**UniGen: A Unified Generative Framework for Retrieval and Question Answering with Large Language Models**

UniGen：利用大型语言模型进行检索和问题解答的统一生成框架

**太长不看版：**

在本文中，研究人员提出了开发了一个生成式框架UniGen，其中包含一个多头解码器结构，用于同时学习检索和问答任务，并引入了由LLM生成的Q-connector和D-connector，使用其克服了UniGen的开发难题，提出了一种迭代方法，通过利用生成的答案和检索到的文档来改进生成式检索和问答任务，有效地增强了模型的能力，UniGen较传统模型有着更低的成本和更好的表现，引起了广泛的关注

**摘要**

生成式信息检索（Generative Information Retrieval）在信息检索和自然语言处理领域引起了广泛关注，涵盖了生成式文档检索（Generative Document Retrieval，GDR）和基于语境的答案生成（Grounded Answer Generation，GAR）这两个主要任务。现有的GDR和GAR方法依赖于独立的检索和阅读器模块，这限制了同时进行的优化。为了克服这个问题，研究人员提出了UniGen，这是一个用于检索和问答的统一的生成框架，它利用大型语言模型的功能将这两个任务集成到一个生成模型中。UniGen采用共享编码器和两个不同的解码器来生成检索和问答。为了促进两个任务的学习，研究人员引入了由大型语言模型生成的连接器，以弥合查询输入和生成目标之间以及文档标识符和答案之间的差距。此外，研究人员提出了一种迭代增强策略，利用生成的答案和检索的文档循环改进这两项任务。通过在MS MARCO和NQ数据集上的广泛实验，研究人员证明了 UniGen 的有效性，展示了其在检索和问答任务中的卓越性能

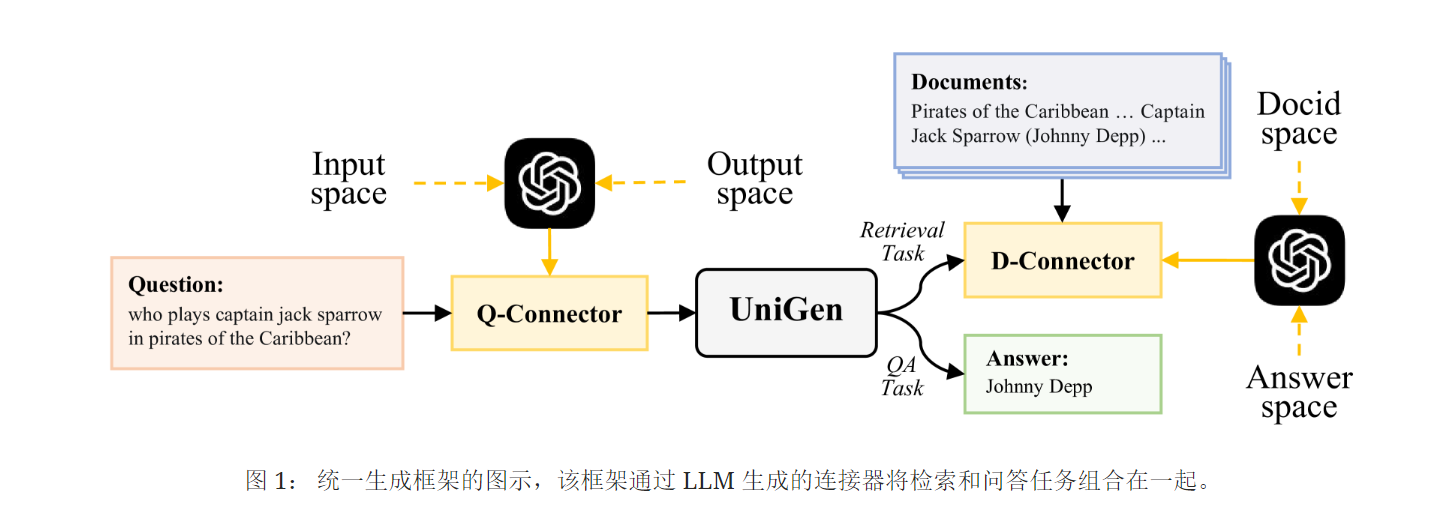
**简介**

近年来，生成式信息检索一直是研究的焦点，它涉及从庞大的语料库（如维基百科）中生成与特定查询相关的信息。该领域主要包括两个任务：生成式文档检索（GDR）和基于语境的答案生成（GAR）。GDR通过可以直接生成文档标识符（docids）的编码器-解码器架构来对文档进行排序。同时，GAR生成与特定段落相匹配的答案。

近年来GDR和GAR技术均有所发展，但对生成式检索和问答（QA）任务分别进行优化需要不同的训练技术和训练数据，还要付出额外的时间成本。研究人员注意到这两个任务都可以采用编码器-解码器结构，并具有两个基本特点：

（1）需要深入理解查询输入背后的语义意义；

（2）需要理解和记忆语料库中的知识。

受这些共享特性的启发，研究人员提出了一个统一的生成式框架UniGen，用于同时提高检索和QA能力。UniGen采用一个共享编码器和两个独立的解码器：检索解码器和QA解码器。使用共享的编码器可以提高模型对输入的理解能力，共享两个任务知识可以增强整体性能。如图1所示，检索解码器为检索任务生成docids，而QA解码器为QA任务生成答案。

此框架的设计旨在通过共享底层表示来减少冗余，并允许两个任务之间的信息交换。例如，当模型在优化检索任务时学到的语义理解可能会提升QA任务中答案生成的准确性，反之亦然。这种统一的框架有望提高模型的效率，减少训练时间，并可能提升两个任务的最终性能。

然而，在这样一个统一的生成式信息检索（IR）框架中，存在两个明显的差距阻碍训练过程，包括：

（1）输入-输出差距：输入的查询通常很简短且缺乏上下文语义，导致查询输入与生成目标之间存在差异。

（2）docid-答案差距：传统的docid通常是不可读的序列，将其与“答案生成”进行联合学习较为困难，因此docid和答案之间有一定差距。

为了弥补这些差距，研究人员引入了连接器（Connectors）的概念作为桥梁。具体来说，他们引入了Q-Connector和D-Connector，它们分别丰富了查询的上下文和精炼了文档的内容，从而弥合了输入-输出差距和docid-答案之间的鸿沟。考虑到生成这些连接器是一项高度依赖知识的任务，他们提出利用大型语言模型（LLMs），以有效地完成此任务。亦如图1所示。

此外，已有研究表明，检索和QA任务的结合可以产生有益的联系。具体来说，被检索解码器获取的文档可以作为补充知识来提高答案生成的能力。同样，由QA解码器生成的答案可以有效促进文档检索。基于这一见解，研究人员进一步提出了一种迭代增强策略，以在数据级别（date level）上优化检索和QA任务的性能。该策略涉及将前一次迭代的检索到的文档和生成的答案作为后续模型迭代的输入。研究人员通过采用这种迭代过程不断精炼模型输入，从而在两个任务中都实现了卓越的性能。

研究人员在公开数据集MS MARCO和Natural Questions（NQ）上进行了一系列实验，验证了方法的有效性。

纵览全文，他们主要贡献如下：

• 统一的生成式框架：研究人员开发了一个生成式框架，其中包含一个多头解码器结构，用于同时学习检索和问答任务。

• LLM增强连接器：研究人员引入了由LLM生成的Q-connector和D-connector，它们在输入-输出和docid-答案之间建立了语义连接，分别增强了查询语义和精炼了文档内容。

• 迭代增强策略：研究人员提出了一种迭代方法，通过利用生成的答案和检索到的文档来改进生成式检索和问答任务。

**相关工作**

**生成式检索（Generative Retrieval）**

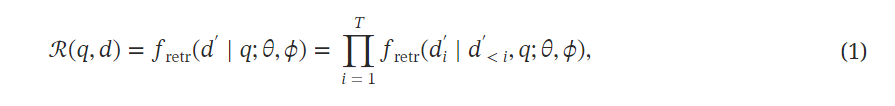
生成式检索是一种创新的信息检索方法，它利用预训练语言模型的参数作为可微分索引，从而能够直接生成相关文档标识符。

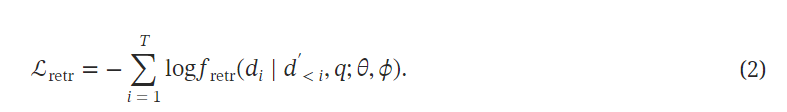
**开放域问答（Open-Domain Question Answering）**

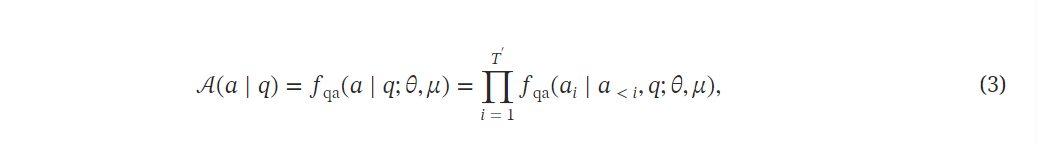
开放域问答是指在不依赖上下文信息的情况下为查询提供解决方案。它涉及两种主要形式：闭卷和开卷。 在闭卷 QA 中，模型无法访问外部知识库，必须在其参数内内化所有必要的信息。在开卷 QA 中，模型可以在答案生成过程中利用维基百科等知识库。典型的过程包括两个主要组成部分：一个检索模块，用于搜索相关上下文的知识库，另一个阅读模块，用于分析检索到的信息以制定解决方案。

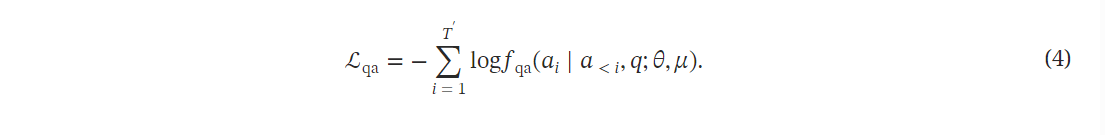
**方法论**

**任务制定**

在文档语料库中涉及文档*d*时，让*d’*表示文档*d*的预构建 docid。 对于生成式检索任务，在给定的查询*q*下，通过下式获得*q*，*d*之间的相关性ℛ

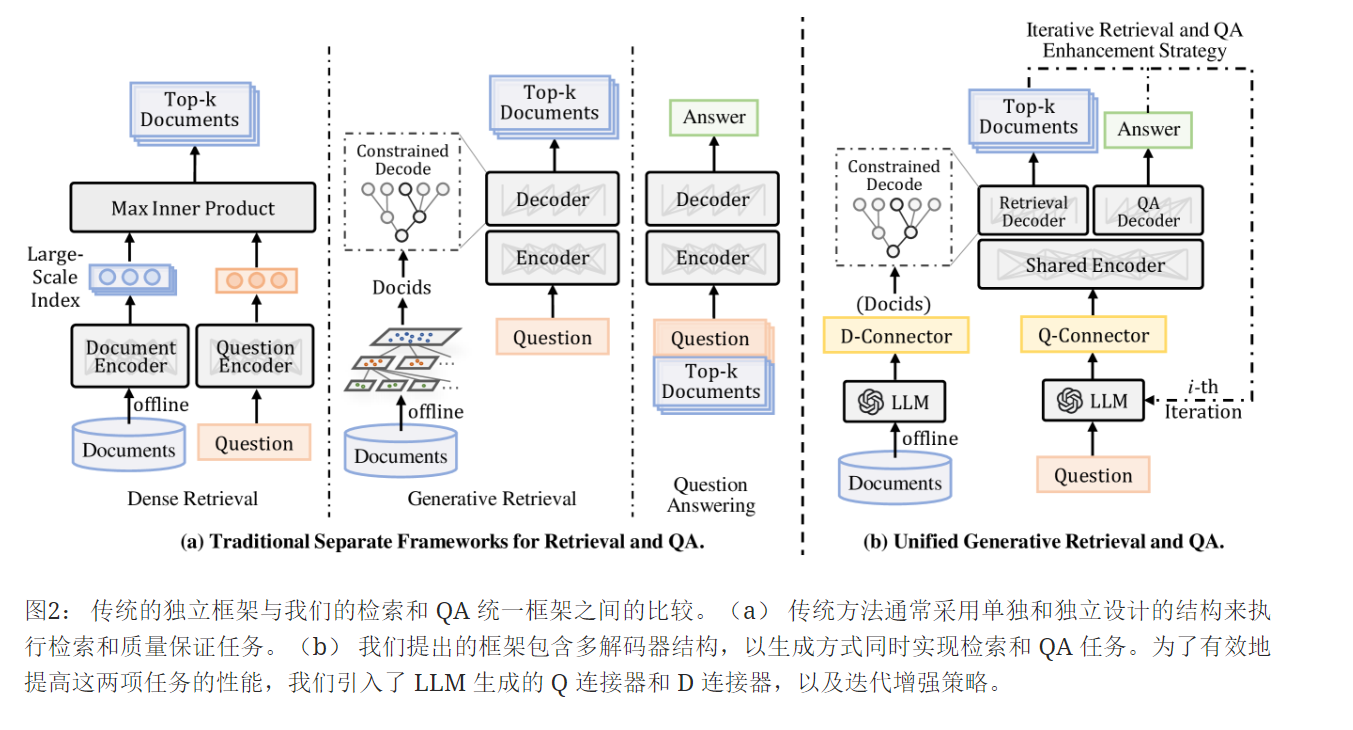
这里的*T*是目标标识符*d’*的长度，*d’i*是*d’*的第i个token，*fretr*是生成式检索模型，包含带有参数θ的编码器和带有参数ф的检索解码器。以最大限度地提高在方程（[1](https://ar5iv.labs.arxiv.org/html/2312.11036#Sx3.E1)）中生成目标文档标识符的可能性。在训练期间使用teacher-forcing来优化以下交叉熵损失：

同样，对于 QA 任务，给定一个查询*q*，生成答案*a*的概率*A*由下式获得

这里的*T’*是答案*a*的长度，*ai*是答案*a*的第i个token，*fqa*是生成式QA模型，拥有共享参数θ的编码器和一个独有的QA解码器，其参数为μ。同样，参数θ和μ的优化是通过标准的seq-to-seq任务实现的，该任务通过使用teacher-forcing最大化方程（3）中目标序列的概率，QA损失函数可由下式表示：

**UniGen：统一的生成式检索和 QA**

**模型架构**

UniGen 框架引入了一个多解码器结构，以同时处理检索和 QA 任务。这与传统方法不同，传统方法依赖于每个任务的单独和独立设计的体系结构。图 2 展示了UniGen 框架与传统方法之间的对比，在传统方法中，密集检索依赖于大规模文档索引，而生成检索和 QA 方法通常是不同的模块。

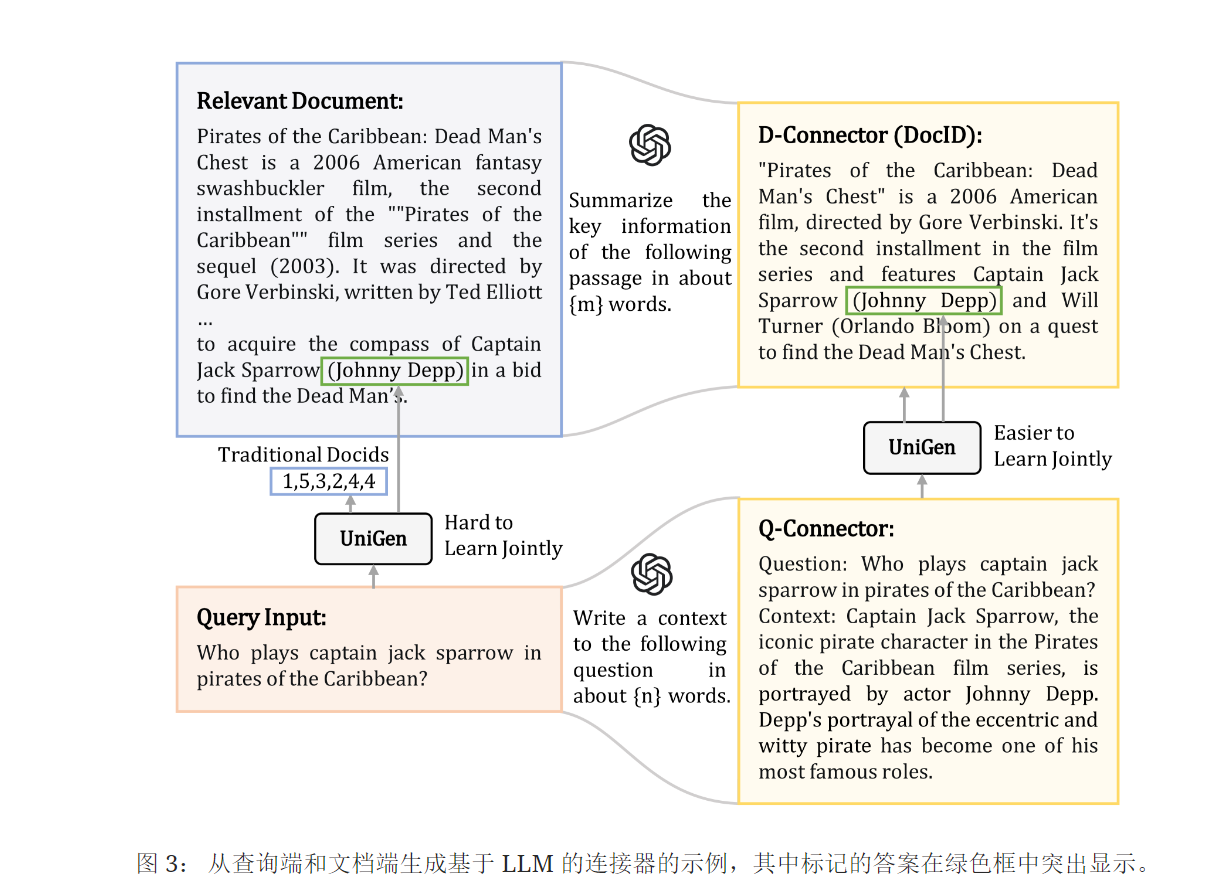
模型的架构包括一个编码器和两个独立的解码器头：一个检索解码器和一个 QA 解码器。编码器将 LLM 生成的增强查询（用 Q-Connector表示）作为输入。检索解码器在前缀树中采用限制集束搜索（Constrained Beam Search）来生成文档标识符的排名列表。这些标识符由 LLM 从文档端生成的连接器（用 D-Connector表示）表示。同时，QA解码器生成答案文本。通过使用检索和 QA 的联合架构，研究人员的模型可以同时优化这两个任务，从而增强整体系统性能。

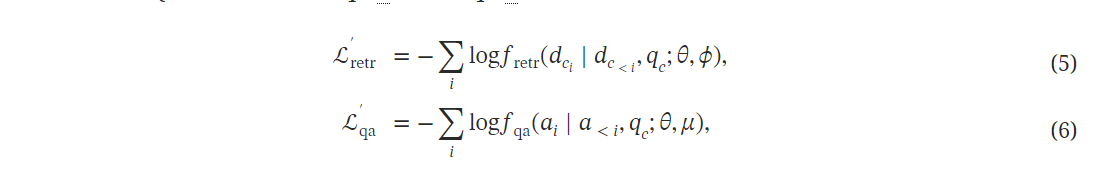
**基于 LLM 的连接器生成**

学习根据查询输入同时生成文档和答案是一项具有挑战性的任务。查询输入通常很短且缺少上下文，而文档很长且包含冗余信息。将查询直接映射到文档和答案是很困难的。此外，现有的 docid 通常表现为无意义的序列，阻碍了生成式检索和 QA 任务的联合学习。为了解决这些问题，研究人员使用 LLM 在查询端和文档端生成 Q-Connector和 D-Connector。这些连接器充当查询输入、文档和答案输出之间的桥梁。图 [3](https://ar5iv.labs.arxiv.org/html/2312.11036#Sx3.F3) 提供了 LLM 生成的连接器的示例。

将“***将以下文档的关键信息汇总到大约 {m} 个字。\n 文档：{d}***”作为LLM输入，输出文档摘要D-Connector称为dc. D-Connector 作为文档的文档标识符（docid），捕获其基本信息，这大大降低了模型记忆长文本的难度。此外，由于答案通常是一个简短的短语或句子，因此使用本文提出的统一框架更容易与QA任务共同学习。

采用“***补充下述问题的上下文，大约为{n}个字\n 问题：{q}***”，作为输入，生成 Q-Connector，称为qc. Q-Connector 提供查询的上下文表示，有助于生成相关的文档和准确的答案。Q-Connector 使模型能够更好地理解查询及其相关上下文，从而使其能够有效地映射到相关文档，并为 QA 任务提供上下文知识。这种方法不依赖于外部语料库，可以为QA取得令人印象深刻的结果。

**生成式检索和QA的联合学习**

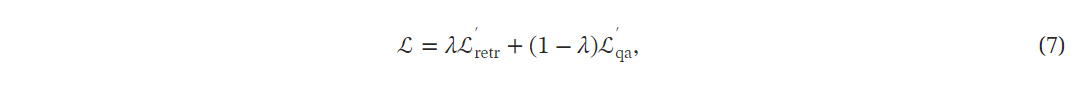
采用 Q-Connector,即qc作为模型输入，团队分别通过 fretr (dc|qc; θ,ф)建立查询q和在集合中𝒟的每个文件d之间的相关性，通过fqa(a|qc;θ,μ)建立查询q和生成答案a的概率之间的联系。这里θ,ф和μ分别表示模型编码器、检索解码器和 QA 解码器的参数。研究人员修改方程（[2](https://ar5iv.labs.arxiv.org/html/2312.11036#Sx3.E2)）和方程（[4](https://ar5iv.labs.arxiv.org/html/2312.11036" \l "Sx3.E4" \o "In Task Formulation ‣ Methodology ‣ UniGen: A Unified Generative Framework for Retrieval and Question Answering with Large Language Models)）中的检索和QA损失，如下所示：

这里的dci和ai分别表示dc和a生成的第i个token

为了给模型配备这两项任务的初始生成功能，研究人员首先在合成训练数据上对其进行训练。先前的研究表明，使用合成数据可以提高生成检索和问答的有效性。因此，研究人员提出了一个两阶段的训练方法，包括预训练阶段和微调阶段：

在预训练阶段，对于每个文档d，使用 DocT5query模型生成K个伪查询qk,（k∈{1,...,K}）.接下来，研究人员送每个伪查询qk及其相应文件d进入大型语言模型 LLaMA-13B-Chat生成标签答案ak.为了模拟由 LLM 生成的Q-Connector，即qc，他们合并（concatenate）qk和d作为生成模型的输入，表示为qk+d。这种方法允许对没用过文档d生成K对的检索<qk+d,dc>和 QA 训练数据<qk+d,ak>.

在微调阶段，研究人员基于被标记的<qc，dc>和<qc,a>数据集继续训练，这里的qc由LLM根据查询生成

为了优化生成检索和 QA 任务的模型， UniGen 框架同时使用了生成检索损失和 QA 损失，分别用方程 （[5](https://ar5iv.labs.arxiv.org/html/2312.11036#Sx3.E5)） 和方程 （[6](https://ar5iv.labs.arxiv.org/html/2312.11036#Sx3.E6)） 表示。共同优化编码器参数θ、检索解码器参数ф和 QA 解码器参数μ，研究人员将这两个损失合并为一个整体损失：

这里λ是正则化权重。通过遵循此训练过程并优化所述的损失函数，该模型可以有效地同时学习检索和 QA 任务。研究人员将这个基础模型称为 UniGen-Base。

**迭代增强策略**

在每次迭代中，研究人员将上一轮的前 k 个文档、答案和查询输入到一个大型语言模型中。目的是生成更高质量的 Q-Connector（qc），并增加模型产生正确答案和检索更多相关文档的可能性。为此，他们使用以下提示：“***给定以下可能相关的文件和可能的正确答案，请用{n}字提供文本的上下文。\n 文档：{d} \n 答案：{a} \n 问题：{q}***”。为了在模型性能和效率之间取得平衡，他们创建了一个名为UniGen-Iter的模型增强版本，它在UniGen-Base之上集成了两次迭代**。**

**实验设置**

**数据**

**MS MARCO 问答**旨在训练和测试系统，这些系统可以有效地生成给定真实世界用户查询的最可能的答案。研究人员使用 QnA （v2.1） 数据集，并从包含标记数据的语料库中提取段落，从而收集了大约 100k 个段落和 94,871 个查询­—答案—相关文档（query-answer-relevant document）训练三元组。

**Natural Questions（NQ）**由从 Google 搜索引擎抽取的问题组成。研究人员将每篇维基百科文章分成 100 个单词的不重叠块。为了确保稳定的评估，他们根据训练集识别语料库中包含标记数据的段落。这个过程产生了大约 100k 个段落和 38,191 个与查询­—答案—相关文档（query-answer-relevant document）训练三元组的多样化集合。

**基线**

对于检索任务，研究人员选择了三类模型。第一类由稀疏检索模型组成，其中包括 BM25和 DocT5Query。第二类包括密集检索模型，例如 DPR和 ANCE。最后，生成检索模型类包括 DSI（、DSI-QG型、NCI和Ultron

关于QA任务，他们考虑了三种类型的基线模型。第一种是一代闭卷模型，以T5和 BAR为代表。第二种类型是检索增强生成模型，其中包含 RAG以及利用 DPR、NCI、Ultron 和 Fusions-in-Decoder 的组合模型。最后一种是基于 LLM 的生成模型，研究人员直接评估 gpt-3.5-turbo-0613 和 LLaMA2-13B-Chat 的 QA 性能。

**评估指标**

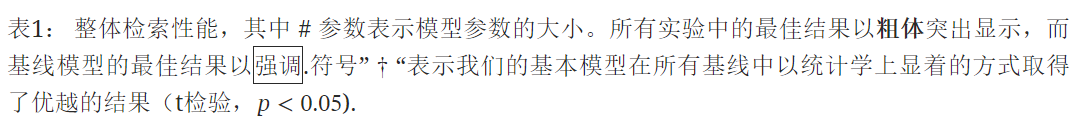
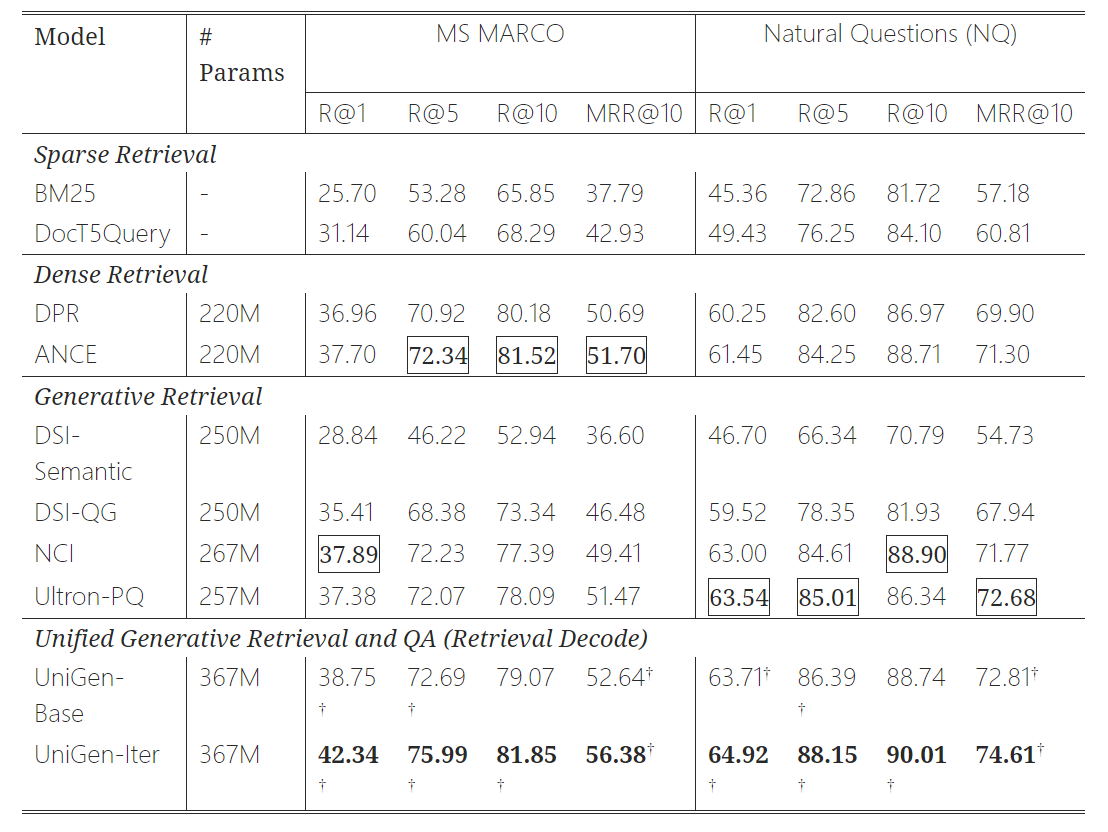
检索模型使用 MRR （平均倒数排名，Mean Reciprocal Rank,MRR，）和召回率（Recall）进行评估，它们分别测量第一个相关文档的平均排名和检索的相关文档的比例。

对于QA评估，研究人员在MS MARCO上使用BLEU-1（B-1）和ROUGE-L（R-L）指标。B-1 测量 uni-gram 重叠，而 R-L 测量最长的公共子序列重叠。在 NQ 数据集上，他们利用精确匹配 （EM） 和 F1 分数，它们分别测量精确匹配以及精确度和召回率的谐波平均值。

**实验细节**

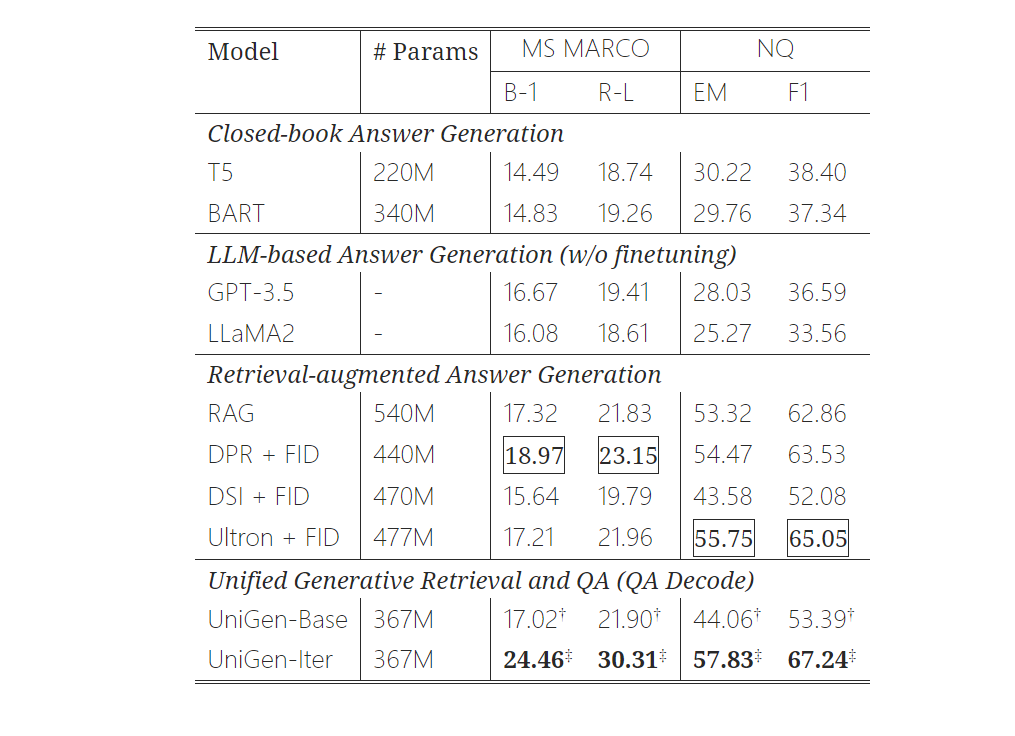
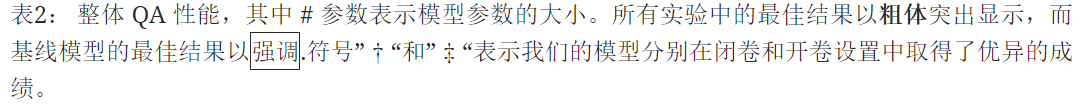
研究人员利用预先训练的 T5 基编码器作为模型的共享编码器。检索解码器和 QA 解码器采用了 T5-base 解码器，其中包含来自 HuggingFace Transformer 的预训练参数。gpt-3.5-turbo-0613 API 作为 LLM 合并到系统中。为了生成训练数据，研究人员为每个文档创建了 10 个伪查询和 10 个伪答案。在训练期间，研究人员将λ（正则化权重）值设置为 0.6。训练过程中一个batch为128、学习率为5e-4 和 2k 个学习率warm-up steps。在推理过程中，采用限制集束搜索进行生成检索解码，并使用贪婪搜索进行 QA 解码。由于内存和时间限制，beam大小限制为最多 10 个。实验在 4 个 NVIDIA RTX 3090 GPU 上进行。

**实验结果**

**检索性能**

（1）UniGen-Base模型在大多数指标方面优于现有的基线模型。对于MRR@10指标，UniGen-Base 在 MS MARCO 和 NQ 数据集上的表现分别优于最佳基线模型 1.81% 和 0.83%。这可以归因于检索和问答任务的联合学习策略，这使得共享编码器更加健壮，缓解了过拟合，并提高了对查询输入的理解。 此外，LLM在查询端和文档端生成的连接器有助于丰富查询的上下文语义，细化语料库文档，从而促进模型学习查询与相关文档的映射关系。

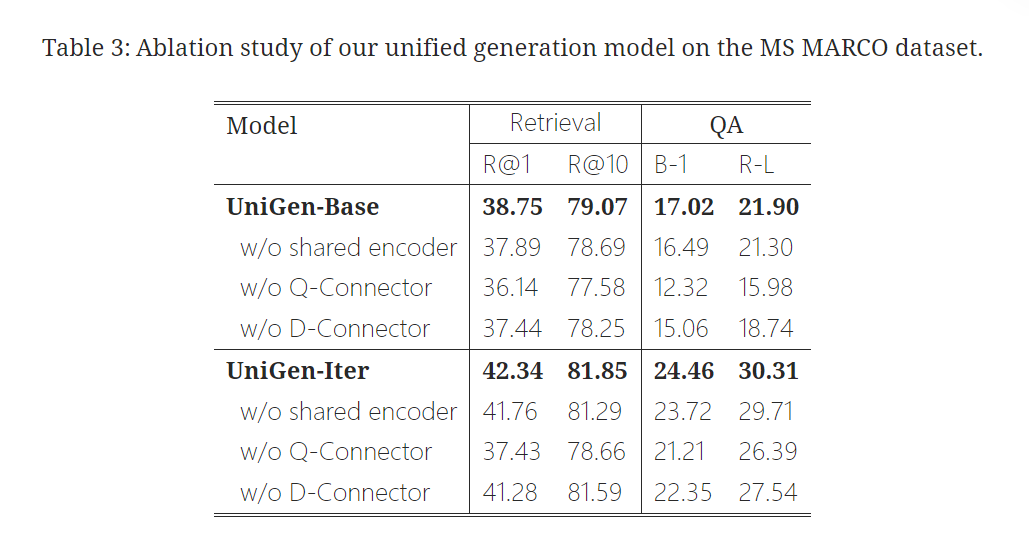
（2）UniGen-Iter模型在两次迭代后取得了最佳的检索性能，优于现有的生成检索模型、密集检索和稀疏检索模型。UniGen-Iter 在R@1方面分别比 MS MARCO 和 NQ 数据集上的最佳基线模型高出 11.76% 和 2.17%。此外，如图 4 中的蓝线所示，当将非迭代方法 （UniGen-Base） 与迭代方法进行比较 1 到 5 次迭代时，可以观察到 MS MARCO 和 NQ 数据集上的MRR@10检索性能持续提高。这清楚地表明了所提出的迭代增强策略在提高检索性能方面的有效性。这是因为先前检索到的文档可以提供相关的外部知识，并且生成的答案也可以作为参考，使 LLM 能够生成更相关的 Q-Connector，并在迭代中不断提高检索性能。

**问答表现**

（1）在闭卷设置下，外部语料库无法访问，模型直接生成输入问题的答案。将使用标记数据微调的小模型与未微调的大模型进行比较，UniGen-Base模型在统计学上明显优于现有基线模型（p<0.05)。对于 MS MARCO 数据集，UniGen-Base 在 Bleu-1 方面比 BART 高出 9.10%，对于 NQ 数据集，它在精确匹配 （EM） 方面比 T5 高出 45.80%。对此可以归因于 LLM 生成的 Q-Connector，它为查询输入提供了有效的上下文信息。此外，答案生成和 D-Connector 的联合学习增强了模型生成答案的鲁棒性。

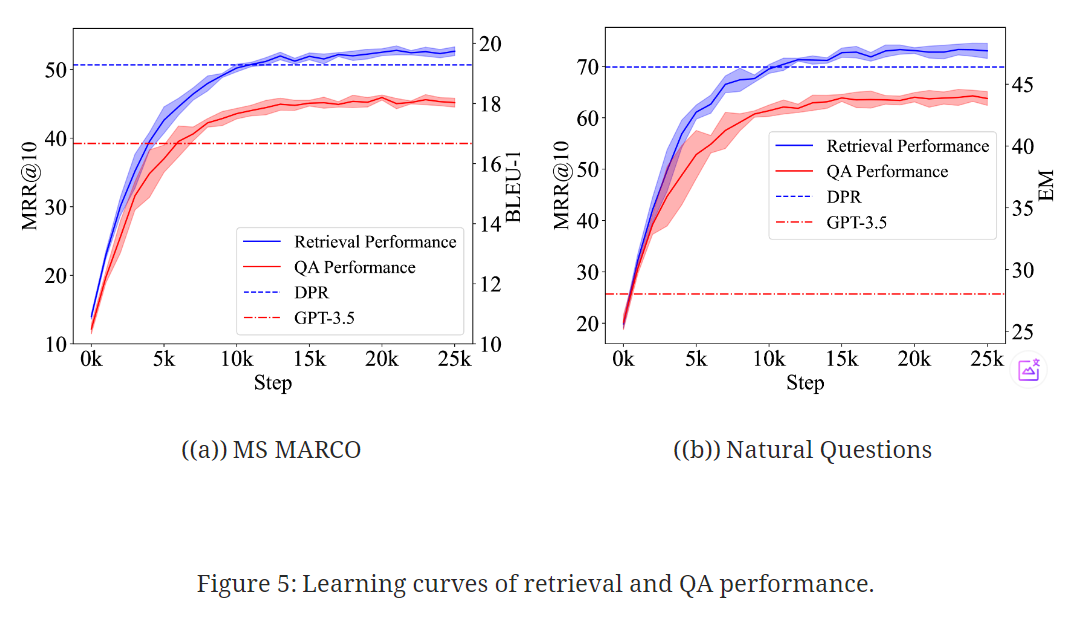
（2）在开卷设置下，与现有的检索增强答案生成模型相比，所提出的UniGen-Iter在MS MARCO数据集上的Bleu-1性能优于DPR+FID模型28.94%，在NQ数据集上的EM性能优于Ultron+FID3.73%。 此外，图 4 说明了与非迭代方法（UniGen-Base）相比，通过使用迭代方法提高了 QA 性能，如红线所示。这也凸显了所提出的迭代增强策略在提高QA绩效方面的有效性。因此，增强的 Q 连接器也有助于在各种迭代中持续增强 QA 任务的性能。

**消融研究**

研究人员系统地删除了每个模块并观察由此产生的性能下降

移除共享编码器、Q-Connector 或 D-Connector中任何模块，都会导致检索和 QA 性能明显下降。值得注意的是，当移除 Q-Connector时，性能下降幅度最大。这凸显了利用大规模语言模型作为外部知识源来提供与查询相关的上下文的重要性。 此外，移除 D-Connector也会对最终性能产生重大影响。这表明 D-Connector 在弥合文档和答案之间的差距方面做出了贡献，超越了传统方法（例如基于分层聚类的文档标识符）的功能。 对于不使用共享编码器的方法，研究人员仍然观察到性能下降，这凸显了研究人员统一结构的优势。这种结构可以训练更强大的编码器，从而改进输入的表示，并增强检索和 QA 性能。

**学习曲线研究**

研究人员绘制了学习曲线，以展示 MS MARCO 和 NQ 数据集上的检索和 QA 性能。他们利用合成和标记数据的组合来训练 UniGen 模型。图 5 说明了这些曲线，并绘制了每个指标的平均值和标准偏差，这些平均值和标准差是从每个数据集上的五次单独训练运行中获得的

用MRR@10衡量的检索性能由蓝色曲线表示，而红色曲线表示 QA 性能，研究人员在MS MARCO上使用BLEU-1（B-1）和ROUGE-L（R-L）指标。值得注意的是，这两个任务在整个学习过程中都表现出稳定的优化，从而证实了其提出的同时学习检索和QA任务的统一框架是有效的。

**结论**

在本文中，研究人员介绍了UniGen，这是一个用于检索和问答的统一生成框架。他们的方法同时优化了这两个任务，并利用大型语言模型生成的连接器在输入-输出和文档-答案空间中建立语义连接。此外，他们的迭代增强方法被证明在提高检索和 QA 性能方面是有效的。通过对公共数据集进行的大量实验，他们证明了UniGen在检索和QA任务中的有效性。这项工作为共同学习、检索和其他生成任务开辟了新的可能性。