回文本块索引，并为每个 q 和候选集 $\mathcal{C}$构建三个层级（q）：

- 正例 *(P)*：带有 y_k = True 标签、足以回答查询的文本块。

- 干扰项 *(N1)*：大语言模型在改写后的问题中用作干扰项的假标签文本块。这些文本块看似与问题相关，却无法帮助回答问题。

-负例 *(N2)*：改写时未使用的其他假标签文本块。

这三个层级分别记为 P、N1 和 N2。当仅改写数据集的一部分时，其余样本保留原始查询和布尔标签，并映射到这些层级以保持一致。

查询重写数据集构建遵循如图2所示的两步流程。首先，根据所选文本块，从修辞结构理论（RST） [11] 中选取适当的篇章关系。八种常见篇章关系——顺序关系、过渡关系、补充关系、对比关系、因果关系、并列关系、假设关系和解释关系——作为

准则。其次，利用所选篇章关系，将原始问题与干扰信息融合以重写查询，从而得到带有丰富查询和篇章关系标注的修改示例。构建的训练数据集CoEnTrain已发布。

3.2. 对比学习

对比损失 该框架采用两个独立的编码器：查询编码器 $\mathcal{M}_q$ 和文档编码器$\mathcal{M}_d$。查询和文本块分别通过对应的编码器嵌入到共享向量空间中。令$\mathbf{h}_q = \mathcal{M}_q(q)$和$\mathbf{h}_k = \mathcal{M}_d(c_k)$表示归一化后的嵌入， s_k 表示查询与第 k个候选文本块之间的余弦相似度：

$$\mathbf{h}_q,\, \mathbf{h}_k \in \mathbb{R}^d,\; s_k = \cos(\mathbf{h}_q, \mathbf{h}_k).$$

对于每个（q， $\mathcal{C}(q)$），在logit空间中应用温度缩放和负样本分层加权。令 $\tilde{s}_k$ 表示调整后的相似度：

$$\tilde{s}_k = \begin{cases} s_k/\tau, & k \in \mathrm{P} \\ s_k/\tau + \log(\beta), & k \in \mathrm{N1} \\ s_k/\tau + \log(\alpha), & k \in \mathrm{N2} \end{cases} \quad (\beta > \alpha > 0).$$

差异化加权 $\beta > \alpha$ 对于处理复杂的逻辑表达式至关重要，因为它确保干扰项（N1）比负样本（N2）受到更强的惩罚，从而促使模型学习细粒度的区分能力。正样本 $k \in$ P的概率仅使用负样本（N1和N2）作为分母：

$$p_k = \frac{\exp(\tilde{s}_k)}{\sum_{t \in N1 \cup N2} \exp(\tilde{s}_t) + \exp(\tilde{s}_k)}, \quad k \in \mathrm{P}.$$

遵循InfoNCE框架 [12], ，对查询 q 进行最小化的目标为

$$\mathcal{L}(q) = -\frac{1}{|P|} \sum_{k \in P} \log p_k.$$

该设计在 logit空间中扩展干扰项/负例样本（含 $\beta > \alpha$），将正例排除在分母之外以避免稀释，并强调层次分离 P $\succ$ N1 $\succ$ N2，以提升对复杂逻辑表达式的理解。

训练过程 文档编码器 $\mathcal{M}_d$ 使用预训练参数并保持冻结，而查询编码器 $\mathcal{M}_q$ 在训练期间进行微调。基于上述定义的嵌入和相似度，对比损失作为优化目标。形式上，通过以下方式优化查询编码器参数θ_q：

$$\theta_q^* = \arg\min_{\theta_q} \; \mathbb{E}_{q \sim \mathcal{D}}\, \mathcal{L}(q),$$

其中 $\mathcal{D}$ 表示训练集。

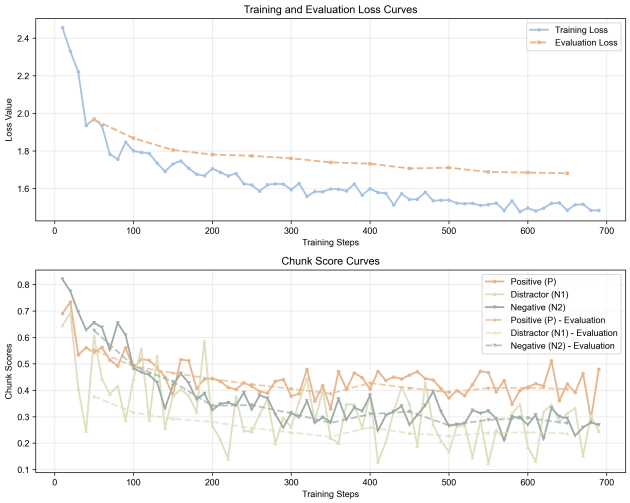

图3。Qwen3-Embedding-0.6B 微调期间的训练与验证损失曲线（上）以及不同文本块类别的文本块得分变化（下）。

4. 实验

4.1. 实验设置

数据集 实验在三个广泛使用的公开数据集上进行：HotpotQA [13], MS MARCO [14], 和 MuSiQue [15]。对于训练集和验证集，在 MS MARCO 数据上进行查询重写，并标注细粒度类别对比关系信息，而 HotpotQA 和 MuSiQue 数据集则不进行标注。对于测试集，所有数据集均采用查询重写，同时保留原始测试集以供比较。重写后的测试集经过人工筛选，以确保符合重写要求。

模型与基线 实验使用两种嵌入模型：Qwen3 Embedding [16] 和 BGE M3-Embedding [17]。在原始测试集和受扰测试集上，将 LORE 的有效性与原始模型及传统对比学习方法进行比较。

评估指标检索性能主要通过前k 位的正例召回率（@3、@5、@10）进行评估，该指标衡量相关文本块在前k 个结果中被成功检索到的比例。对于查询改写测试集，还会记录干扰样本的召回率。

实验设置 模型训练在单张 A800 GPU 上进行。学习率设为 1×10^{-5}，温度参数 $\tau = 0.05$，层级加权参数 $\alpha = 1.0$, $\beta = 3.0$。训练进行 1 个 epoch，批量大小为 32。为确保结果可复现，使用不同的随机种子（0、1、2）训练三个模型，并报告各次运行结果的均值和标准差。

表 1。正向块和干扰块的召回率。

召回率		HotpotQA			MuSiQue			MSMARCO		
		原始数据集	扰动数据集		原始数据集	扰动数据集		原始数据集	扰动数据集	
		P↑	P↑	N1↓	P↑	P↑	N1↓	P↑	P↑	N1↓
Qwen/Qwen3-Embedding-0.6B										
@3	**原始模型**	56.37±0.00	33.76±0.00	70.03±0.00	59.07±0.00	39.55±0.00	83.88±0.00	62.63±0.00	46.98±0.00	84.25±0.00
	+InfoNCE	**70.87**±0.40	59.71±0.56	50.51±1.06	**67.23**±0.42	53.65±0.47	70.36±1.16	**68.39**±0.25	51.63±0.40	82.03±0.56
	+LORE	70.72±0.09	**65.45**±0.54	**23.81**±1.39	66.68±0.52	**59.28**±0.34	**36.87**±2.06	68.35±0.23	**69.30**±0.35	**20.15**±0.87
@5	**原始模型**	68.04±0.00	49.23±0.00	81.59±0.00	70.13±0.00	56.38±0.00	88.80±0.00	82.41±0.00	73.25±0.00	93.34±0.00
	+InfoNCE	**81.96**±0.20	74.77±0.68	69.65±1.13	**79.93**±0.34	70.03±0.23	81.25±0.85	84.96±0.03	76.68±0.24	92.71±0.05
	+LORE	81.84±0.44	**77.79**±0.34	**39.73**±2.09	79.41±0.30	**73.32**±0.05	**52.81**±1.76	**85.06**±0.31	**85.77**±0.22	**44.22**±0.78
@10	**原始模型**	80.55±0.00	69.13±0.00	90.96±0.00	84.28±0.00	77.46±0.00	92.09±0.00	96.75±0.00	96.70±0.00	99.45±0.00
	+InfoNCE	**92.61**±0.08	89.30±0.21	86.95±0.64	**92.81**±0.23	88.75±0.17	89.97±0.29	96.75±0.00	96.70±0.00	99.45±0.00
	+LORE	92.45±0.02	**89.62**±0.32	**62.84**±2.28	92.37±0.16	**88.87**±0.06	**74.66**±0.96	96.75±0.00	96.70±0.00	99.45±0.00
BAAI/bge-m3										
@3	**原始模型**	64.68±0.00	40.04±0.00	71.84±0.00	60.12±0.00	42.33±0.00	84.07±0.00	68.10±0.00	48.95±0.00	84.02±0.00
	+InfoNCE	**71.53**±0.17	57.77±0.62	53.96±0.89	**66.51**±0.06	52.94±0.09	73.48±0.36	70.21±0.27	52.49±0.29	80.93±0.21
	+LORE	70.76±0.22	**65.57**±0.58	**20.43**±1.38	65.88±0.08	**59.94**±0.53	**25.80**±2.78	**70.52**±0.23	**69.28**±0.08	**22.80**±0.75
@5	**原始模型**	76.09±0.00	58.24±0.00	82.89±0.00	71.96±0.00	60.28±0.00	88.23±0.00	85.25±0.00	75.20±0.00	93.51±0.00
	+InfoNCE	**83.06**±0.23	73.74±0.11	72.22±0.57	**77.85**±0.05	69.19±0.16	84.02±0.20	85.98±0.18	78.17±0.20	92.31±0.20
	+LORE	82.52±0.11	**77.59**±0.38	**33.05**±2.09	77.28±0.23	**72.29**±0.06	**40.00**±3.06	**86.20**±0.21	**86.10**±0.09	**44.89**±0.60
@10	**原始模型**	88.28±0.00	78.86±0.00	92.47±0.00	86.58±0.00	81.27±0.00	91.69±0.00	96.75±0.00	96.70±0.00	99.45±0.00
	+InfoNCE	**93.26**±0.03	89.14±0.12	87.55±0.21	**91.69**±0.05	87.78±0.07	90.90±0.13	96.75±0.00	96.70±0.00	99.45±0.00
	+LORE	93.10±0.08	**89.49**±0.61	**54.90**±2.96	91.37±0.08	**88.36**±0.09	**61.53**±2.59	96.75±0.00	96.70±0.00	99.45±0.00

4.2. 结果与分析

训练动态 图 3 展示了 LORE 的学习过程。初期，尽管干扰样本（N1）与回答查询无关，模型仍因表层词汇重叠而更偏好干扰样本（N1），而非正例（P）。

在训练过程中，干扰项（N1）样本的分数持续下降，最终在较低水平与负样本（N2）趋于一致；而正例（P）样本的分数则相对稳定，并占据明显优势。这种逐步分离验证了分层加权对比损失设计的有效性。

检索性能如表 1 所示，InfoNCE 和 LORE 在原始数据集和扰动数据集上的表现均显著优于原始模型。由于实验中仅对查询编码器 $\mathcal{M}_q$ 进行了微调，这表明对比学习能够增强模型对查询的理解，从而提升检索性能。然而，与 InfoNCE 相比，利用大语言模型细粒度标注的 LORE 带来了更显著的提升。

泛化能力训练集中只有 MSMARCO 获得了细粒度标注，但 HotpotQA 和 MuSiQue 数据集也同样受益于这种方法。这表明 LORE 方法具有跨任务泛化效应。构建足够通用且丰富的细粒度数据集后，预计嵌入模型在各种任务上的性能都将得到更广泛的提升。

这种可迁移性体现了 LORE 在实际应用中的价值，因为在实际应用中，对所有目标领域进行全面标注往往并不可行。

5. 局限性与结论

现有嵌入模型受训练语料和预训练范式的限制，对包含复杂逻辑表达式的查询理解不足，并且过度依赖表层相似性。以往的工作主要着眼于在特定场景中使用大模型提升性能，而非增强嵌入模型的基础能力。为此，本文基于修辞结构理论（RST）构造了包含复杂逻辑表达式的扰动查询，并针对三类文本块应用了细粒度对比标注。对比学习使嵌入模型能够更好地捕捉复杂逻辑表达式，同时与现有方法完全兼容，从而减少对额外特定任务设计的需求。

尽管已证明该方法有效，但其应用受到逻辑结构和数据集多样性的限制，其泛化能力仍有待进一步研究。未来的方向包括：针对更贴近现实且包含复杂逻辑表达式的查询，开发基于智能体的数据标注系统；通过对比学习，将大模型对查询意图的理解迁移至较小的嵌入模型，从而增强其基础能力。

6. 致谢

本研究得到国家社会科学基金重大项目（项目编号：21&ZD238）资助。

7. 参考文献

[1] Tom B. Brown、Benjamin Mann、Nick Ryder 等人，“语言模型是少样本学习器”，载于 神经信息处理系统进展第*33*卷：*2020*年神经信息处理系统年度会议，*NeurIPS 2020*，*2020*年*12*月*6*日至*12*日，线上，Hugo Larochelle、Marc’Aurelio Ranzato、Raia Hadsell、Maria-Florina Balcan 和 Hsuan-Tien Lin 编，2020年。

[2] Muhammad Arslan、Hussam Ghanem、Saba Munawar 和 Christophe Cruz，“基于大语言模型的检索增强生成综述”，*Procedia*计算机科学，第246卷，第3781–3790页，2024。

[3] Lei Huang、Weijiang Yu、Weitao Ma、Weihong Zhong、Zhangyin Feng、Haotian Wang、Qianglong Chen、Weihua Peng、Xiaocheng Feng、Bing Qin 等人，“大语言模型幻觉综述：原理、分类体系、挑战与开放问题”，*ACM*信息系统汇刊，第43卷，第2期，第1–55页，2025年。

[4] Junyi Li、Xiaoxue Cheng、Xin Zhao、Jian-Yun Nie 和 Ji-Rong Wen，《Halueval：面向大语言模型的大规模幻觉评测基准》，载于《*2023* 年自然语言处理实证方法会议论文集（*EMNLP 2023*）》，*2023* 年 *12* 月 *6—10* 日，新加坡，Houda Bouamor、Juan Pino 和 Kalika Bali 编，2023 年，第 6449–6464 页，计算语言学协会。

[5] Patrick Lewis、Ethan Perez、Aleksandra Piktus、Fabio Petroni、Vladimir Karpukhin、Naman Goyal、Heinrich K¨uttler、Mike Lewis、Wen-tau Yih、Tim Rockt¨aschel 等人，《面向知识密集型 NLP 任务的检索增强生成》，神经信息处理系统进展，第 33 卷，第 9459–9474 页，2020 年。

[6]Jinhao Jiang、Kun Zhou、Zican Dong、Keming Ye、Wayne Xin Zhao和Ji-Rong Wen，“Structgpt：一个让大型语言模特对结构化数据进行推理的通用框架”，*arXiv*预印本 *arXiv:2305.09645*，2023年。

[7]Parth Sarthi、Salman Abdullah、Aditi Tuli、Shubh Khanna、Anna Goldie 和 Christopher D Manning，《Raptor：用于树状组织检索的递归式抽象处理》，载于第十二届国际学习表征会议，2024。

[8] Laura Caspari、Kanishka Ghosh Dastidar、Saber Zerhoudi、Jelena Mitrovic 和 Michael Granitzer，《超越基准：评估检索增强生成系统的嵌入模型相似度》，*arXiv* 预印本 *arXiv:2407.08275*，2024。

[9]Garima Agrawal、Tharindu Kumarage、Zeyad Alghamdi 和 Huan Liu，《Mindful-rag：检索增强生成中的失效点研究》，载于*2024*年第二届基础模型与大语言模型国际会议（*FLLM*）。IEEE，2024，第607–611页。

[10] Chunjing Gan、Dan Yang、Binbin Hu、Hanxiao Zhang、Siyuan Li、Ziqi Liu、Yue Shen、Lin Ju、Zhiqiang Zhang、Jinjie Gu等，“相似性并非你所需要的全部：为检索增强生成赋予多层次思维”，*arXiv*预印本 *arXiv:2405.19893*，2024.

[11] William C Mann和Sandra A Thompson，“修辞结构理论：迈向文本组织的功能理论”，《*Text*：话语研究跨学科期刊》，第8卷，第3期，第243–281页，1988年。

[12] Aaron van den Oord、Yazhe Li 和 Oriol Vinyals，《基于对比预测编码的表示学习》，*arXiv* 预印本 *arXiv:1807.03748*，2018年。

[13] Zhilin Yang、Peng Qi、Saizheng Zhang、Yoshua Bengio、William W. Cohen、Ruslan Salakhutdinov 和 Christopher D. Manning，《 HotpotQA：一个用于多样化、可解释多跳问答的数据集》，载于自然语言处理实证方法会议（*EMNLP*），2018年。

[14] Tri Nguyen、Mir Rosenberg、Xia Song、Jianfeng Gao、Saurabh Tiwary、Rangan Majumder 和 Li Deng，《MS MARCO：一个由人工生成的机器阅读理解数据集》，*CoRR*，第 abs/1611.09268 卷，2016年。

[15] Harsh Trivedi、Niranjan Balasubramanian、Tushar Khot 和 Ashish Sabharwal，《MuSiQue：通过单跳问题组合构建多跳问题》，《计算语言学协会汇刊》，第10卷，第539–554页，2022年。

[16]Yanzhao Zhang、Mingxin Li、Dingkun Long、Xin Zhang、Huan Lin、Baosong Yang、Pengjun Xie、An Yang、Dayiheng Liu、Junyang Lin、Fei Huang 和 Jingren Zhou，《Qwen3 embedding：通过基础模型推进文本嵌入与重排序》，*arXiv* 预印本 *arXiv:2506.05176*，2025年。

[17] JianlvChen、Shitao Xiao、PeitianZhang、KunLuo、Defu Lian 和 Zheng Liu，《BGE M3-Embedding：通过自知识蒸馏实现多语言、多功能、多粒度文本嵌入》，2024。