arXiv:2602.01239v1 [cs.CL] 1 Feb 2026

推理式问答

Jamshid Mozafari*因斯布鲁克大学 因斯布鲁克，奥地利
ja
mshid.mozafari@uibk.ac.at

Hamed Zamani 马萨诸塞大学阿默斯特分校 阿默斯特，美国
za
mani@cs.umass.edu

Guido Zuccon 昆士兰大学 布里斯班，澳大利亚
g.zuccon
@uq.edu.au

Adam Jatowt 因斯布鲁克大学 因斯布鲁克，奥地利
adam.jatowt@uibk.ac.at

摘要

尽管针对各种问答（QA）系统的研究十分广泛，但现有工作大多聚焦于答案包含性，即假设答案可以直接从语料库中的文档提取和/或生成。然而，有些问题需要推理，即根据现有信息推导出文档中未明确陈述、但可以推知的答案。我们提出**推理式问答**——一项新任务，要求模型从支持答案的 段落中推断答案，而这些段落只提供线索。为研究这一问题，我们构建了QUIT（QUestions requiring Inference from Texts，要求从文本中进行推理的问题）数据集，包含7,401个问题和240万个段落，基于人工和机器撰写、收敛度高的提示构建，并通过基于大语言模型的可回答性和人工验证，标注为三个相关性等级。通过全面评估检索器、重排序器和基于大语言模型的阅读器，我们发现，在传统问答任务中表现出色的方法在推理式问答中难以奏效：检索器表现不佳，重排序器带来的提升有限，微调所带来的改进也不稳定。即使是面向推理的大语言模型，表现也不如规模更小的通用模型。这些发现表明，当前的问答流程尚未做好基于推理进行推断的准备。因此，推理式问答确立了一类新的问答任务，推动系统从间接文本证据中理解并进行推理。

CCS 概念

• **信息系统** → **检索模型与排序**；**检索任务与进球**；**检索结果评估**。

关键词

推理式问答、提示、检索增强推理、推理、检索增强生成、大型语言模型、评估

ACM 参考格式：
Jamshid Mozafari、Hamed Zamani、Guido Zuccon 和 Adam Jatowt。2026。推理式问答。载于《*2026*年*ACM Web*会议论文集》（*WWW '26*），*2026*年*4*月*13*日至*17*日，阿拉伯联合酋长国迪拜。ACM，纽约州纽约市，美国，12页。https://doi.org/10.1145/3774904.3792653

*通讯作者。

本作品采用知识共享署名 4.0 国际许可协议授权。*WWW '26*，阿拉伯联合酋长国迪拜© 2026 版权所有归所有者/作者所有。ACM ISBN 979-8-4007-2307-0/2026/04 https://doi.org/10.1145/3774904.3792653

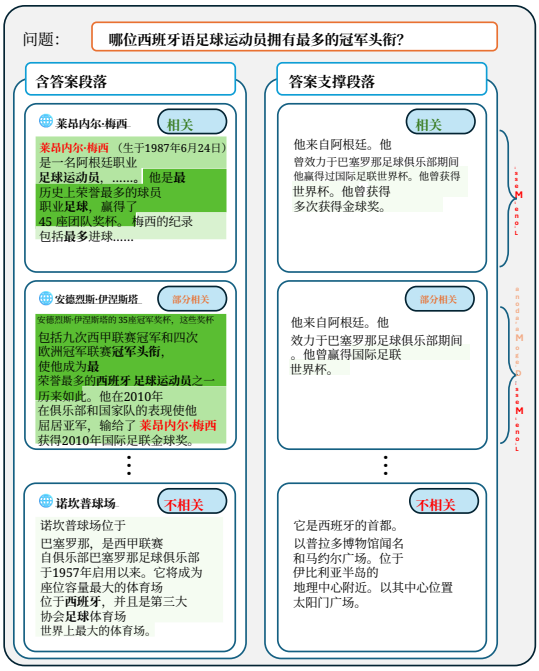

图 1：问题的含答案段落和答案支撑段落示例：哪位西班牙籍足球运动员获得的冠军头衔最多？，**答案是莱昂内尔·梅西。绿色高亮表示问题与句子之间的语义相似度（绿色越深，= 相似度越高）。粗体词语表示问题与段落之间的词汇重合。蓝色圆角矩形表示各段落与问题的相关性级别。在答案支撑列中，每个段落根据问题隐含描述的实体显示在段落右侧。**

1 引言

问答（QA）系统 [2] 旨在直接回答自然语言问题。现有的问答任务包括但不限于事实型问答 [54], 描述型（长篇）和非事实型问答 [18, 27], 布尔问答 [60], 以及多跳问答 [59]。数十年来，人们一直在研究问答系统 [38]，无论专注于上述哪类任务，早期问答系统主要采用抽取式方法，直接从检索到的段落中定位答案片段。较新的问答系统则更注重生成式方法，即根据段落内容生成答案，主要使用大语言

模型（大语言模型）。抽取式问答系统依赖 BERT [11] 和 RoBERTa [30]等基于编码器的模型。然而，随着基于解码器的生成模型（包括 LLaMA [13] 和 Gemma [53]等大语言模型）的出现，范式已转向检索增强生成（RAG） [28]。这一转变源于大语言模型强大的生成和推理能力，使其能够在广泛的自然语言任务中超越抽取式系统 [32]。

尽管大语言模型在许多现有问答基准上表现出色，但我们认为，在检索增强生成和推理的背景下，仍有一类相对缺乏研究、值得进一步关注的问答任务——**推理式问答（推理式问答）**。在这类任务中，问题的答案并不存在于语料库中；相反，模型必须根据上下文线索、背景知识以及文档所支持的逻辑推理来推断出正确答案。例如，如图 1 所示，对于问题“哪位西班牙语足球运动员获得的冠军头衔最多？”，包含答案的相关段落明确指出莱昂内尔·梅西是荣誉最多的球员，因此可以直接回答。相比之下，支持答案的相关段落提供了若干描述性线索——该球员来自阿根廷，曾效力于巴塞罗那足球俱乐部，赢得过国际足联世界杯，并多次获得金球奖——模型必须据此推断答案是莱昂内尔·梅西。这项任务在现实世界中有诸多应用，例如基于知识的推理系统 [55] 和教育辅导 [4], ；在这些场景中，从间接或不完整的证据中推导答案，对于评估理解能力和推理能力至关重要。此外，推理式问答还能评估大语言模型和检索增强生成模型在答案未被明确陈述、但可依据上下文证据推断出来的情境下的推理能力。至关重要的是，它不同于依赖通用世界知识的常识推理问答 [50], 。它也不同于多跳问答，因为它不需要整合多个段落和实体来提取或生成答案。

我们首先构建了**QUIT**（**QU**estions requiring **I**nference from**T**exts，意为需要从文本中进行推理的问题），[1] 一个包含专门划分的训练集、开发集和测试集的数据集，其中有 7,401 个问题和 2,405,325 个段落。Quit 中的段落源自**提示** [20, 21], ，即在不直接给出答案的情况下引导人们找到解决方案的线索。我们将与某个实体相关的多个提示串联起来，构成间接描述该实体的段落。例如，在图 1 中，“答案支持”一栏里的相关、部分相关和不相关段落，分别由关于莱昂内尔·梅西的四条提示、关于莱昂内尔·梅西 和迭戈·马拉多纳的三条提示，以及关于马德里的四条提示串联而成。先前的研究 [33] 表明，与检索或生成的段落相比，句子级提示能提升大语言模型在问答任务中的表现，因此我们在构建基准数据集时采用了这种方法。每个段落都被标注为以下三种相关性标签之一：**2（完全相关）**表示该段落能使大语言模型生成正确答案；**1（部分相关）**表示该段落提及了答案，但缺少生成正确答案所需的充分信息；**0（不相关）**表示该段落无法为答案提供支持。标注详情见第3.2.2节。

遵循采用检索器、重排序器和阅读器流程进行答案抽取或生成的问答领域研究，我们评估了多种检索器、重排序器和阅读器在推理式问答任务上的表现。实验表明，在现有问答数据集上表现有效的方法，在这一设置下却难以奏效。具体而言，当前的检索器和重排序器在推理式问答任务上的表现不佳，而微调带来的提升有限或不稳定。这些发现表明，现有问答流程无法满足推理型问题以及答案支撑段落检索的需求。总而言之，我们的主要贡献如下：

- 我们提出了**推理式问答**，这是一项全新且颇具挑战性的问答任务，要求模型根据文本线索推断答案，而非逐字提取答案，凸显了当前问答范式的不足以及现有流程的局限性。
- 我们构建了**QUIT**（**QU**estions requiring**I**nference from **T**exts，意为需要从文本中推理的问题），这是一个大规模基准，包含 7,401 个问题和 2,405,325 个段落[2]，这些段落源自提示。该数据集设有专门的训练集、开发集和测试集，并根据相关性将段落标注为三个等级。
- 我们全面评估了推理式问答中的检索器、重排序器和大语言模型阅读器。研究结果表明：(i) 在传统问答数据集上表现出色的方法，无法泛化至基于推理的任务；(ii) 对现有模型进行微调，带来的提升有限或不稳定。

我们认为，**推理式问答**为检索增强推理模型开辟了一个充满希望的方向，例如 Search-R1 [22],Rea- sonIR [47], RaDeR [10], 和 DIVER [31], 。这类模型能够从证据中推断答案，而非提取答案，从而推动问答更接近真正的推理与理解。

2 相关工作

2.1 问答数据集

问答研究的发展一直由大规模数据集推动。早期基准包括 SQuAD [42], TriviaQA [23],WebQuestions [5],Natural Questions (NQ) [26], 和 BoolQ [9] ，它们侧重于答案包含，即段落会明确给出答案。随后出现了面向推理的数据集，包括用于多跳推理的 HotpotQA [59]、用于数值推理的 DROP [12] ，以及用于隐式推理的 StrategyQA [16] 。较新的工作（例如 Rea- sonChainQA [61],）鼓励复杂推理，而 BRIGHT [48] 则强调推理感知检索。然而，这些数据集仍假设存在明确包含答案的段落。相比之下，Quit 关注答案支撑段落，其中只有间接证据可用，必须通过推理得出答案。

2.2 检索增强生成

大语言模型的兴起推动问答转向生成式范式，其中最具代表性的是检索增强生成 [28]。为提升这种设定下的推理能力，先前的研究探索了多跳检索 [41, 59], 、迭代检索与生成循环 [57], , 以及能够引导模型逐步推理的提示策略 [56]。尽管取得了这些进展，检索增强生成方法通常仍假设检索到的段落中明确包含答案。相比之下，Quit 会在段落仅提供间接证据的场景下评估问答系统，要求模型根据上下文线索推断答案。

[1] https://github.com/DataScienceUIBK/InferentialQA

[2] 除非另有说明，本文中的“段落”均指答案支持段落；含答案段落始终会明确标注。

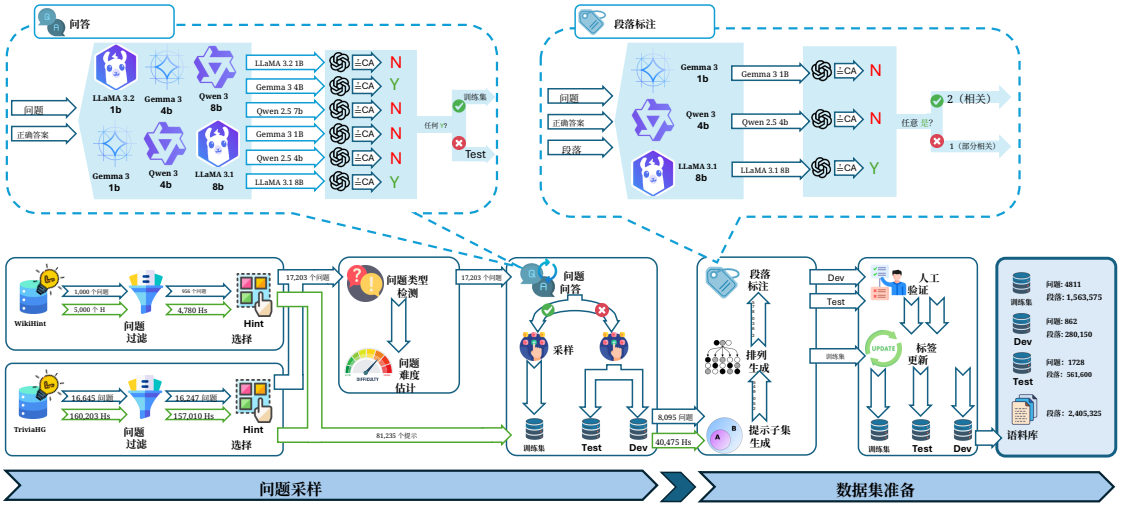

图 2：Quit数据集的构建流程。(1) 问题采样：首先筛选问题（问题）及其提示（*Hs*），以避免答案泄漏。根据收敛度为每个问题选取排名前五的提示，随后进行问题类型检测和难度估计。然后将有效问题采样到临时的训练集、开发集和测试集中。(2) 数据集准备：为每个问题生成提示的子集和排列，以形成多样的段落（*Ps*）。这些段落由系统自动标注，其中开发集和测试集的标签还会经过人工验证并更新。最终语料库包含7,401个问题和2,405,325个段落，分属训练集、开发集和测试集。

2.3 推理式问题

近期研究开始探讨如何利用提示引导大语言模型，而不直接揭示答案 [33, 36]。这类证据与推理式问题的概念相关：这类问题的答案无法从给定文本中逐字提取，而必须根据间接线索、背景知识或推理得出。除了自然语言处理领域，推理式提问长期以来也被语言学和教育学视为理解与高阶推理的核心 [17]。在自然语言处理领域，相关研究通过多跳数据集 [59] 或隐含常识 [16],探讨推理，但这些研究仍假设支持性段落中明确包含答案。据我们所知，目前还没有数据集基准测试那些仅提供间接证据的段落上的问答流程。Quit 通过拼接提示构建大规模段落，填补了这一空白，使研究者能够利用检索器、重排序器和阅读器研究推理式问答。

3 方法论

本节介绍 Quit 构建流程，包括问题采样 和数据集准备，如图 2 所示。

3.1 问题采样

在此阶段，我们识别并筛选有效问题及其对应提示，以构建 Quit 基准的基础。

3.1.1 数据集。 为构建 Quit 基准，我们使用了两个现有资源，它们提供问题—提示对：TriviaHG [35] 和 WikiHint [34]。TriviaHG 包含 16,645 个问题和 160,203 条自动生成的提示，这些提示使用 Microsoft Copilot[3], 生成， 每个问题大约对应 10 条提示。相比之下，WikiHint 包含 1,000 个问题和 5,000 条人工撰写的提示。这两个数据集有一个共同的关键特点：它们的提示旨在描述某个概念，却不直接说出其名称——这正是构建答案支撑段落所需的间接信息类型。这些提示为生成段落奠定了基础，使段落能够暗示而非直接陈述正确答案。

3.1.2 问题筛选。 我们首先过滤掉至少有一条提示泄漏答案的问题。为识别此类情况，我们使用 BEM [6] 来判断相关提示中是否有任何词语与正确答案语义等价。若至少有一个词被判定为等价，则丢弃该问题，以防止答案泄漏。接下来，对于每个剩余问题，我们根据其提示的收敛度 [35] 分数对提示进行排序，并保留前 5 条提示[4]。收敛度衡量提示在多大程度上缩小了可能答案的空间。换言之，它反映提示引导用户排除错误答案候选并聚焦正确答案的有效程度。较高的收敛度分数表明某条提示强烈指向该实体即为正确答案。这确保我们仅保留信息量最大且最准确的提示，以构建高质量段落。在此过滤与选择阶段之后，我们共获得 17,203 个问题，并配有 81,235 条高收敛度提示。

3.1.3 问题类型与难度。为了进一步丰富数据集的元数据，我们检测每个问题的类型并估算其难度。对于问题类型检测，我们使用 HintEval

[3]https://copilot.microsoft.com/

[4]为保证计算可行性，我们选取排名前 5 的提示，因为超过这一数量后，生成并评估所有子集和排列的成本会高得难以承受。

问题： <问题>答案： <标准_答案>候选答案： <候选答案>候选答案是否正确？请选择“是”或“否”

图 3：GPT-Eval 的提示。占位符 <question>表示问题， <ground_truth> 表示正确答案， <candidate> 表示由不同大语言模型生成的答案。

框架 [36], 采用 Tayyar Madabushi 和 Lee [52]提出的分类方法。为了估计问题难度，我们采用基于参考的问句复杂度方法 [14]，通过分析问题检索到的段落中有多少包含正确答案，并衡量这些段落与问题之间的语义相关性来评估难度。

3.1.4 问答。大语言模型无需依赖外部上下文 [39]，仅凭其参数化知识就能直接回答许多问题。这给我们的设置带来了挑战：如果大语言模型仅凭参数化知识就能生成正确答案，而不依赖任何外部上下文，我们便无法恰当地评估段落的影响。为解决这一问题，我们会筛除大语言模型（作为阅读器使用）以及用于段落标注的大语言模型在没有任何外部上下文时也能正确回答的问题。

为减少偏差，我们使用来自不同模型家族且参数规模各异的模型。具体而言，我们使用 Gemma 3 1B [53], Qwen 3 4B [58], 以及 LLaMA 3.1 8B [13] 进行段落标注，并使用 LLaMA 3.2 1B [13], Gemma 3 4B [53], 以及 Qwen 3 8B [58] 作为阅读器[5]。每个大语言模型仅接收问题提示，不提供任何辅助上下文。如果至少有一个模型给出正确答案，我们将该问题归类为 参数化可回答。反之，如果所有模型都未能给出正确答案，则将该问题归类为 非参数化可回答。

为将生成答案与标准答案进行比较，我们采用 GPT-Eval 方法 [24], ，因为在大语言模型时代，词汇匹配已不足够。例如，大语言模型可能输出 美利坚合众国 ，而标准答案是 美国；在这种情况下，简单的字符串匹配会失败，而 GPT-Eval 能够正确识别二者的语义等价性。该方法使用图 3 所示的提示，若生成答案与标准答案匹配则输出 是 ，否则输出 否 。为确保可靠性，我们对每个问题重复此评估三次，以确认所选大语言模型均无法通过参数回答该问题。

最后，我们随机抽取 5,000 个参数化可回答问题作为临时训练集，因为训练期间答案泄漏[6] 的问题较小。我们从非参数化可回答问题池中抽取 2,000 个问题作为临时测试集，其余问题则构成临时开发集。这样可以确保开发集和测试集在评估时保持高质量且不含偏见。

[5]也可以使用其他大语言模型；不过，我们选择这些模型来构建一个平衡且公平的评估环境。[6]答案泄漏 [36] 用于衡量提示内容在多大程度上透露了正确答案。它确保提示能够引导用户，而不会直接揭示解答。

表1：Quit数据集统计摘要

指标	数值
问题总数	7,401
段落总数	2,405,325
平均问题长度（词数）	12.56
平均段落长度（词数）	58.62
平均答案长度（词数）	1.90
查询与段落的词汇重叠	0.02
答案包含	0.00
查询-段落语义重叠	0.39

3.2 数据集准备

在此阶段，我们利用每个问题的提示生成段落，并对其进行标注，以构建最终的 Quit 基准。

3.2.1 子集与排列生成。 从前一阶段，我们在训练集、开发集和测试集中共获得 8,095 个问题和 40,475 条提示。为构建多样化的段落，我们为每个问题的五条选定提示生成所有可能的非空子集。对于每个子集，我们再枚举其成员的所有可能排列。这样，每个问题共得到 $\sum_{k=1}^{5}\binom{5}{k}k! = 325$ 个互不重复的段落。例如，包含全部五条提示的子集可产生 120 个不同段落，它们表达相同含义，但句子顺序不同。

3.2.2 段落标注。 我们使用大语言模型对所有生成的段落进行标注。Salemi 和 Zamani [45] 发现，段落相关性标签与大语言模型对问题的可回答性之间存在很强的相关性。基于这一观察，对于每个问题，我们提供问题、候选段落和正确答案。我们将问题和段落输入三个模型：Gemma 3 1B、Qwen 3 4B 和 LLaMA 3.1 8B，让它们根据段落[7] 使用附录 A.4 中所示的少样本提示策略[8] 生成答案。为了确定大语言模型生成的答案是否正确，我们采用第 3.1.4 节所述的 GPT-Eval。如果至少有一个模型能够根据段落生成正确答案，我们就将该段落标记为 **2（完全相关）**，因为这表明该段落提供了足够的信息，使大语言模型能够推断出正确答案。在这种情况下，我们还会将生成的答案加入该问题的有效答案集合中。否则，我们将该段落标记为 **1（部分相关）**。这是因为，即使某个段落无法直接导出正确答案，它仍可能通过间接描述目标实体来提供有用的上下文信息。然而，此类段落通常会指向多个候选答案，而不是唯一确定正确答案。例如，在图 1（答案支持列）中，相关段落能够生成莱昂内尔·梅西，而部分相关段落可能导出莱昂内尔·梅西 或迭戈·马拉多纳。此外，包含对应于其他问题的提示的段落，会针对目标问题标记为**0（不相关）**。

[7]请注意，在第 3.1.4 节中，我们在没有任何上下文的情况下评估问题，而此处我们纳入了答案支持段落。[8]我们采用少样本策略，以更好地引导模型生成有效答案。

表2：检索器在 Quit、MS MARCO 和维基百科语料库上的性能。我们报告 Hit@k ($k \in \{1, 10, 50, 100\}$) 和平均倒数排名，并突出显示性能最佳的检索器。

检索器	语料库	Hit@1	Hit@10	Hit@50	Hit@100	MRR
BM25	Quit	0.00%	0.25%	0.44%	0.57%	0.04%
	MS MARCO	23.09%	41.15%	56.77%	63.08%	29.31%
	维基百科	45.60%	72.22%	84.55%	88.25%	54.73%
DPR	Quit	9.89%	16.62%	21.22%	23.74%	11.28%
	MS MARCO	14.41%	32.47%	49.19%	58.04%	20.55%
	维基百科	29.17%	57.41%	74.65%	80.32%	38.36%
ColBERT	Quit	12.41%	16.44%	19.40%	20.21%	12.62%
	MS MARCO	33.28%	49.88%	62.79%	68.63%	38.94%
	维基百科	60.53%	83.04%	90.86%	92.94%	68.86%
Contriever	Quit	6.49%	13.29%	18.95%	22.54%	8.15%
	MS MARCO	19.85%	43.63%	60.30%	66.72%	28.06%
	维基百科	45.37%	76.79%	87.44%	90.80%	56.56%
BGE	Quit	12.85%	21.98%	27.96%	30.23%	14.68%
	MS MARCO	27.55%	47.69%	63.31%	71.93%	34.35%
	维基百科	53.47%	81.71%	91.72%	94.33%	63.31%

3.2.3 人工验证。 为确保开发集和测试集的质量与可靠性，我们执行人工验证步骤。具体而言，我们会审查 GPT-Eval 判定为正确的答案，因为大语言模型——即使是 GPT 这样的先进模型——也可能偶尔生成错误答案或幻觉输出。因此，人工验证有助于我们构建更加可靠的开发集和测试集。在这一过程中，我们设计了附录 A.1 中图 7 所示的评估界面，使标注人员能够验证答案的正确性，并有效解决潜在问题。

3.2.4 标签更新。 在人工验证大语言模型生成的答案时，可能会无意间发现新的答案泄漏案例。例如，GPT-Eval 标记为错误的答案，之后可能被标注人员判定为正确，而这个新确认的正确答案需要接受答案泄漏风险验证。为解决这一问题，我们对所有问题重新执行第 3.1.2 节所述的问题筛选 流程，并删除至少有一个提示现已出现答案泄漏的问题。人工验证也可能影响段落标签。具体而言，如果某个段落经审核后不再能得出任何正确答案，其标签便会从完全相关（标签 2）更新为部分相关（标签 1）。完成这些调整后，我们得到经过清理并最终确定的训练集、开发集和测试集，它们共同构成了

Quit 基准。表1展示了其统计数据。

4 实验设置

在实验中，我们评估了多种检索器、重排序器和阅读器在推理问答任务及 Quit 基准上的表现。

对于检索器，我们采用五种检索方法：BM25 [43],DPR [25], ColBERT v 2 [46], Contriever [19], 和 BGE [8], , 所有方法均通过 Pyserini 工具包 [29] 进行访问。我们使用 Tevatron 库 [15],微调 DPR，并使用 ColBERT 的官方代码库[9]进行微调。

对于重排序器，我们使用 Rankify [1] 实现了五种模型：LiT5 (T5-3B) [51], 、MonoT5 (T5-3B) [37], 、RankGPT (LLaMA 3.1 8B) [49],、RankT5 (T5-3B) [62], 和 UPR (T0-3B) [44], 。MonoT5 使用 PyGaggle [40] 进行微调。

[9]https://github.com/stanford-futuredata/ColBERT

表3：Quit、MS MARCO 和维基百科语料库上各重排序器的比较。我们报告 Hit@k ($k = 1, 10, 50$) 和平均倒数排名。每个语料库中表现最佳的重排序器已突出显示，最高分数以粗体标出。

检索器	语料库	重排序器	Hit@1	Hit@10	Hit@50	MRR
BGE	Quit	LiT5	26.03%	29.21%	**33.80%**	28.10%
		MonoT5	**27.60%**	29.98%	32.35%	**28.54%**
		RankGPT	24.02%	29.05%	33.33%	25.70%
		RankT5	26.62%	**30.44%**	32.52%	27.80%
		UPR	26.85%	29.86%	32.70%	27.89%
ColBERT	MS MARCO	LiT5	**40.28%**	53.53%	64.24%	44.80%
		MonoT5	38.60%	55.44%	65.10%	44.27%
		RankGPT	28.88%	49.88%	62.79%	36.76%
		RankT5	39.53%	**55.56%**	**65.39%**	**45.01%**
		UPR	36.40%	54.69%	65.22%	42.62%
ColBERT	维基百科	LiT5	68.52%	85.82%	91.38%	74.59%
		MonoT5	68.17%	88.02%	91.84%	75.33%
		RankGPT	52.60%	82.99%	90.86%	64.60%
		RankT5	**71.88%**	**88.48%**	91.96%	**78.06%**
		UPR	65.57%	86.52%	**92.13%**	73.37%

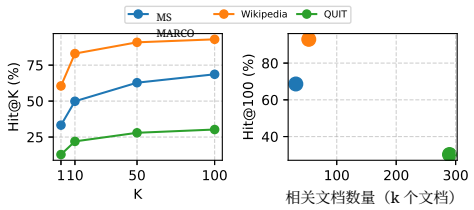

图4：左：不同语料库之间的 Hit@k 比较。右：不同语料库中，Hit@100 与相关段落（标签2）的数量之间的关系。

对于阅读器，我们采用了来自不同模型系列和参数规模的三种大语言模型：LLaMA 3.2 1B [13] 和 Gemma 3 4B [53]作为通用大语言模型，以及 Qwen 3 8B [58] 作为以推理为导向的模型。此设置使我们能够评估具有不同专长的大语言模型在推理式问题上的效果。如第 3.1.4 节所述，我们选择这些模型，是因为已验证它们都无法仅凭参数知识回答这些问题，从而确保对段落进行公平评估。所有模型均在两块 40GB 的 NVIDIA A40 GPU 上运行。

我们使用标准指标评估实验结果：**Hit@k**，用于衡量标准答案段落是否出现在前k 名中（针对 $k = 1, 5, 10, 50, 100$ 报告）；**Recall@k**，用于评估相关段落的覆盖率（针对 $k = 5, 10, 50$）；**MRR**，第一个相关段落排名的平均倒数；**NDCG@k**，用于通过分级相关性反映排名质量（针对 $k = 10$ 和$k = 100$）；以及**EM**，阅读器组件中预测答案与标准答案之间的精确匹配。

5 结果与讨论

5.1 对比基准测试

为了评估推理问答任务对当前检索器和重排序器而言有多大挑战，我们将Quit语料库[10] 与

[10]我们将训练集、开发集和测试集中所有答案支持段落的并集称为 Quit语料库。

表4：原始版和微调模型上的检索器性能。我们报告 Hit@k、召回率@k、平均倒数排名和 nDCG@k。

检索器	Hit@K		召回率@K			平均倒数排名@K	nDCG@K	
	1	5	5	10	50	100	10	100
原始版								
BGE	23.73%	27.37%	0.37%	0.75%	3.80%	25.45%	18.95%	21.14%
DPR	18.58%	21.18%	0.28%	0.54%	2.54%	19.77%	13.68%	13.64%
ColBERT	20.31%	21.12%	0.31%	0.63%	3.15%	20.69%	16.14%	17.51%
微调								
FT-DPR	20.91%	28.07%	0.32%	0.63%	3.12%	23.56%	14.98%	16.69%
FT-ColBERT	21.11%	23.84%	0.27%	0.53%	2.56%	20.83%	13.00%	13.98%

两个广泛使用的基准测试：维基百科 [25] 和MS MARCO [3]进行比较。具体而言，我们针对测试集中的问题，从全部三个语料库（Quit、维基百科和MS MARCO）中检索段落。

在传统问答系统中，相关性通常由答案包含来定义，即段落若包含正确答案，就被视为相关。图 1 中的 包含答案的 列展示了这一界定方式。然而，正如第3.2.2节所述，我们采用的标注方法有所不同，因为我们的段落经过专门设计以避免答案泄漏，因此答案包含并不适用。为此，我们使用自己的方法重新标注了从维基百科和 MS MARCO 检索到的段落。由于在我们的框架中，部分相关（标签 1）是根据提示与问题之间的关系定义的，因此无法直接应用于这些缺少提示的语料库。于是，我们采用标注为完全相关（标签 2）的段落；其判定依据是大语言模型能否利用该段落正确回答问题，从而确保与答案支撑段落进行公平比较。

为确保公平，我们使用五种不同的检索器，从 Quit、MS MARCO 和维基百科中检索排名前 100 的段落，结果见表 2。这些结果显示，Quit 与其他语料库在 Hit@k 和平均倒数排名方面存在巨大差距，凸显了在推理式问答任务中检索相关段落的固有难度。表中还显示，Quit 上表现最佳的检索器与 MS MARCO 和维基百科上的不同，进一步凸显了支持答案的段落与含答案段落之间的差异。图4进一步展示了这一差距。左图显示，维基百科的 Hit@k 始终最高，其次是 MS MARCO，Quit 最低，说明从 Quit 中检索的挑战更大。右图显示，尽管 Quit 检索到的段落中包含 289,660 个相关段落（标签 2），Hit@100 却只有 30.23%。相比之下，MS MARCO 和维基百科分别包含 31,195 个和 52,945 个相关的检索到的段落，Hit@100 分别达到 68.63% 和 92.94%。这表明，尽管人们可能会认为相关段落数量越多，Hit@100 就越高，但 Quit 并非如此。最后值得注意的是，MS MARCO 中段落的平均长度为 56.57 个词元，而 Quit 中为 58.62 个词元，这证实了观察到的差距并非由段落长度差异造成。

除检索器外，表3表明，Quit的最佳重排序器与MS MARCO和维基百科的最佳重排序器不同，这说明答案支持段落的表现不同于包含答案的段落，而且关于何为最相关段落的主流假设并不适用于答案支持段落。

表 5：应用于 BGE 和微调后的检索器的重排序器性能对比。我们报告 Hit@K、召回率@K、平均倒数排名和 nDCG@K。每种检索器的最佳重排序器以高亮显示，每列的最高分以粗体标出。

检索器	重排序器	Hit@K		召回率@K			平均倒数排名@K	归一化折损累计增益@K	
		1	5	5	10	50	100	10	100
BGE	LiT5	27.03%	29.17%	0.41%	0.82%	3.86%	28.10%	21.15%	21.49%
	MonoT5	**27.60%**	**29.46%**	**0.42%**	**0.84%**	**4.01%**	**28.54%**	**22.36%**	**21.81%**
	RankGPT	24.02%	27.49%	0.37%	0.75%	3.78%	25.70%	18.98%	21.14%
	RankT5	26.62%	29.05%	0.41%	0.82%	4.00%	27.80%	21.67%	21.69%
	UPR	26.85%	28.94%	0.42%	0.83%	4.01%	27.89%	21.06%	21.61%
FT-DPR	LiT5	27.84%	32.58%	0.40%	0.77%	3.08%	30.34%	19.31%	16.20%
	MonoT5	**28.01%**	31.89%	0.40%	**0.78%**	**3.39%**	30.24%	**20.34%**	**16.63%**
	RankGPT	20.25%	28.41%	0.31%	0.61%	2.90%	24.55%	14.21%	15.36%
	RankT5	26.97%	31.71%	0.40%	0.78%	3.39%	29.60%	19.98%	16.56%
	UPR	27.89%	**32.64%**	**0.41%**	0.78%	3.37%	**30.38%**	19.60%	16.49%
FT-ColBERT	LiT5	**23.32%**	25.41%	0.33%	0.63%	2.66%	24.53%	16.37%	14.52%
	MonoT5	22.69%	25.58%	**0.33%**	**0.65%**	**2.86%**	24.16%	**17.07%**	**14.80%**
	RankGPT	18.00%	23.84%	0.27%	0.53%	2.55%	20.75%	13.05%	13.98%
	RankT5	22.92%	25.52%	0.33%	0.64%	2.84%	**24.29%**	16.82%	14.74%
	UPR	22.80%	**25.81%**	0.33%	0.63%	2.84%	24.33%	16.31%	14.67%

表6：采用微调重排序器（FT-MonoT5）的检索器性能。我们报告原始版和微调检索器的 Hit@k、召回率@k、平均倒数排名和k 归一化折损累计增益。

检索器	Hit@K		召回率@K			平均倒数排名@K	归一化折损累计增益@K	
	1	5	5	10	50	100	10	100
原始版								
BM25	0.00%	0.06%	0.00%	0.00%	0.00%	0.05%	0.02%	0.01%
Contriever	12.69%	16.53%	0.18%	0.34%	1.53%	14.69%	8.95%	8.27%
BGE	23.44%	26.98%	0.37%	0.74%	3.76%	25.10%	18.67%	20.77%
DPR	18.29%	20.88%	0.28%	0.53%	2.50%	19.47%	13.50%	13.37%
ColBERT	20.07%	20.84%	0.31%	0.62%	3.11%	20.43%	15.89%	17.21%
微调								
FT-DPR	19.91%	28.07%	0.30%	0.60%	2.93%	24.07%	13.96%	15.36%
FT-ColBERT	18.11%	23.84%	0.27%	0.53%	2.56%	20.83%	13.00%	13.98%

5.2 检索器

为评估检索器在推理问答任务中对部分相关（标签1）和完全相关（标签2）段落的表现，我们测试了BGE（如表2所示，Quit上表现最佳的检索器）以及DPR和ColBERT，每种均包含原始版和微调变体。该设置使我们能够分析微调对检索答案支持段落的影响。

在对 DPR 和 ColBERT 进行微调时，我们参照 Chang [7], 的研究结论，尝试使用不同数量的正、负训练样本（1、5、10、50、100 和 200）。Chang 指出，训练期间增加正样本的数量可以提升检索性能。我们发现，DPR 的最优配置为 10 个正样本（和 10 个负样本[11]），ColBERT 的最优配置为 50 个正样本。所有配置的完整结果见附录 A.2 中的表10和表11。

如表4所示，BGE始终取得最高性能。微调使DPR的性能略有提升，但对ColBERT没有带来明显收益。我们推测，这种差异源于架构设计：DPR为问题和段落分别采用独立的BERT编码器，这可能有助于其更好地捕捉段落。相比之下，ColBERT依赖单个共享编码器和后期交互，这可能会限制其适应推理问答任务挑战的能力。此外，BGE的

[11]我们使用数量相同的正例段落和负例段落，负例从其他问题的正例段落中采样。

表 7：oracle设定下重排序器的性能。我们报告原始版和微调后的重排序器的 nDCG@k(k = 5，10，50，100)。最佳结果已突出显示。

重排序器	nDCG@K			
	5	10	50	100
原始版				
LiT5	72.94%	75.49%	79.34%	82.99%
RankGPT	65.02%	69.74%	78.09%	82.24%
RankT5	78.96%	80.18%	84.69%	87.49%
UPR	78.56%	79.72%	84.30%	87.25%
MonoT5	82.01%	82.95%	86.46%	88.71%
微调				
FT-MonoT5	83.56%	84.24%	87.08%	89.17%

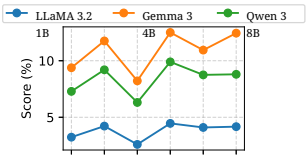

图5：在检索增强生成设置下，以 DPR（原始版和微调版）为检索器、MonoT5（原始版和微调版）为重排序器，并使用 LLaMA 3.2 1B、Gemma 3 4B 和 Qwen 3 8B 作为阅读器进行评估时，不同设置下精确匹配（EM）的趋势。

出色性能可能归功于其基于大语言模型的架构，该架构更擅长建模更深层的语义联系和隐含线索，而这些正是答案支撑段落的关键特征。

总体而言，这些结果表明微调只能带来有限的收益，不足以应对推理问答任务的复杂性。要解决这一挑战，可能需要全新的检索方法和范式。

5.3 重排序器

为了评估重排序器对推理问答任务的影响，我们使用五种不同的重排序器测试 Quit 基准。这些重排序器分别应用于 BGE（表现最佳的检索器）以及微调后的 DPR 和 ColBERT，以便分析它们对该任务的影响。表5显示，MonoT5 与 BGE 搭配时表现最佳，同时也是 FT-DPR 和 FT-ColBERT 的最佳重排序器。与表4中的检索器结果相比，重排序器仅带来有限的提升，这表明仅靠重排序不足以可靠地将段落提升至检索结果列表的前列。这表明，理解答案支撑段落对重排序器而言也颇具挑战。

除了原始版重排序器之外，我们还在不同设置下对 MonoT5 进行微调，并像第 5.2 节那样调整正例和负例样本的数量。我们发现，最佳配置使用 10 个正例段落，详细结果见附录 A.3 中的表12。然而，如表6所示，微调后的 MonoT5 表现不如其原始版（表5），这表明微调并不能帮助重排序器适应推理式问答

表8：基于大语言模型的阅读器在不同检索器–重排序器策略下的结果。我们报告了三种大语言模型的精确匹配（EM）。*UN* 指 Union$_{norm}$ 方法，*UF* 指Union^{freq} 方法。

检索器	重排序器	K	策略	LLaMA-32-1b	Gemma-3-4b	Qwen-3-8b
最优						
Oracle	Oracle	Oracle	Oracle	40.68%	90.16%	62.50%
预言检索器 +重排序器（原始版 & 微调）						
Oracle	MonoT5	5	UF	20.25%	50.41%	34.32%
Oracle	FT-MonoT5	5	UN	21.12%	51.10%	34.43%
检索器（原始版和微调版）						
BGE	–	5	UF	4.11%	13.14%	9.61%
DPR	–	5	UN	3.24%	9.38%	7.29%
FT-DPR	–	5	UN	2.60%	8.22%	6.31%
ColBERT	–	5	UN	3.82%	11.86%	8.28%
FT-ColBERT	–	5	UF	3.36%	8.85%	6.42%
检索器（原始版与微调） + 重排序器（原始版）						
BGE	MonoT5	5	UN	4.98%	15.34%	12.38%
DPR	MonoT5	5	UN	4.22%	11.75%	9.20%
FT-DPR	MonoT5	3	UF	4.46%	12.50%	9.90%
ColBERT	MonoT5	3	UF	4.46%	11.98%	8.46%
FT-ColBERT	MonoT5	3	UF	4.22%	11.75%	8.45%
检索器（原始版和微调版） + 重排序器（微调版）						
BGE	FT-MonoT5	5	UF	4.86%	13.89%	10.82%
DPR	FT-MonoT5	3	UF	4.10%	10.94%	8.75%
FT-DPR	FT-MonoT5	3	UN	4.17%	12.44%	8.80%
ColBERT	FT-MonoT5	5	UF	4.46%	11.81%	8.68%
FT-ColBERT	FT-MonoT5	5	UN	4.22%	11.69%	7.87%

任务。与检索器类似，重排序器需要采用新的方法和范式，才能有效处理答案支持段落。

我们还在 oracle 设定下评估了重排序器，在该设定中，我们假设检索器是完美的，能够检索出所有相关段落。在这一设置下，我们比较了 MonoT5 的原始版和微调版。表7显示，在假设检索器为 oracle 的情况下，微调后的 MonoT5 达到了最佳性能。这表明，原则上，微调可以帮助重排序器提升性能，但即使在理想检索条件下，改进幅度仍然很小。因此，表6中微调后的 MonoT5 表现不佳，可能应归因于检索器的局限性，而不是重排序器本身。尽管如此，结果证实仅靠微调是不够的，还需要更先进的方法来有效处理答案支持段落。

5.4 阅读器

在检索并重排序段落后，我们在 RAG 设置下使用排名前1、前3和前5的段落作为阅读器的输入。该设置使我们能够在推理问答任务上评估检索器和重排序器。前3和前5设置中的一个关键挑战是，简单地将段落拼接起来形成上下文会引入冗余，因为多个段落可能包含重叠的提示[12]。为了解决这一问题，我们提出了两种构建最终上下文的方法：(1) Union$_{norm}$ 和 (2) Union^{freq}。

我们表示每个段落 $p_i = \{s_{i1}, \ldots\ldots, s_{ij}\}$，其中 s_{ij} 表示段落 p_i 的第 j 个句子。Union$_{norm}$ 方法在保持顺序的同时应用标准并集，生成一组新的句子，并将其连接成一个段落。相比之下，Union ^{freq} 方法根据两个因素为每个句子评分：(i) 该句子所属段落的排名 ($rank(P_i)$)，以及 (ii) 该句子在该段落中的位置 ($pos_i(s)$)。评分函数定义如下：

[12]如第 3.2 节所述，这种重叠之所以出现，是因为问题的相关段落是根据为该问题设计的五条提示的子集和排列生成的。

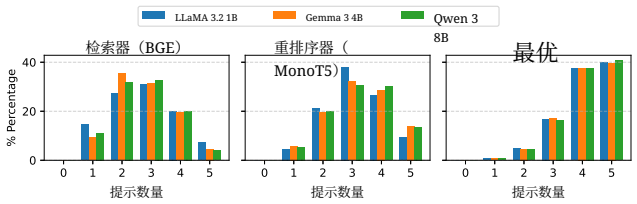

图6：在三种设置下，按提示数量划分的段落分布：不使用重排序器的检索器、使用重排序器的预言检索器，以及最优。

$$\text{score}(s) = \alpha \cdot \sum_{i:\, s \in P_i} \frac{1}{\text{rank}(P_i)} + \beta \cdot \sum_{i:\, s \in P_i} \frac{1}{\text{pos}_i(s)} \qquad (1)$$

其中，α 控制赋予排名较高的段落的权重，β 控制赋予每个段落中较早的句子的权重。在我们的实验中，我们设置 $\alpha = 0.6$ 和 $\beta = 0.4$[13]。计算分数后，按照分数顺序降序排列句子，并将排名前 k 个句子连接起来形成一个段落。

表8报告了基于大语言模型的阅读器中不同检索器和重排序器配置的结果。最优 配置对应一个神谕场景，即检索器和重排序器都能完美发挥作用。为模拟这种配置，我们穷尽评估每个问题的所有完全相关和部分相关段落；如果至少有一个段落能使大语言模型生成正确答案，就将该问题标记为可回答。由此展示了使用 LLaMA 3.2 1B、Gemma 3 4B 和 Qwen 3 8B 作为阅读器时所能达到的性能上限。这些结果表明，如果检索和重排序都能得到完美解决，使用 Gemma 3 4B 时，精确匹配率最高可达 90.16%，其他大语言模型也能达到同样高的数值。这说明 Quit 的质量很高，能够有效支持大语言模型生成正确答案。

然而，在预言检索器 配置下——即假设检索完美，但依靠重排序器对段落排序——精确匹配率仅为最优 分数的一半左右，这表明当前的重排序器无法有效地为段落排序。微调也未能缩小这一差距，说明重排序器需要针对推理式问答任务开发新的方法和范式。此外，仅使用检索器的结果最差，远低于最优 分数，凸显了当前检索器同样不够完善，必须超越现有范式。虽然重排序器能带来些许提升，但无法解决根本性挑战。如图5所示，微调检索器甚至可能降低性能，而重排序器带来的提升也微乎其微。

表4显示，微调后的DPR性能优于DPR-vanilla；然而，在基于大语言模型的阅读器阶段，其精确匹配低于原始模型。一个可能的解释是，微调后的检索器总体上能够检索到更多相关段落，但其排序并非最优。在这种情况下，重排序器可以帮助改善排序，从而在应用重排序器时取得优于原始模型的结果。微调后的重排序器在基于大语言模型的阅读器中的表现略逊于其原始版本，如表6所示，这表明它们带来的收益很小。然而，该结果显示，将微调后的检索器与微调后的重排序器结合使用，比其他设置能够取得更好的性能，尽管提升幅度非常小。

结果表明，Gemma 3 4B 等通用大语言模型的表现优于 Qwen 3 8B 等面向推理的大语言模型，即使前者的参数更少。这说明，依赖面向推理的大语言模型并不是解决推理问答任务的有效方案。总体而言，结果证实，当前的检索器和重排序器并不适用于推理问答任务，而仅靠微调也不可行；我们需要全新的方法。

图6展示了这一挑战。检索器和重排序器在识别包含三个句子的段落时表现最佳，而最优结果表明，对基于大语言模型的阅读器最有帮助的段落包含五条提示。这是因为较长的段落能提供更多与答案相关的信息，使基于大语言模型的阅读器能够更有把握地作答。检索器和重排序器在包含四条或五条提示的段落上表现较弱，这或许可以部分解释它们为何难以处理这些段落。

6 结论

本文介绍了**推理式问答**——一种全新的推理问答任务，要求问答（QA）系统从答案支撑段落中推断答案，而不是逐字提取答案。为开展这项研究，我们构建了**Quit**（**QU**estions requiring **I**nference from**T**exts，即需要从文本中进行推断的问题）数据集，其中包含7,401个问题和240万段落，均源自高收敛度提示，并通过结合基于大语言模型的可回答性判断与人工验证进行标注。我们对多种检索器、重排序器和基于大语言模型的阅读器进行了广泛评估，结果显示，现有问答方法虽然在传统的答案包含型数据集上表现有效，但在推理情境下却难以胜任。检索器往往无法找到正确的答案支撑段落，重排序器带来的改进有限，而微调所取得的提升也有限或不稳定。即使是面向推理的大语言模型，表现也不及规模更小的通用模型，这表明基于推断的推理仍是当前问答流水线中的一大瓶颈。这些发现揭示了问答中提取与推断之间的根本差距。通过将重点转向需要基于间接线索进行推理的问题，**推理式问答**开辟了一个新的研究方向，旨在开发能够从细微且分散的证据中得出结论的检索、重排序和推理方法。我们相信，这项工作为未来研究设计更接近人类推断式理解的检索增强推理系统铺平了道路。

致谢

此处展示的计算结果使用因斯布鲁克大学的 LEO高性能计算基础设施取得。本研究部分得到了智能信息检索中心、美国国家科学基金会资助项目编号2143434以及美国海军研究办公室合同编号N000142412612的支持。本文所表达的任何观点、研究结果、结论或建议均属于作者个人意见，不一定反映资助方的立场。

[13]在开发集上进行的网格搜索表明，$\alpha = 0.6$ 和 $\beta = 0.4$ 取得了最高的精确匹配分数。

参考文献

[1] Abdelrahman Abdallah、Bhawna Piryani、Jamshid Mozafari、Mohammed Ali 和 Adam Jatowt。2025年。Rankify：用于检索、重排序和检索增强生成的综合 Python 工具包。*arXiv* 预印本 *arXiv:2502.02464*（2025年）。[2] Heba Abdel-Nabi、Arafat Awajan 和 Mostafa Z. Ali。2022年。基于深度学习的问答：一项综述。*Knowl. Inf. Syst.* 65卷4期（2022年12月），1399–1485。doi:10.1007/s10115-022-01783-5[3] Payal Bajaj、Daniel Campos、Nick Craswell、Li Deng、Jianfeng Gao、Xiaodong Liu、Rangan Majumder、Andrew McNamara、Bhaskar Mitra、Tri Nguyen 等。2016年。MS MARCO：一个由人工生成的机器阅读理解数据集。*arXiv* 预印本 *arXiv:1611.09268*（2016年）。[4] Amy E. Barth 和 Amy Elleman。2017年。评估针对中年级阅读困难学生的多策略推理干预的影响。*Language, Speech, and Hearing Services in Schools* 48卷1期（2017年），31–41。arXiv:https://pubs.asha.org/doi/pdf/10.1044/2016_LSHSS-16-0041 doi:10.1044/2016 LSHSS-16-0041[5] Jonathan Berant、Andrew Chou、Roy Frostig 和 Percy Liang。2013年。基于问答对的 Freebase 语义解析。载于*2013*年自然语言处理实证方法会议论文集，David Yarowsky、Timothy Baldwin、Anna Korhonen、Karen Livescu 和 Steven Bethard（编）。计算语言学协会，美国华盛顿州西雅图，1533–1544。https://aclanthology.org/D13-1160/[6] Jannis Bulian、Christian Buck、Wojciech Gajewski、Benjamin Börschinger 和 Tal Schuster。2022年。To mayto, Tomahto：超越问答评估中的词元级答案等价性。载于*2022*年自然语言处理实证方法会议论文集，Yoav Goldberg、Zornitsa Kozareva 和 Yue Zhang（编）。计算语言学协会，阿拉伯联合酋长国阿布扎比，291 305 10 18653 1 2022 20 – 。doi: 。/v / 。[7] 2025 emnlp-main。Shuai Chang。。通过多个正例段落改进稠密段落检索。*arXiv* 预印本 *arXiv:2508.09534*（2025年）。[8] Jianlv Chen、Shitao Xiao、Peitian Zhang、Kun Luo、Defu Lian 和 Zheng Liu。2024年。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。*arXiv* 预印本 *arXiv:2402.03216*（2024年）。[9] Christopher Clark、Kenton Lee、Ming-Wei Chang、Tom Kwiatkowski、Michael Collins 和 Kristina Toutanova。2019年。BoolQ：探究自然是非问题令人意外的难度。载于*2019*年计算语言学协会北美分会会议论文集：人类语言技术，第*1*卷（长短论文），Jill Burstein、Christy Doran 和 Thamar Solorio（编）。计算语言学协会，美国明尼苏达州明尼阿波利斯，2924 2936 10 18653 1 19 1300 – 。doi: 。/v /N - [10] 2025 Debrup Das、Sam O'Nuallain 和 Razieh Rahimi。。RaDeR：具备推理能力的稠密检索模型。载于*2025*年自然语言处理实证方法会议论文集，Christos Christodoulopoulos、Tanmoy Chakraborty、Carolyn Rose 和 Violet Peng（编）。计算语言学协会，中国苏州，19970 19997 10 18653 1 2025 1011 – 。[11] 2019 doi: 。/v / 。emnlp-main。Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。。BERT：用于语言理解的深度双向变换器预训练。载于*2019*年计算语言学协会北美分会会议论文集：人类语言技术，第*1*卷（长短论文），Jill Burstein、Christy Doran 和 Thamar Solorio（编）。计算语言学协会，美国明尼苏达州明尼阿波利斯，4171–4186。doi:10.18653/ v1/N19-1423[12] Dheeru Dua、Yizhong Wang、Pradeep Dasigi、Gabriel Stanovsky、Sameer Singh 和 Matt Gardner。2019年。DROP：一个需要对段落进行离散推理的阅读理解基准。载于*2019*年计算语言学协会北美分会会议论文集：人类语言技术，第*1*卷（长短论文），Jill Burstein、Christy Doran 和 Thamar Solorio（编）。计算语言学协会，美国明尼苏达州明尼阿波利斯，2368–2378。doi:10.18653/v1/N19-1246[13] Abhimanyu Dubey、Abhinav Jauhri、Abhinav Pandey、Abhishek Kadian、Ahmad Al-Dahle、Aiesha Letman、Akhil Mathur、Alan Schelten、Amy Yang 等。2024年。Llama 3 模型群。*arXiv* 电子预印本，文章 arXiv:2407.21783（2024年7月），arXiv:2407 21783 10 48550 2407 21783 。页。doi: 。/arXiv. 。[14] Matteo Gabburo、Nicolaas Paul Jedema、Siddhant Garg、Leonardo F. R. Ribeiro 和 Alessandro Moschitti。2024年。衡量问答系统中的检索复杂度。载于计算语言学协会会议论文集：*ACL 2024*，Lun-Wei Ku、Andre Martins 和 Vivek Srikumar（编）。计算语言学协会，泰国曼谷，14636–14650。doi:10.18653/ v 1 2024 872 / 。findings-acl。Luyu Gao、Xueguang Ma、Jimmy Lin 和 Jamie Callan。2023年。Tevatron：一种高效灵活的神经检索工具包。载于第*46*届 *ACM SIGIR* 信息检索研究与发展国际会议论文集 （中国台北）（*SIGIR '23*）。计算机协会，美国纽约州纽约，3120–3124。doi:10.1145/3539618.3591805[16] Mor Geva、Daniel Khashabi、Elad Segal、Tushar Khot、Dan Roth 和 Jonathan Berant。2021年。亚里士多德用过笔记本电脑吗？一个问答基准，包含 Implicit Reasoning Strategies. 交易 *o* *f* 该协会 *f*或计算 *p*性《语言学》 9 2021 346 361 10 1162 _ _00370 ( )，–。doi: 。/ [17] 1994 tacl a Arthur C. Graesser 和 Natalie K. Person。问题提问——辅导期间的问题提问。《美国教育研究杂志》31，1（1994），104–137。arXiv：https://doi.org/10 3102 00028312031001104 10 3102 00028312031001104 。/ doi: 。/ [18] Helia Hashemi、Mohammad Aliannejadi、Hamed Zamani 和 W. Bruce Croft。2020。ANTIQUE：非事实型问答基准。在《信息检索进展》：第*42*届欧洲信息检索研究会议（*ECIR 2020*），葡萄牙里斯本，*2020*年*4*月*14*日至*17*日，论文集，第二部分（葡萄牙里斯本）。施普林格出版社，柏林、海德堡，166 173 10 1007 978 3 030 45442 5_21 –。doi: 。/ - - - - Gautier [19] Izacard、Mathilde Caron、Lucas Hosseini、Sebastian Riedel、Piotr Bo- janowski、Armand Joulin 和 Edouard Grave。2022。通过对比学习进行无监督稠密信息检索。《机器学习研究汇刊》（ 2022 = 1 7 0）。[20] https://openreview.net/forum?id jKN pXi b Anubhav Jangra、Jamshid Mozafari、Adam Jatowt 和 Smaranda Muresan。2025。探索提示生成研究领域：从过去到未来。《计算语言学协会汇刊》13（06 2025），505–528。doi:10.1162/tacl a 00751 [21] Adam Jatowt、Calvin Gehrer 和 Michael Färber。2023。自动提示生成。在*2023*年 *ACM SIGIR* 信息检索理论国际会议论文集（中国台北）（*ICTIR '23*）。计算机协会，纽约州纽约市，美国，117–123。doi:10.1145/3578337.3605119[22] Bowen Jin、Hansi Zeng、Zhenrui Yue、Jinsung Yoon、Sercan O Arik、Dong Wang、Hamed Zamani 和 Jiawei Han。2025。Search-R1：通过强化学习训练大语言模型进行推理并利用搜索引擎。在第二届语言建模会议。https://openreview.net/forum?id=Rwhi91ideu [23] Mandar Joshi、Eunsol Choi、Daniel Weld 和 Luke Zettlemoyer。2017。TriviaQA：用于阅读理解的大规模远程监督挑战数据集。在第*55*届计算语言学协会年会论文集（第一卷：长篇论文），Regina Barzilay 和 Min-Yen Kan（编）。计算语言学协会，加拿大温哥华，1601–1611。doi:10. 18653/v1/P17-1147[24] Ehsan Kamalloo、Nouha Dziri、Charles Clarke 和 Davood Rafiei。2023。在大语言模型时代评估开放域问答。在第*61*届计算语言学协会年会论文集（第一卷：长篇论文），Anna Rogers、Jordan Boyd-Graber 和 Naoaki Okazaki（编）。计算语言学协会，加拿大多伦多，5591–5606。doi:10.18653/v1/2023.acl-long.307[25] Vladimir Karpukhin、Barlas Oguz、Sewon Min、Patrick Lewis、Ledell Wu、Sergey Edunov、Danqi Chen 和 Wen-tau Yih。2020。用于开放域问答的稠密段落检索。在*2020*年自然语言处理实证方法会议（*EMNLP*）论文集，Bonnie Webber、Trevor Cohn、Yulan He 和 Yang Liu（编）。计算语言学协会，线上，6769–6781。doi:10.18653/v1/2020.emnlp-main.550[26] Tom Kwiatkowski、Jennimaria Palomaki、Olivia Redfield、Michael Collins、Ankur Parikh、Chris Alberti、Danielle Epstein、Illia Polosukhin、Jacob Devlin、Kenton Lee、Kristina Toutanova、Llion Jones、Matthew Kelcey、Ming-Wei Chang、Andrew M. Dai、Jakob Uszkoreit、Quoc Le 和 Slav Petrov。2019。Natural Questions：问答研究基准。《计算语言学协会汇刊》 7 2019 452 466 10 1162 _ _00276 （），–。doi: 。/ [27] 2005 tacl a Hyo-Jung O. Lee、Hyeon-Jin Kim 和 Myung-Gil Jang。。百科全书中的描述性问答。在*ACL*互动海报与演示环节论文集，Masaaki Nagata 和 Ted Pedersen（编）。计算语言学协会，美国密歇根州安娜堡，21–24。doi:10.3115/1225753。1225759[28] Patrick Lewis、Ethan Perez、Aleksandra Piktus、Fabio Petroni、Vladimir Karpukhin、Naman Goyal、Heinrich Küttler、Mike Lewis、Wen-tau Yih、Tim Rocktäschel、Sebastian Riedel 和 Douwe Kiela。2020。面向知识密集型自然语言处理任务的检索增强生成。在《神经信息处理系统进展》，H. Larochelle、M. Ranzato、R. Hadsell、M.F. Balcan 和 H. Lin（编），第33卷。Curran Associates, Inc.，9459–9474。https://proceedings.neurips.cc/ paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf [29] Jimmy Lin、Xueguang Ma、Sheng-Chieh Lin、Jheng-Hong Yang、Ronak Pradeep 和 Rodrigo Nogueira。2021。Pyserini：用于可复现信息检索研究的 Python 工具包，支持稀疏与稠密表示。在第*44*届 *ACM SIGIR* 信息检索研究与发展国际会议论文集（线上会议，加拿大）（*SIGIR '21*）。计算机协会，纽约州纽约市，美国，2356–2362。doi:10.1145/3404835.3463238[30] Yinhan Liu、Myle Ott、Naman Goyal、Jingfei Du、Mandar Joshi、Danqi Chen、Omer Levy、Mike Lewis、Luke Zettlemoyer 和 Veselin Stoyanov。2019。RoBERTa：经过稳健优化的 BERT 预训练方法。*arXiv*电子文献，文章 arXiv:1907.11692（2019年7月），arXiv:1907.11692页。doi:10.48550/arXiv.1907。11692[31] Meixiu Long、Duolin Sun、Dan Yang、Junjie Wang、Yecheng Luo、Yue Shen、Jian Wang、Hualei Zhou、Chunxiao Guo、Peng Wei 等。2025。Diver：一种用于推理密集型信息检索的多阶段方法。*arXiv*预印本 *arXiv:2508.07995*（2025年）。

[32] Shervin Minaee、Tomas Mikolov、Narjes Nikzad、Meysam Chenaghlu、Richard Socher、Xavier Amatriain 和 Jianfeng Gao。2024。大语言模型：综述。*arXiv*预印本 *arXiv:2402.06196*（2024年）。[33] Jamshid Mozafari、Abdelrahman Abdallah、Bhawna Piryani 和 Adam Jatowt。2024。探索开放域问答的提示生成方法。在计算语言学协会年会论文集：*EMNLP 2024*，Yaser Al-Onaizan、Mohit Bansal 和 Yun-Nung Chen（编）。计算语言学协会，美国佛罗里达州迈阿密，9327–9352。doi:10.18653/v1/2024.findings-emnlp.546[34] Jamshid Mozafari、Florian Gerhold 和 Adam Jatowt。2025。WikiHint：用于提示排序与生成的人工标注数据集。在第*48*届 *ACM SIGIR* 信息检索研究与发展国际会议论文集（意大利帕多瓦）（*SIGIR '25*）。计算机协会，纽约州纽约市，美国，3821 3831 10 1145 3726302 3730284 –。doi：。/。Jamshid Mozafari、Anubhav Jangra 和 Adam Jatowt。2024。TriviaHG：从事实型问题自动生成提示的数据集。在第*47*届 *ACM SIGIR* 信息检索研究与发展国际会议论文集（美国华盛顿哥伦比亚特区）（*SIGIR '24*）。计算机协会，纽约州纽约市，美国，2060 2070 10 1145 3626772 3657855 –。doi：。/。[35] Jamshid Mozafari、Bhawna Piryani、Abdelrahman Abdallah 和 Adam Jatowt。2025。HintEval：面向问题提示生成与评估的综合框架。*arXiv*预印本 *arXiv:2502.00857*（2025年）。[36] Rodrigo Nogueira、Zhiying Jiang、Ronak Pradeep 和 Jimmy Lin。2020。使用预训练序列到序列模型进行文档排序。在计算语言学协会年会论文集：*EMNLP 2020*，Trevor Cohn、Yulan He 和 Yang Liu（编）。计算语言学协会，线上，708–718。doi: 10 18653 1 2020 63 。/v /。findings-emnlp。[38] Hariom A Pandya 和 Brijesh S Bhatt。2021。问答综述：方向、挑战、数据集与评估矩阵。*arXiv*预印本 *arXiv:2112.03572*（2021年）。[39] Fabio Petroni、Tim Rocktäschel、Sebastian Riedel、Patrick Lewis、Anton Bakhtin、Yuxiang Wu 和 Alexander Miller。2019。语言模型可以作为知识库吗？在*2019*年自然语言处理实证方法会议和第*9*届国际自然语言处理联合会议（*EMNLP-IJCNLP*）论文集，Kentaro Inui、Jing Jiang、Vincent Ng 和 Xiaojun Wan（编）。计算语言学协会，中国香港，2463–2473。doi:10.18653/v1/D19-1250[40] Ronak Pradeep、Haonan Chen、Lingwei Gu、Manveer Singh Tamber 和 Jimmy Lin。2023。PyGaggle：开放域问答资源集。在《信息检索进展》，Jaap Kamps、Lorraine Goeuriot、Fabio Crestani、Maria Maistro、Hideo Joho、Brian Davis、Cathal Gurrin、Udo Kruschwitz 和 Annalina Caputo（编）。Springer Nature Switzerland，查姆，148–162。[41] Peng Qi、Haejun Lee、Tg Sido 和 Christopher Manning。2021。从文本中回答具有不同推理步骤的开放域问题。在*2021*年自然语言处理实证方法会议论文集，Marie-Francine Moens、Xuanjing Huang、Lucia Specia 和 Scott Wen-tau Yih（编）。计算语言学协会，线上及多米尼加共和国蓬塔卡纳，3599 3614 10 18653 1 2021 292 –。doi：。/v /。emnlp-main。[42] Pranav Rajpurkar、Jian Zhang、Konstantin Lopyrev 和 Percy Liang。2016。SQuAD：100,000+ 个用于文本机器阅读理解的问题。在*2016*年自然语言处理实证方法会议论文集，Jian Su、Kevin Duh 和 Xavier Carreras（编）。计算语言学协会，美国得克萨斯州奥斯汀，2383 2392 10 18653 1 16 1264 –。doi：。/v /D - 2009。[43] Stephen Robertson 和 Hugo Zaragoza。。概率相关性框架：BM 25 3, 4 2009 333 389 10 1561 1500000019 及其后续。[44] 〔4月〕，–。doi：。/ Devendra Sachan、Mike Lewis、Mandar Joshi、Armen Aghajanyan、Wen-tau Yih、Joelle Pineau 和 Luke Zettlemoyer。2022。使用零样本问题生成改进段落检索。在*2022*年自然语言处理实证方法会议论文集，Yoav Goldberg、Zornitsa Kozareva 和 Yue Zhang（编）。计算语言学协会，阿布扎比、阿拉伯联合酋长国，3781 3797 10 18653 1 2022 249 –。doi：。/v /。emnlp-main。[45] Alireza Salemi 和 Hamed Zamani。。评估检索增强生成中的检索质量。在第*47*届 *ACM SIGIR* 信息检索研究与发展国际会议论文集（美国华盛顿哥伦比亚特区）（*SIGIR '24*）。计算机协会，纽约州纽约市，美国，2024。2395 2400 10 1145 3626772 3657957 –。doi：。/。Keshav Santhanam、Omar Khattab、Jon Saad-Falcon、Christopher Potts 和 Matei Zaharia。2022。ColBERTv2：通过轻量级后期交互实现高效检索。在*2022*年计算语言学协会北美分会会议：人类语言技术论文集，Marine Carpuat、Marie-Catherine de Marneffe 和 Ivan Vladimir Meza Ruiz（编）。计算语言学协会，美国华盛顿州西雅图，3715–3734。doi:10.18653/v1/2022.naacl-main.272[47] Rulin Shao、Rui Qiao、Varsha Kishore、Niklas Muennighoff、Xi Victoria Lin、Daniela Rus、Bryan Kian Hsiang Low、Sewon Min、Wen-tau Yih、Pang Wei Koh 等。2025。ReasonIR：训练用于推理任务的检索器。*arXiv*预印本 *arXiv:2504.20595*（2025年）。[48] Hongjin SU、Howard Yen、Mengzhou Xia、Weijia Shi、Niklas Muennighoff、Han yu Wang、Liu Haisu、Quan Shi、Zachary S Siegel、Michael Tang、Ruoxi Sun、Jinsung Yoon、Sercan O Arik、Danqi Chen 和 Tao Yu。2025年。BRIGHT：面向高强度推理检索的真实且具有挑战性的基准。载于第十三届国际学习表征会议。https://openreview.net/ forum?id=ykuc5q381b[49] Weiwei Sun、Lingyong Yan、Xinyu Ma、Shuaiqiang Wang、Pengjie Ren、Zhumin Chen、Dawei Yin 和 Zhaochun Ren。2023年。ChatGPT 擅长搜索吗？探究大语言模型作为重排序智能体的表现。载于*2023*年自然语言处理实证方法会议论文集，由 Houda Bouamor、Juan Pino 和 Kalika Bali（编）主编。计算语言学协会，新加坡，14918 14937 10 18653 1 2023 923 –。doi：。/v / .emnlp-main。[50] Alon Talmor、Jonathan Herzig、Nicholas Lourie 和 Jonathan Berant。。常识问答：一项针对常识知识的问答挑战。载于*2019*年计算语言学协会北美分会会议论文集：人类语言技术，第*1*卷（长短论文），由 Jill Burstein、Christy Doran 和 Thamar Solorio（编）主编。计算语言学协会，美国明尼苏达州明尼阿波利斯，4149–4158。doi:10.18653/v1/N19-1421[51] Manveer Singh Tamber、Ronak Pradeep 和 Jimmy Lin。2023年。缩减规模，提升效果：使用序列到序列编码器—解码器模型进行高效的零样本列表式重排序。*arXiv* 预印本 *arXiv:2312.16098*（2023年）。[52] Harish Tayyar Madabushi 和 Mark Lee。2016年。利用问题句法与语义实现高准确率的基于规则的问题分类。载于*COLING 2016*：第*26*届国际计算语言学会议论文集：技术论文，由 Yuji Matsumoto 和 Rashmi Prasad（编）主编。COLING 2016 组织委员会，日本大阪，1220–1230。[53] Gemma 团队、Aishwarya Kamath、Johan Ferret、Shreya Pathak、Nino Vieillard、Ramona Merhej、Sarah Perrin、Tatiana Matejovicova、Alexandre Ramé 等。2025年。Gemma 3 技术报告。arXiv:2503.19786 [cs.CL] https://arxiv.org/ abs/2503.19786[54] Mengqiu Wang 等。2006年。事实型问答中答案抽取技术综述。《计算语言学》1，1（2006），1–14。[55] Yu Wang、Vijay Srinivasan 和 Jin Hongxia。2022年。用于多跳推理的知识型问答（KBQA）系统新概念。载于*2022*年计算语言学协会北美分会会议论文集：人类语言技术，由 Marine Carpuat、Marie-Catherine de Marneffe 和 Ivan Vladimir Meza Ruiz（编）主编。计算语言学协会，美国西雅图，4007–4017。doi:10.18653/v1/2022.naacl-main.294[56] Jason Wei、Xuezhi Wang、Dale Schuurmans、Maarten Bosma、brian ichter、Fei Xia、Ed H. Chi、Quoc V Le 和 Denny Zhou。2022年。思维链提示能够激发大语言模型的推理能力。载于神经信息处理系统进展，由 Alice H. Oh、Alekh Agarwal、Danielle Belgrave 和 Kyunghyun Cho（编）主编。https://openreview.net/forum?id=_VjQlMeSB_J[57] Benfeng Xu、Chunxu Zhao、Wenbin Jiang、PengFei Zhu、Songtai Dai、Chao Pang、Zhuo Sun、Shuohuan Wang 和 Yu Sun。2023年。语言模型的检索增强领域适应。载于第八届自然语言处理表征学习研讨会论文集（*RepL4NLP 2023*），由 Burcu Can、Maximilian Mozes、Samuel Cahyawijaya、Naomi Saphra、Nora Kassner、Shauli Ravfogel、Abhilasha Ravichander、Chen Zhao、Isabelle Augenstein、Anna Rogers、Kyunghyun Cho、Edward Grefenstette 和 Lena Voita（编）主编。计算语言学协会，加拿大多伦多，54–64。doi:10.18653/v1/2023.repl4nlp-1.5[58] An Yang、Anfeng Li、Baosong Yang、Beichen Zhang、Binyuan Hui、Bo Zheng、Bowen Yu、Chang Gao、Chengen Huang、Chenxu Lv 等。2025年。Qwen3 技术报告。*arXiv* 预印本 *arXiv:2505.09388*（2025年）。[59] Zhilin Yang、Peng Qi、Saizheng Zhang、Yoshua Bengio、William Cohen、Ruslan Salakhutdinov 和 Christopher D. Manning。2018年。HotpotQA：多样化、可解释多跳问答数据集。载于*2018*年自然语言处理实证方法会议论文集，由 Ellen Riloff、David Chiang、Julia Hockenmaier 和 Jun'ichi Tsujii（编）主编。计算语言学协会，比利时布鲁塞尔。doi:10.18653/v1/D18-1259[60] Zongmeng Zhang、Jinhua Zhu、Wengang Zhou、Xiang Qi、Peng Zhang 和 Houqiang Li。2024年。BoolQuestions：稠密检索能理解语言中的布尔逻辑吗？载于计算语言学协会 *EMNLP 2024* 论文集，由 Yaser Al-Onaizan、Mohit Bansal 和 Yun-Nung Chen（编）主编。计算语言学协会，美国佛罗里达州迈阿密，2767–2779。doi: 10 18653 1 2024 156 。/v / .findings-emnlp。[61] Minjun Zhu、Yixuan Weng、Shizhu He、Kang Liu 和 Jun Zhao。2022年。Reason-ChainQA：基于文本的复杂问答与可解释证据链。载于*2022*年中国自动化大会（*CAC*）。5431–5436。doi:10.1109/ CAC57257.2022.10055048[62] Honglei Zhuang、Zhen Qin、Rolf Jagerman、Kai Hui、Ji Ma、Jing Lu、Jianmo Ni、Xuanhui Wang 和 Michael Bendersky。2023年。RankT5：通过排序损失微调 T5 以进行文本排序。载于第*46*届 *ACM SIGIR* 信息检索研究与发展国际会议论文集 （中国台北）（*SIGIR '23*）。计算机协会，美国纽约州纽约，2308–2313。doi:10.1145/3539618.3592047

A 附录 A.1 人工验证



图7：人工标注界面。问题 部分显示问题及其正确答案，而候选答案 部分列出大语言模型在325个段落中生成的所有答案。标注人员人工验证并选择正确答案。

表 9：评估者的人口统计信息

评估器	性别	Age	教育程度
1	女性	19	高中
2	女性	30	学士学位
3	Male	40	硕士学位
4	Male	32	PhD

为提高开发集和测试集的可靠性，我们对 GPT-Eval 标记为正确的答案进行人工验证，以减少大语言模型可能产生的错误或幻觉。标注人员使用图7所示的自定义界面验证答案是否正确，并处理有歧义的情况。

具体而言，我们展示 GPT-Eval 判定为正确的所有生成答案，并突出显示那些与从原始来源（TriviaHG [35] 和 WikiHint [34]）中提取的标准答案完全匹配的答案。随后，人工标注人员会审核这些答案，并选择或取消选择他们认为标签正确或错误的答案。对于每个问题，标注人员会看到问题文本、生成答案以及相关的标准答案。他们需要评估生成答案是否与标准答案在语义上等价，即使两者在字面上有所不同（例如，“美国”与“美利坚合众国”）。如果某个生成答案虽然被 GPT-Eval 接受，但被认定为错误，就会手动取消其标记。反之，对于正确但未突出显示的答案，可以将其选中，以确保没有遗漏任何有效答案。这个流程有助于纠正假阳性和假阴性，从而为下游评估提供更高质量的段落标签。表 9 总结了参与这一验证流程的人工标注人员的人口统计背景。

A.2 微调后的检索器

为微调 ColBERT 和 DPR，我们尝试使用不同数量的正例样本和负例样本。鉴于答案支撑段落的复杂性和推理要求，我们假设即使只有一个正例，也可能足以实现有效的微调。为验证这一点，我们针对 ColBERT 和 DPR，分别测试了 1、5、10、50、100 和 200 个正例样本及负例样本，旨在找出最优配置。

结果见表10和表11，其中报告了每种组合的性能。根据这些结果，我们发现，使用50个正样本时 ColBERT 的性能最佳，而使用10个正样本时 DPR 的性能最佳。因此，我们在实验中采用这些配置作为 ColBERT 和 DPR 的最终微调版本。

A.3 微调重排序器

为了微调重排序器，我们使用 MonoT5 进行实验，因为它是一种广泛使用的重排序模型，并且具有较强的基线性能。与检索器类似，我们测试不同数量的正样本和负例样本——1、5、10、50、100和200——以分析训练数据规模对答案支撑段落重排序的影响。

结果列于表12，其中报告了每种配置的性能。根据这些结果，我们发现，使用 10 个正样本时，MonoT5 的性能最佳。因此，我们将此配置作为实验中使用的 MonoT5 最终微调版本。

表10：不同正例段落数量下微调后的 ColBERT 检索器的性能。我们报告 Hit@k、Recall@k、平均倒数排名和 nDCG@k。最佳结果以高亮显示。

检索器	Hit@K		召回率@K			平均倒数排名@K	nDCG@K	
	1	5	5	10	50	100	10	100
原始版 ColBERT	20.31%	21.12%	0.31%	0.63%	3.15%	20.69%	16.14%	17.51%
微调								
ColBERT（1 个位置）	13.95%	17.88%	0.21%	0.42%	1.95%	15.93%	10.40%	10.36%
ColBERT（5 个位置）	15.74%	20.14%	0.24%	0.46%	2.20%	17.93%	11.61%	11.93%
ColBERT（10 个位置）	16.90%	20.37%	0.25%	0.49%	2.40%	18.64%	12.21%	13.20%
ColBERT（50 位置）	21.11%	23.84%	0.27%	0.53%	2.56%	20.83%	13.00%	13.98%
ColBERT（100 个位置）	18.46%	22.92%	0.29%	0.56%	2.75%	20.56%	13.80%	14.87%
ColBERT（200 位置）	17.77%	23.32%	0.28%	0.55%	2.69%	20.35%	13.21%	14.58%

表11：不同数量正例段落下微调后的DPR检索器的性能。我们报告 Hit@k、Recall@k、平均倒数排名和 nDCG@k。最佳结果已突出显示。

检索器	Hit@K		召回率@K			平均倒数排名@K	归一化折损累计增益@K	
	1	5	5	10	50	100	10	100
原始版 DPR	18.58%	21.18%	0.28%	0.54%	2.54%	19.77%	13.68%	13.64%
微调								
DPR（1 个正样本）	19.97%	24.83%	0.31%	0.61%	2.97%	22.42%	14.69%	16.05%
DPR（5 个位置）	20.83%	25.98%	0.31%	0.62%	3.08%	23.50%	14.61%	16.44%
DPR（10 位置）	20.91%	28.07%	0.32%	0.63%	3.12%	23.56%	14.98%	16.69%
DPR（50 个位置）	20.31%	26.97%	0.31%	0.62%	3.05%	23.70%	14.49%	16.17%
DPR（100 位置）	20.83%	27.43%	0.31%	0.61%	3.03%	24.19%	14.14%	15.86%
DPR（200 位置）	20.06%	25.75%	0.30%	0.60%	2.93%	24.07%	13.96%	15.36%

系统： 你是一个根据所提供的上下文回答问题的助手。你只需给出准确答案，不要用句子作答。

用户： 请根据上下文在以下条件下回答问题：1. 答案不应是句子，而应是若干词语。2. 不要生成“抱歉”或“我无法……”这类句子；请改用“无答案”。3. 不要生成解释、推理或完整句子——只提供确切答案。4. 如果无法从上下文推断出答案，则仅回复“无答案”。

上下文： 他是美国第44任总统。他于2009年至2017年担任总统。他是美国首位非裔美国人总统。他是民主党成员。他于1961年8月4日出生于夏威夷檀香山。**问题：** 谁在2009年获得了诺贝尔和平奖?

助手： 巴拉克·奥巴马

上下文： 这个国家的首都是巴黎。这个国家位于西北欧。这个国家历史悠久，在国际事务中发挥着重要作用。这个国家的官方语言是法语。这个国家使用的货币是欧元。**问题：** Édouard Daladier 于 1933年成为哪个国家的总理?

助手： 法国

上下文： 这是一年中最寒冷的季节。这个季节许多地区会下雪。许多人会在这个季节庆祝圣诞节和除夕。这个季节白昼较短，夜晚较长。许多动物会在这个季节冬眠。**问题：** 如果你有一株日本十大功劳，它会在哪个季节开花?

助手： 冬季

上下文： 这是一项起源于美国的团队运动，使用椭圆形球。比赛目标是将球推进到对方的达阵区以得分。可以通过带球越过对方的得分线、将球传给达阵区内的队友，或将球踢过对方的球门柱得分。比赛分为四节，每节 15 分钟。**问题：** 哪项运动采用“哈佛规则”?

助手： 美式橄榄球

上下文： 他于1889年4月20日出生在奥地利因河畔布劳瑙。他是纳粹党的领导人。他于1933年成为德国总理。1934年，他获得了元首兼帝国总理的头衔。1939年9月1日，他入侵波兰，由此在欧洲挑起了第二次世界大战。**问题：** 谁于1938年被授予奥地利哈斯拉赫的荣誉市民称号，并于2004年被撤销这一荣誉?

助手： 阿道夫·希特勒

上下文： 其首都是北京，人口超过10亿。这个国家拥有超过3000年绵延不断的文明。
问题： 哪个国家与14个国家接壤，并在其辽阔的国土上统一使用一个时区?

助手： 中国

图8：推理期间提供给语言模型的少样本提示。每个示例都展示了如何利用基于提示的上下文，以简短回答简洁地回答问题（或在上下文信息不足时回答无答案 ）。最后一组问答展示了待回答的新查询，模型生成的答案以红色 标出，以便识别。

表12：正例段落数量不同时，微调后的 MonoT5 重排序器的性能。我们报告归一化折损累计增益@k。

重排序器	nDCG@K					
	1	5	10	20	50	100
原始版 MonoT5	80.83%	82.01%	82.95%	84.24%	86.46%	88.71%
微调						
MonoT5（1 个位置）	71.72%	76.07%	78.35%	80.61%	83.88%	86.78%
MonoT5（5 个位置）	81.91%	82.94%	83.61%	84.67%	86.65%	88.87%
MonoT5（10 位置）	82.95%	83.56%	84.24%	85.30%	87.08%	89.17%
MonoT5（50 个位置）	75.42%	77.87%	79.49%	81.29%	84.09%	86.92%
MonoT5（100 个位置）	75.58%	77.45%	78.73%	80.47%	83.47%	86.43%
MonoT5（200 位置）	75.12%	77.14%	78.08%	80.07%	82.71%	85.44%

A.4 少样本提示

图8展示了用于评估语言模型的少样本提示。它由若干个问题–上下文–答案 示例组成，其中每个上下文都由一些提示拼接而成，这些提示间接描述某个实体、概念或事件。该提示演示了如何根据上下文线索推断答案，而不生成解释；系统指令要求输出简短的短语级答案，在信息不足时使用“无答案”，并避免进行推理或论证。最后一个示例给出了一个包含未见过的提示的新推理型问题，促使模型按照学到的模式给出答案。