RETRIEVAL IS ACCURATE GENERATION

太长不看版

介绍了一种新的语言建模方法，该方法直接从一组支持文档中选择与上下文相关的短语；提出了一种将文本生成分解为顺序的下一个短语检索的新方法，该方法由语言学驱动的启发式和迭代自我强化的引导支持；在各种下游任务上验证了模型的有效性，包括开放域和特定域的问答以及开放式文本生成，突出了与标准 LMs 和几个检索增强基线相比的显著改进

摘要

标准语言模型通过从**固定、有限和独立的词汇表**中选择 tokens 来生成文本。本文介绍了一种新的方法，**从一组支持文档中选择上下文感知到的短语**。这种范式转变最重要的挑战之一是**确定训练预设答案(training oracles)**，因为一串文本可以被分割成各种片段，并且每个片段可以从许多可能的文档中检索。为了解决这个问题，我们提出使用语言学启发式来初始化训练 oracle，更重要的是，通过迭代自增强来引导优化预设答案。

大量的实验表明，模型不仅在各种知识密集型任务上优于标准语言模型，而且在开放式文本生成中表现出更高的生成质量。例如，与标准语言模型相比，本文的模型将 OpenbookQA 的准确率从 23.47%提高到 36.27%，并将开放式文本生成的 MAUVE 分数从 42.61%提高到 81.58%。值得注意的是，模型在几个检索增强的基准实验中也实现了最佳性能和最低延迟。总之，我们断言，**检索是更准确的生成**，并鼓励进一步研究这种新的范式转变。

1. 引言

本文讨论了语言模型（LMs）在处理文本生成时的新范式：通过直接检索短语而非仅从固定、有限的词汇表中逐个选择单词或子词来生成文本。Lan 等人（2023）提出了一种名为 **CoG** 的方法，该方法从相似上下文中检索短语，其中“短语”指的是任何连续的可变长度文本段。值得注意的是，CoG 与其他检索增强的生成框架类似，采用了**两阶段流程**，即**先文档检索后基于文档的短语提取，最终性能受限于第一阶段返回的质量和数量**。而本文提出了一种全新的范式，完全去除了对文档检索的依赖，这是第一次通过直接检索短语进行文本生成。

采用这种新方法的一个核心挑战是训练预设答案（oracles）的构建，即将文本字符串映射到创建训练示例的动作序列的功能。对于给定文本，有许多不同的方式将其分段成短语，每个潜在短语都可以从大量文档中检索到。为了更好地对齐生成过程和支持文档，采用了双重方法：首先，利用**基于语言学的启发式方法初始化训练预设答案（training oracles）**；其次，通过迭代自我强化的引导机制逐渐细化指导。

模型评估覆盖了广泛的知识密集型任务，例如开放域问答，展现了优越的零样本性能，超越了基线方法。例如，在 OpenbookQA 数据集上，模型将准确率从 23.47%提高到 36.27%，在开放式文本生成中的 MAUVE 得分也有显著提高。此外，当切换到更大或特定领域的短语表时，模型表现得更好，无需进一步训练。模型还在检索增强基准实验中实现了最快的生成速度。

2. 生成与检索的统一视角

标准语言模型（LM）将一个序列 $X = [x_1, x_2, \dots, x_n]$ 的生成概率分解为一

系列条件概率$p(x)=\prod_{i=1}^{n} p(x_i|X_{<i})$。因此，生成过程通常是根据迄今为止生成的序列反复预测下一个标记(i.e., prefix)。下一个标记预测概率的计算公式为：

$$p(x_i|X_{<i})=\frac{\exp\left(E_p(X_{<i})\cdot E_c(x_i)\right)}{\sum_{x'\in V}\exp\left(E_p(X_{<i})\cdot E_c(x')\right)}$$

其中，$\boldsymbol{E_p}(X_{<i})$是前缀 $X_{<i}$的向量表示，$\boldsymbol{E_c}(x)$表示标记x的向量表示，V代表标记词汇。通过上述符号，我们可以将标准 LM 视为**连接不同前置词和标记的双编码器匹配网络**。通常情况下，如图 1 左侧所示:

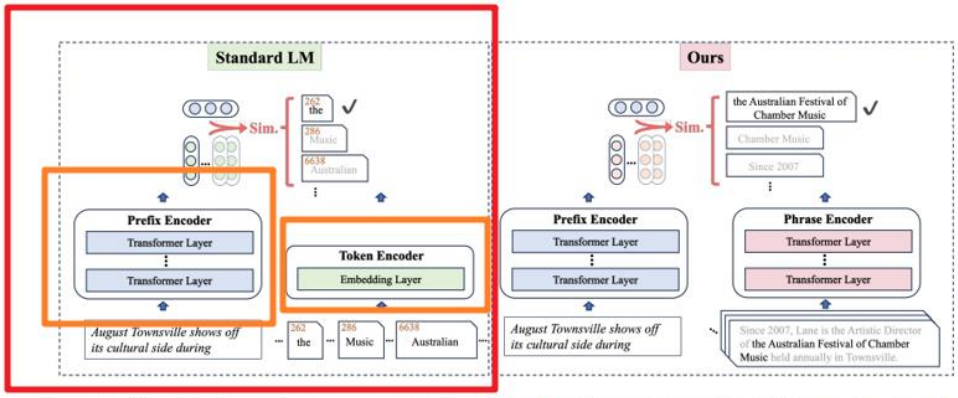


Figure 1: Comparison between our method and standard language models. Both can be viewed as dual-encoder matching networks connecting source prefixes and target continuations. On the target side, standard language models employ an immediate embedding layer for target tokens from a fixed, finite, and standalone vocabulary. In contrast, our methods uses an expressive phrase encoder for target phrase from an editable, extensible, and contextualized phrase table.

源编码器 Ep 由多层神经网络（如 Transformers）实现，而目标编码器 Ec 只是一个标记嵌入层(embedding layer)。由此可见，双编码器网络的设计严重失衡；**源侧比目标侧复杂得多。**

最近，提出了一种增强检索的语言模型，名为 CoG（Lan 等人，2023 年）。CoG 不仅进行标记选择，还允许从一系列支持文档中检索短语（即，可变长度的 n-gram）。CoG 增强了传统 LMs 的目标端:首先，候选池扩大以包括可变长度的短语。其次，目标编码器不仅考虑候选本身，还考虑其上下文。

然而，从大规模语料库中搜索短语是非常消耗资源的。因此，CoG 采用了两阶段搜索策略：**首先检索相关文档以缩小短语选择的搜索范围**。为了构建训练指导(traing oracles)，CoG 使用前向最大匹配算法从检索到的文档中找出最长的匹配短语。尽管结果令人充满希望，CoG **无法保证提供全局最优的短语检索解决方案，而且高度依赖外部文档检索工具**。相比之下，我们提出了一个新范式，通过**直接检索短语来生成文本。**

3. 可行性方法

3.1. 概述

研究旨在通过从**标记生成过渡到短语检索**，提高语言模型（LM）的可解释性和事实性。首先，短语的语义可通过其周围的语境得到增强，从而为推理提供更具区分性的表征。其次，每个检索到的短语都可以追溯到其原始文档，从而增强了输出的可靠性。为了将给定的 prefix(前缀)与一组长度可变的短语联系起来，模型沿用了第 2 节中描述的双编码器结构，但与偏重于源侧的标准 LM 相比，强调均衡设计（见图 1 右侧）。

具体来说，源编码器$Ep(\cdot)$是一个多层神经网络（如 Transformer）。目标编码器 $Ec(\cdot)$也是一个多层神经网络，用于**学习支持文档中短语的上下文感知表征**。

与标准 LM 类似，使用点积作为匹配度量。在推理过程中，可以使用高效的最大内积搜索（MIPS）算法。

3.2. 训练预设答案(training oracles)

将文本生成分解为一系列下一词组检索(next-phrase retrieval)。形式上，每一步都以当前的前缀词 p 作为其状态，一个预设答案策略π^*(oracle policy)，π^*将状态映射为行动 $\pi^*(p) \rightarrow (f, s)$，其中 f 是后续短语，s 是短语 f 在辅助文档中的副本(复制粘贴)。

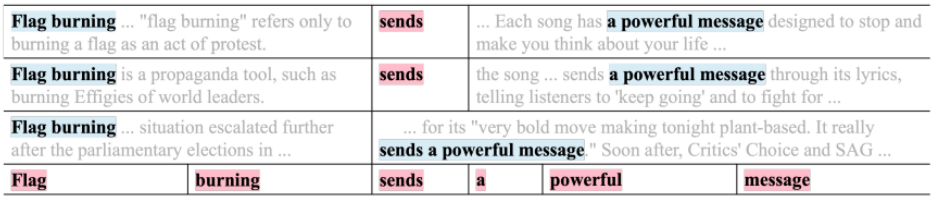


Figure 2: Four possible generation paths for the sentence “Flag burning sends a powerful message”. Content highlighted in blue (red) are phrases retrieved from supporting documents (from the token vocabulary). Standard LMs can be viewed as only considering the generation path at the bottom.

如图 2 所示(蓝色（红色）标出的内容是从辅助文档（标记词汇）中检索到的短语。标准 LM 只考虑底部的生成路径)，从原始语料库中创建这样的三元组（p,f,s）有两个难题：

首先，短语 f 的边界不明确，因为续篇可以有多种划分方式。

其次，每个短语 s 的来源也不清楚，因为一个短语可以在大量文件中出现无数次。在大量文件中多次出现。另一方面，给定文本的生成路径的多样性也表明要使模型达到最佳的快速收敛效果，训练预设答案(oracles)至关重要。

为了解决上述问题，首先提出了一套以语言学为基础的启发式方法来初始化训练预设答案，然后提出如何让模型在自我强化过程中重新指定其生成路径。

a) 语言学启发式方法

训练规则如下：

句法结构：受到语言的句法结构及其对语言生成影响的启发，限制短语是句法解析树中对应于构成单位的连续单词序列。这种方法确保每个短语都具有相对完整且定义良好的意义，同时避免了可能导致语义模糊或无意义组合的任意单词组合。

分布式稀疏性：包含高频短语显著增加了候选池的大小。由于将不同上下文中词汇上相同的短语视为池中的不同条目，因此一个高频短语可能会引入成千上万甚至百万的条目。在分析维基百科时，发现仅仅去掉最高频的 1%的短语就可以减少 50%的条目数量。然而，这些如"as well as"这样的高频短语通常缺乏具体含义，可能导致训练不平衡，进而不利于模型的整体表现。对于极低频率的短语，它们是罕见用法，实际用途有限。包含这些短语会显著增加训练的复杂性，因此选择排除它们。

语义相似性：虽然在不同位置可以找到词汇上相同的短语副本，但考虑到词义多样性是至关重要的，因为词汇上相同的短语在不同上下文中可能显示出不同的意义。此外，即使词汇上相同的短语在不同的上下文中有相似的意义，也可能因上下文的细微差别而产生微妙的差异，因此在选择最合适的匹配时需要对语义相似性进行彻底评估。

具体来说，首先运行斯坦福解析器(Stanford Parser)从训练数据中提取构成成

分。然后根据以下标准过滤这些构成成分：

（1）移除带有如 WHADJP, WHADVP 等标签的琐碎成分；

（2）排除太短（< 2 个单词）或太长（> 10 个单词）的成分；

（3）丢弃逆文档频率（IDF）值过高或过低的成分。

值得注意的是，对较长的成分应使用更宽松的 IDF 阈值。将词汇上相同的短语分组，并使用 BM25 和现成的短语编码器计算成对语义相似性。因此可以基于得分确定每个前缀的最合适的下一个短语。

附录 A 相关内容：

句法分析是 NLP 中研究得非常深入的一项任务，例如，Universal Dependencies 为 100 多种语言提供一致的语法注释。众所皆知，解析准确率在英语、汉语、意大利语、日语、葡萄牙语等主要语言中都相当高。尽管如此，我们预计当解析器准确率相对较低时，语言和领域的性能会有所下降。

在无法使用句法分析器的情况下可采用其他方法，如无监督句法分析和无监督标记化方法（如 BPE、sentence piece）。接下来根据上面提到的标准过滤构成成分。然后，使用现成的短语编码器计算原始短语与检索到的候选短语之间的语义相似性。因此，可以根据得分为每个 prefix 确定最合适的下一个短语。整个预处理过程包括句法分析、短语选择和语义匹配，在 8 个 V100 GPU 上大约需要 24 小时。与训练模型的成本相比微不足道。

b) 迭代自我强化

上述启发式方法确定的生成路径与模型无关，可能存在噪声和次优。为了进一步提高性能，允许模型根据自身能力调整生成路径。也就是说，从模仿预设答案(oracles)过渡到强化自身偏好。

具体来说，提出了一种引导算法来迭代调整目标短语。对于每个 prefix p，首先让模型使用其当前策略检索**整个候选池中的 k 个最佳短语**。然后，从这 k 个短语中**选择语义匹配得分最高的有效短语作为新的目标短语**。如果找不到这样的短语，即没有 k 个最佳短语与地面实况的延续相匹配，我们就保留之前的目标。上述过程会定期重复。

附录 B 中举例说明：

假设有一个 prefix p =" Go right for the top when you"。这个前置词的基本事实是 " Go right for the top when you want to make things happen"。

最初确定的目标短语是”want”，在迭代自我强化过程中，首先让模型从整个候选词库中检索出 k 个最佳短语。假设 k 个最佳短语是["want"、"want to"、"want to make things happen"、"need"、"can"]，那么只有 "want"、"want to "和 "want to make things happen "被视为有效短语。如果模型语义匹配得分最高的是 "want to make things happen"，我们就会将 prefix 的目标短语更新为该短语。如果 k 个最佳短语都无效，我们将保留之前的目标短语 "want"。

3.3. 训练目标

使用 InfoNCE 损失来优化模型，为此为每个三元组(p, f, s)引入一个否定短语集 $N(p)$。

$$L_p = \frac{\exp\left(E_p(p) \cdot E_c(s)\right)}{\exp\left(E_p(p) \cdot E_c(s)\right) + \sum_{t \in N(p)} \exp\left(E_p(p) \cdot E_c(t)\right)}$$

为了保留标记级生成能力，还使用标准的下一个标记预测损失L_t来训练我们

的模型。训练目标为$L_p + \alpha L_t$.。

否定短语集 $N(p)$ 的构建：

为了提高模型区分短语的能力，加入了两种类型的负面示例：

(1) 批量内负面示例(In-batch negatives)：将同一训练批次中的所有其他候选短语都视为这类负面示例。这些负面例子可以帮助模型大规模地学习更具区分性的表征，而不会产生大量成本。

(2) 硬否定(Hard negatives)：回顾迭代自我强化的过程，通过检索每个 prefix 的前 k 个候选短语来定期更新生成目标。在这 k个词组中，尽管有一个词组可能被选为新的生成目标，但其余的词组可以作为强否定词组，因为它们很可能会混淆模型。 注意:上述否定词组可能包含假否定词组，这些词组虽然没有被选为目标，但仍然是有效的后续词组。为了将风险降到最低，会删除所有构成基本事实(ground truth)延续前置词的短语。

3.4. 模型

a)前缀编码器(Preffx Encoder)

将 prefix 视为一个词组序列，并将先前预测的短语拆分为标记。这个标记序列使用标准的 Transformer 架构进行编码，并带有因果关系。prefix 表示是通过序列中最后一个标记的最后一层表示的线性投影。

b)短语编码器

采用了深度双向变换器来生成支持文档的上下文化标记表示。短语的表征是通过连接其第一个和最后一个标记的表征而获得的，然后将连接后的表征投射到支持文档中。将连接后的表示投影到与 prefix 表示相同的维度。

为了保持使用单个标记来组成输出的能力，还将标记词汇添加到短语表中。这些独立的标记可视为特殊短语，它们的表示通过 LM 的标准嵌入层获得。

4. 实验设置

4.1. 实施细节

在 MiniPile 的训练集上训练模型，并使用英文维基百科作为辅助文档。具体来说，将每篇维基百科文章分割成多个不相连的文本块，最多可分割 128 个单词作为文档，这样就有 29,488,431 个文档。短语索引的大小为 137,101,097 个。同时使用 GPT-2 和 DensePhrases 来分别初始化前缀编码器和短语编码器。为了提高效率，只对前置词编码器进行了微调。这样就避免了重新计算与更新短语编码器相关的短语嵌入。而在通过自我强化修改训练字典的同时，会检索每个前缀的前$k = 128$个短语。

4.2. 推理细节

在推理过程中，我们采用 FAISS（向量相似性搜索和聚类库）进行高效检索。

文本生成：直接从整个短语表（包括上下文感知的短语和独立的标记）中检索前 k 个候选项。接着，对这些候选项的匹配得分应用一个 softmax 函数，创建下一个短语的概率分布，并使用 top-p 采样方法选择下一个短语。在所有实验中，将 k 设置为 128，并将 p 设置为 0.95。为了控制短语检索的比例，过滤掉概率低于阈值的短语。如果没有特别指明，阈值 ϕ 设置为 0.4。

附录 G（消融实验）中的表 7：

k	TruthfulQA	OpenbookQA	ARC-Challenge	MedMCQA	Med-USMILE	Avg.
1	32.74	**36.80**	27.94	**29.95**	25.68	30.62
2	32.88	36.80	28.04	29.90	25.68	30.66
4	33.29	36.80	27.84	29.84	25.68	30.69
8	33.42	36.80	27.84	29.76	25.42	30.65
16	34.25	36.80	27.64	29.61	25.15	30.69
32	34.11	36.27	27.64	29.50	26.12	30.73
48	**34.38**	36.27	28.04	29.27	**26.21**	**30.83**
64	33.84	36.53	**28.34**	29.38	25.59	30.74
128	34.27	36.27	28.24	29.44	25.69	30.78
256	33.42	36.27	27.37	29.24	24.80	30.22
512	32.88	35.73	27.64	29.33	25.68	30.25
768	32.47	35.47	27.74	29.67	25.42	30.15
1024	32.47	35.47	27.54	29.61	24.89	30.00

Table 7: Ablation studies on the impact of k on knowledge-intensive tasks.

如表 7 所示，k 对我们的模型在知识密集型任务中的表现没有显著影响。由于在自我强化过程中会检索前 128 个短语，因此在所有实验中都设置 $k = 128$。

计算给定文本的可能性：通过求和所有可能的生成路径来近似估计可能性。例如，对于句子“The Moon rises”，可能存在以下生成路径：

(1) The→moon→rises；

(2) The moon→rises；

(3) The moon rises。

每条路径的概率是该路径上所有短语（标记）概率的乘积。例如，路径(2)的概率由$p(rises|Themoon) \cdot p(Themoon)$计算得出。每一步的概率的获得方式与我们构建续写生成中的下一短语概率分布的方式相同。注意，所有可能路径的总和可以通过时间复杂度为$O(n^2)$的动态规划高效计算，其中n代表文本中的标记数量。

4.3. 基准

本文将提出的方法与在零样本设置中的标准语言模型进行了比较，并将以下最新的检索增强方法作为基线进行了对照：

基础 LM(Base LM):使用 Transformer 架构的标准令牌级语言模型。对预训练的 GPT-2 进行了微调。

kNN-LM:一种检索增强型语言模型，将基础 LM 的下一个标记分布与 k 最近邻（kNN）模型进行插值。

RETRO:一种与预训练文档检索器、文档编码器和交叉注意力机制相结合的检索增强型语言模型。

CoG:另一种检索增强型语言模型，采用了两阶段搜索流程。它首先检索语义相关的文档，然后考虑其中的所有 n-gram 作为候选短语。

5. 实验

5.1. 知识密集型任务

a) 数据集

使用了五个知识不敏感(knowledge-insensitive)的数据集，包括三个开放域问答数据集：OpenbookQA、ARC-Challenge 和 TruthfulQA，以及两个特定领域（医学）数据集：MedMCQA 和 MedUSMILE。

根据以往的研究，采用了一个带选项的分类方法来量化模型的性能。这种方法包括向模型提供一系列选项，并计算每个选项是正确答案的可能性。概率最高的选项被选为模型的预测，得出模型预测的准确度。

b) 结果

如表 1 所示：

	TruthfulQA	OpenbookQA	ARC-Challenge	MedMCQA	Med-USMILE
Base LM (w/o FT)	30.27	22.67	24.52	27.96	24.89
Base LM	29.73	23.47	23.92	28.33	24.19
*k*NN-LM	30.27	22.93	24.82	27.96	24.72
RETRO	27.53	26.13	22.21	25.68	25.33
CoG	34.11	35.47	27.24	29.07	25.07
Ours	**34.27**	**36.27**	**28.27**	**29.44**	**25.69**
Ours (w/o phrase)	28.63	23.73	22.51	27.42	24.80

Table 1: Experiments on knowledge-intensive tasks. Ours (w/o phrase): a variant of our model that restricts the model to only use standalone tokens without retrieving context-aware phrases.

模型在所有数据集上一致地超过了各种基线模型。与基础语言模型（base LM）相比，模型在 TruthfulQA 和 OpenBookQA 数据集上的准确率分别从 29.73%提高到 34.27%和从 23.47%提高到 36.27%。

当从模型中移除短语检索，仅使用独立的标记时（模型不含短语检索），性能有明显下降，这证明了在方法中加入短语检索的有效性。请注意，表 1 中展示的模型是从预训练的语言模型初始化的。为了分析预训练模型在框架中的作用，我们从头开始训练所有模型，使用随机初始化。结果显示在附录 G 的表 8 中：

	TruthfulQA	OpenbookQA	ARC-Challenge	MedMCQA	Med-USMILE
Base LM	30.14	22.40	22.41	28.27	23.58
*k*NN-LM	30.14	22.40	23.32	27.99	23.14
COG	32.88	34.13	25.13	29.16	25.15
Ours	**33.29**	**35.20**	**27.04**	**30.24**	**26.21**
Ours (w/o phrase)	28.22	21.87	23.02	27.99	24.89

Table 8: The results of models trained from scratch.

模型在所有数据集上超过了基线。例如，在 OpenbookQA 上比基础语言模型高出 12.8%的绝对改进，这表明我们的训练框架并不严重依赖于预训练模型。为了阐明**短语检索在知识密集型任务中的作用**，深入研究了**附录 D** 中描述的案例研究：

为了阐明短语检索在知识密集型任务中的作用，深入研究了图 3 中描述的一个案例研究（具体见补充说明）：

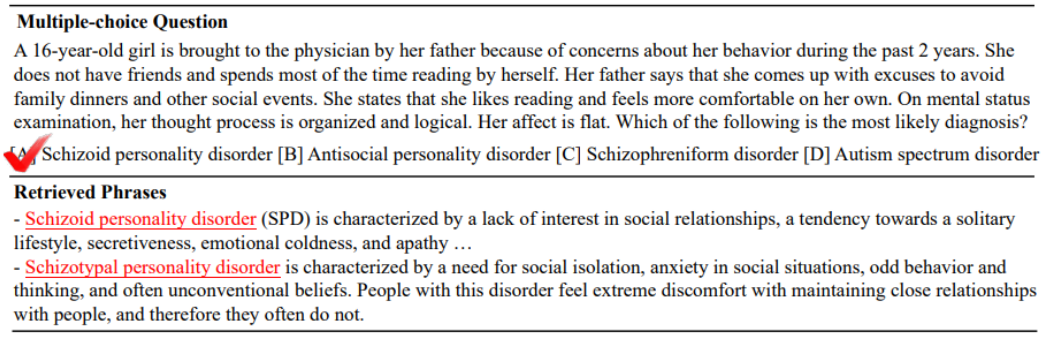


Figure 3: An illustrative example from Med-USMILE: The two highlighted phrases in red are retrieved in response to the posed question.

如第 4.2 节先前讨论的，方法涉及为每个选项中的标记检索短语，这使我们能够估计除了简单生成标记序列之外的替代生成路径的概率。在 MedUSMILE 数据集的这个特定案例中，选项是通过将问题与每个候选答案连接起来形成的。我们发现，对于问题最后一个标记检索到的短语包含了答案，这是一个需要医学知识才能理解的适当名词。这引入了一个新的生成路径：问题→分裂型人格障碍。我们观察到，检索到的短语的上下文，例如“分裂型人格障碍（SPD）的特点是对社会关系缺乏兴趣...”，与问题的上下文“她没有朋友，大部分时间都一个人读书...”紧密对齐。这些上下文编码的短语有助于答案选择，从而展示了模型的可解释性。它还突出了模型利用上下文信息的能力，特别是在需要专业知识的任务中。

排除了 IDF 值过高或过低的短语这个策略不仅稳定了训练过程，还提高了训练效率。然而，最初被过滤掉的短语可以重新利用，以无需训练的方式扩大短语索引。这个扩展后的短语索引现在是原来的三倍大，突出了方法的可扩展性。如表 2 所示：

	TruthfulQA	OpenbookQA	ARC-Challenge	MedMCQA	Med-USMILE
Ours	34.27	36.27	**28.27**	29.44	25.69
w/ enlarged index	**39.59**	**37.07**	27.14	**31.63**	**27.87**

Table 2: Results for our model with an enlarged phrase index.

这种扩展提升了模型的性能，例如在 TruthfulQA 上准确率增加了 5.32%。这不仅凸显了模型泛化到未见短语和文档的潜力，还强调了它的即插即用特性，能够适应更大的短语表而无需重新训练。

通过构建特定领域的短语索引，可以提高模型在特定领域问答任务的表现，而无需特定领域的训练。为此创建了一个包含 300 万个短语的索引，这些短语是从医学领域的小型文本集中提取的。为了进行公平比较，还对基础语言模型（base LM）进行了微调。

	MedMCQA	Med-USMILE
Base LM (FT)	28.79	25.15
General index	29.44	25.69
Medical index	**29.50**	**26.38**
w/o phrase	27.42	24.80

Table 3: Results on medical datasets.

表 3 显示，尽管与原始维基百科索引相比索引大小大幅减小（300 万对比

13700 万），模型在两个医学问答数据集上的表现甚至更好。这结果突出了模型通过利用特定领域的、精心策划的短语索引，在无需训练的情况下提升特定领域性能的能力。

5.2. 开放式文本生成

a) 评价指标

使用了三种自动评估指标来衡量生成文本的质量：

(i) MAUVE:衡量生成文本的整体效用，通过估算内容的平均效用；

(ii) 一致性(coherence):衡量生成文本的逻辑连贯性和流畅性:确保输出内容结构良好易于理解；

(iii) 多样性(diversity):评估生成内容的多样化，促进生成独特和有创造性的文本。

以百分比的形式报告 MAUVE 和多样性指标。这些指标的详细信息可以在附录 E 中找到。同时，还测量了模型解码一个由 128 个标记组成的续写内容所需的平均时间成本，这称为延迟。

附录 E:

MAUVE (Pillutla et al., 2021) measures how closely the token distribution in generated text matches that in human-written text across the entire test set.

Coherence (Su & Collier, 2022; Su et al., 2022) measures the semantic coherence between the prompt x and the generated text $\hat{x}$ by calculating the average log-likelihood as: $\text{coherence}(\hat{x}; x) = \frac{1}{|\hat{x}|}\sum_{i=1}^{|\hat{x}|} \log p_M(\hat{x}_i|[x : \hat{x}_{<i}])$, where [:] is the concatenation operation and M is a pre-trained LM. We follow prior work and set M as the OPT-2.7B model (Zhang et al., 2022). In our implementation, we introduce a slight modification by taking the negative of the average log-likelihood. This adjustment transforms the typically negative log-likelihood into a positive value, facilitating a more intuitive interpretation of the results.

Diversity (Welleck et al., 2020; Su et al., 2022; Lan et al., 2023) measures the repetition in generated text at different n-gram levels by computing the proportion o[跳转到页面: 11]ns to total n-grams in the generated text. It is defined as: $\text{diversity} = \prod_{n=2}^{4}(1.0 - \frac{\text{rep-n}}{100})$, where $\text{rep-n} = 100 \times (1.0 - \frac{|\text{unique n-grams}(\hat{x})|}{|\text{total n-grams}(\hat{x})|})$, and $\hat{x}$ is the text generated by the model.

b) 结果

	MAUVE↑	Coherence↓	Diversity↑	Latency↓
Base LM (w/o FT)	69.68	3.64	83.14	1.00x
Base LM	42.61	3.56	78.72	1.00x
kNN-LM	13.07	5.63	**88.10**	6.29x
RETRO	62.39	4.82	80.96	1.51x
CoG	52.27	**2.08**	55.04	4.40x
Ours	**81.58**	3.25	76.26	**1.29x**

Table 4: Results for open-ended text generation.

在表 4 中，模型获得了所有模型中最高的 MAUVE 得分，这说明了生成文本的高质量。与基线语言模型相比，其他使用检索增强的方法由于文本退化，在 MAUVE 得分上表现不佳。模型在连贯性和多样性之间展现了很好的平衡，连贯性得分为 3.25，除了 CoG 模型外，优于大多数基准。但 CoG 模型生成的句子词汇上相似且无意义，多样性得分较低（55.04%）。而本模型的多样性得分为 76.26%，虽然略低于某些基准模型，但那些模型通常生成不连贯的句子。

人类评价(Human Evaluation)：随机抽取了 100 个案例，从流畅性、连贯性、信息量和语法四个角度评估基线语言模型、未经微调的基线模型和本模

型。每个方面都用 1 到 4 的 Likert 量表评分。

Model	Fluency	Coherence	Informativeness	Grammar
Base LM (w/o FT)	2.91	2.33	2.35	3.00
Base LM	2.81	2.37	2.40	2.79
Ours	**2.95**	**2.70**	**2.67**	**3.02**

Table 5: Human evaluation results.

表 5 报告了平均分数，结果显示本模型在所有四个类别上都优于基线语言模型，特别是在连贯性和信息量方面，说明模型基于短语检索，在遵循前文上下文和提供更多信息内容方面表现更好。至于基线模型的低分主要是由于格式问题。

不同模型的生成速度(Generation Speed):表 4 报告了相对延迟时间，以基线语言模型为基准。kNN-LM 由于需要将基线模型的标记分布与使用其数据存储计算的另一个分布进行插值而产生最高的成本。CoG 模型也显示出显著的开销，因为它涉及提取检索到的文档中的所有 n-grams，并对标记和所有 n-grams 进行 softmax 运算及抽样。RETRO 模型虽然比前两个快，但仍需要时间来应用检索到的文本块的表示在注意力计算中。本文方法在生成速度上表现突出，因为它直接检索并使用短语。

自我强化（SR）机制的效果:对于知识密集型任务，SR 机制对模型性能的影响并不显著（参见附录 G 的表 9）。

	TruthfulQA	OpenbookQA	ARC-Challenge	MedMCQA	Med-USMILE
w/o SR	34.11	**37.07**	27.14	**30.32**	**25.85**
round1	33.97	36.80	27.34	29.84	25.77
round2	**34.27**	36.27	**28.24**	29.44	25.69

Table 9: Ablation studies on the effect of self-reinforcement.

这表明，即便没有 SR 机制的帮助，该框架本身在处理这类任务时就已经很有效。然而，在开放式文本生成任务中，情况则不同。

	MAUVE↑	Coh.↓	Div.↑
w/o SR	7.86	4.14	**81.14**
round1	64.49	**3.23**	70.15
round2	**81.58**	3.25	76.26

Table 6: Ablation study on the effect of self-reinforcement.

表 6 显示，经过 SR 训练的模型在多轮评估中 MAUVE 分数有显著提升，这表明 SR 在提高文本生成质量方面的重要性。经过第二轮之后，随着 SR 迭代轮数的增加，模型没有观察到明显的改进，这表明模型已趋于其最优状态。

6. 相关工作

本文探讨了标准的语言模型（LMs）在面对自然语言处理任务时，如何通过**预测文本前缀后的下一个标记来展现出强大的零样本泛化能力（zero-shot performance）**。但是，增加模型参数和训练语料的规模成本高昂，且耗时。为应对这些问题，越来越多的工作通过引入非参数组件来增强参数化的语言模型。使用检索技术获得相关文档来指导下一个标记的预测，以及利用非参数最近邻估计来增强输出概率分布的方法被提出。

此外，检索然后生成的范式在特定的下游任务中得到了广泛研究，比如代码生成、问题回答、开放域对话系统、机器翻译和多模态检索等。

与此工作最相关的是 Min 等人（2022 年）和 Lan 等人（2023 年）的研究。Min 等人探索了在遮蔽语言模型中提升自然语言理解的相似理念。而 Lan 等人则允许从支撑文档中复制短语，但他们的方法仍然依赖于一个两阶段流程，仅将生成基于一小组检索到的文档上。尽管 Lan 等人简单地使用最长公共子序列算法来找出可以从检索文档中复制的短语，本文则提出了基于启发式和自我加强机制来构建可靠的训练指导。此外，Lan 等人的评估仅限于开放式文本生成任务。

7. 结论

文章提出了一种使用上下文感知短语检索的基于检索的新型文本生成方法，通过基于文字启发式的初始化和迭代自我强化，解决了构建训练字典(oracles)的主要难题。知识密集型任务和开放式文本生成任务的实验表明，所提出的方法优于标准 LM 和最先进的检索增强方法。此外，与其他检索增强基线相比，该模型在使用扩大或缩小的领域特定索引时都表现出卓越的性能，并实现了最低的生成延迟。这项工作推动了通过检索进行更精确生成的范式转变，从而为 NLP 研究界做出了贡献。

8. 局限性（附录 H）

可扩展性：

在目前的实验中，我们在英文维基百科语料库中训练模型并建立短语索引。当扩展到更大的语料库时，可能会遇到计算上的挑战。由于可能的短语数量显著增加。为了使方法具有可扩展性，一些可能的解决方案包括聚类、降维以及具有亚线性时间复杂度的快速向量搜索算法。

对齐(alignment)：

最近的研究表明，对齐（即按照人类指令对语言模型进行调整）对语言模型的普遍实用性非常重要。因此，将对齐技术纳入方法也是一个重要的未来的研究方向。

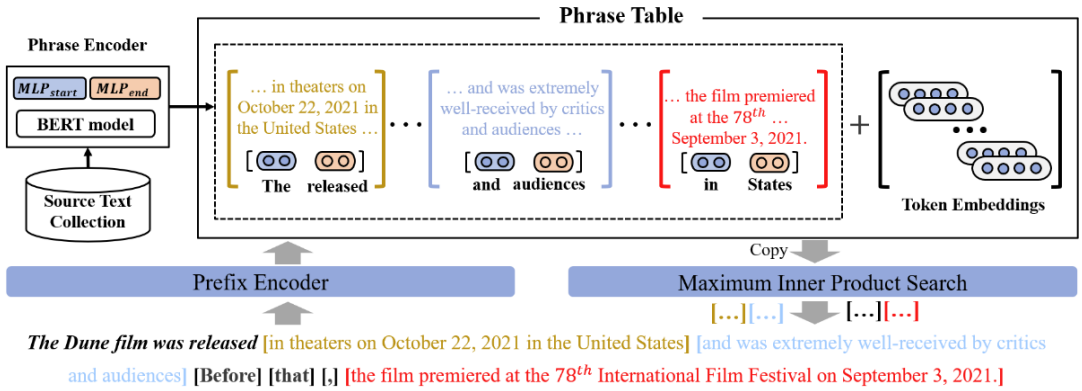
9. 补充说明

CoG:

gmftbyGMFTBY/Copyisallyouneed: [ICLR 2023] Codebase for Copy-Generator model, including an implementation of kNN-LM (github.com)

Tian Lan, Deng Cai, Yan Wang, Heyan Huang, and Xian-Ling Mao. Copy is all you need. In The Eleventh International Conference on Learning Representations, 2023.

在本文中将文本生成表述为从现有文本集合中逐步复制文本片段（如单词或短语）。计算有意义文本片段的上下文表示，并使用高效的矢量搜索工具包对其进行索引。然后，文本生成任务被分解为一系列复制和粘贴操作：在每个时间步骤，我们都会从文本集中的现有文章中寻找合适的文本片段，而不是从独立的词汇表中进行选择。



MIPS:

1405.5869v1.pdf (arxiv.org)

这篇文献提出了一种称为非对称局部敏感哈希（Asymmetric Locality Sensitive Hashing, ALSH）的算法，这是第一个被证明能在次线性时间内完成近似最大内积搜索（Maximum Inner Product Search, MIPS）的方法。传统的局部敏感哈希（Locality Sensitive Hashing, LSH）框架并不适合解决 MIPS 问题，因为它主要用于基于欧几里得距离的近似最近邻搜索，而 MIPS 问题则是基于内积的相似性搜索，特别是在数据无法规范化时。ALSH 通过在哈希之前对数据进行非对称变换，将 MIPS 问题转换成传统的近邻搜索问题，使得搜索更加高效。该算法不仅简单易实施，而且在 Netflix 和 Movielens 数据集上的项目推荐任务中比传统的 LSH 方案（如基于p −稳定分布的 L2LSH 和符号随机投影）有显著的计算效率提高。这表明 ALSH 在特定任务上，特别是在推荐系统中，能够有效地提高内积检索的速度和质量

BM25：

S. Robertson and H. Zaragoza. The probabilistic relevance framework: Bm25 and beyond. Found. Trends Inf. Retr., 3, 2009.

介绍了一种形式化的文档检索框架——概率相关性框架（Probabilistic Relevance Framework, PRF），这个框架在 20 世纪 70 至 80 年代的研究基础上发展而来。它导致了一种极为成功的文本检索算法——BM25 的诞生。这个框架主要从概念的角度来描述背后的概率建模假设，以及不同的排序算法，例如二元独立模型（binary independence model）、相关反馈模型（relevance feedback models）、BM25 及 BM25F。

BM25 算法尤为重要，因为它适用于"词袋"文档检索，能够在文档长度归一化的概率框架内纳入文档内部词频信息，这对于网络搜索和企业搜索算法非常有效 。文献中提出的 PRF 框架近年来的研究已经能够生成新的检索模型，这些模型考虑了文档的结构和链接图信息。更具体地说，PRF 框架将文档的检索视为一个概率问题，假设系统无法准确知道文档与信息需求之间的相关性，但可以使用文档和查询的已知属性提供相关性的概率或统计证据。这个框架的核心是，如果检索出的文档按照相关性的概率降序排列，那么系统的效率将是基于已有数据的最优解。

附录 A 提到的短语编码器：

这篇文献的研究成果主要集中在使用密集短语检索方法在自然语言处理问题上的潜力，特别是它与传统的稀疏检索方法相比。文中探讨了短语检索不仅能够直接用于回答问题和填充任务，还可以用于更粗略层面上的检索，包括段落和文档的检索。研究发现，即便没有重新训练，一个密集的短语检索系统在顶部 5 个准确性上也能比段落检索器取得更好的成绩（提高了 3-5%），这也有助于在使用更少段落的情况下实现更优秀的端到端问答性能。此外，该研究提供了为什么基于短语级别的监督能比基于段落级别的监督学到更好的细粒度推理的解释，并且显示了短语检索能够在如实体链接和知识引导对话等文档检索任务中取得有竞争力的表现。最后，研究展示了如何通过短语过滤和向量量化减小索引大小达到 4-10 倍，使得密集短语检索成为一个实用且多功能的多粒度检索解决方案

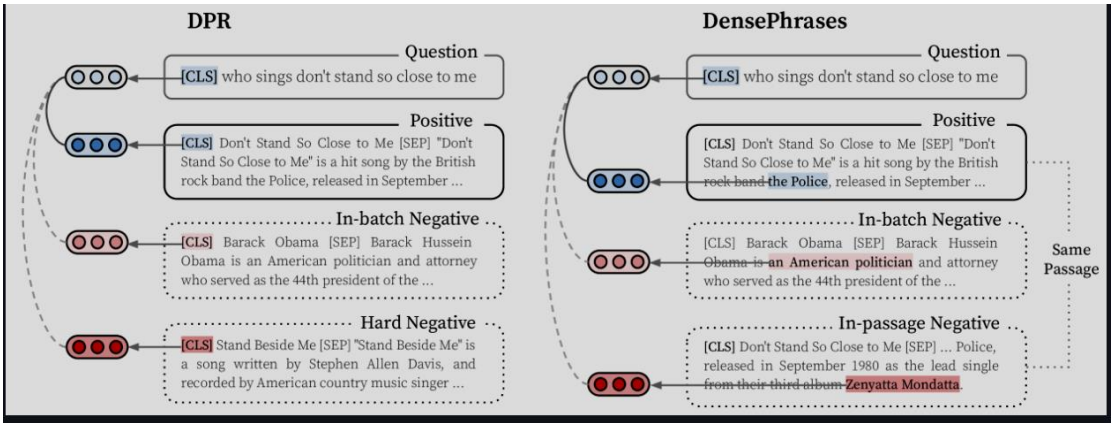


Figure 2: Comparison of training objectives of DPR and DensePhrases. While both models use in-batch negatives, DensePhrases use in-passage negatives (phrases) compared to BM25 hard-negative passages in DPR. Note that each phrase in DensePhrases can directly serve as an answer to open-domain questions.

InfoNCE：

是一种在自我监督学习中用于对比学习的损失函数。它的主要目的是通过对比正负样本对来学习数据的好的表征。在这个损失函数中，我们会有一组样本，其中包含一个正样本（与查询样本类似或相关的样本）和多个负样本（与查询样本不相关的样本）。通过这个损失函数，模型被训练来区分哪些样本是相似的，哪些是不相似的。

具体来说，InfoNCE 损失函数会计算一个分数，这个分数基于查询样本与正样本之间的相似度除以查询样本与整个样本集（包括正样本和所有负样本）的相似度之和。模型的目标是最大化正样本与查询样本之间的相似度的对数概率，同时最小化与负样本之间的相似度。这样可以使模型学会将正样本推近（在特征空间中与查询样本更接近），将负样本推远（在特征空间中与查询样本更远离）。

在实际应用中，InfoNCE 损失函数使得模型能够在无需标签的情况下学习有用的特征，这对于处理大量未标记数据的任务特别有用。由于其能够在预训练时利用大量的未标记数据，因此 InfoNCE 对比学习在计算机视觉和自然语言处理等领域有着广泛的应用。此外，正确地选择负样本，以及调整批量大小和避免错误负样本，都是使用对比学习时需要考虑的重要方面。这个损失函数背后的直观理解是，它可以帮助模型从大量的数据中学习到怎样把相似的东西聚在一起，把不相似的东西分开，以此来学习数据的有效表征

预测损失L_t：

预测损失（Prediction Loss）是用于评估模型对一批训练样本的预测与实际标签之间差异的损失。在分布式训练中，这个损失会在所有副本上计算，并根据全局批量大小（GLOBAL_BATCH_SIZE）进行缩放，以确保不同副本上的损失计算是一致的。具体来说，每个副本计算其接收到的样本的预测损失，将这些损失相加后再除以全局批量大小，如果有的话，还会加上正则化损失

MiniPile：

MiniPile 是从 825GB 的 The Pile 语料库中去重后选出的 6GB 子集。通过预训练 BERT 和 T5 模型，相较于在更大数据集上预训练的原始模型，MiniPile 在 GLUE 和 SNI 基准上的性能下降非常小，展示了其对于语言模型预训练的适用性

DensePhrases：

DensePhrases 是一个密集短语检索系统，它专注于从大量文本中检索具体短语，以提供精确的信息检索和问答功能。通过对文本的细粒度分析和检索，DensePhrases 能够在不同的自然语言处理任务中，如问答和信息检索，提供高效和准确的结果。这个系统展示了短语级别的检索相比于传统的段落或文档级别检索的优势，特别是在需要精确信息时。

FAISS（向量相似性搜索和聚类库）：

FAISS（Facebook AI Similarity Search）是一个专为向量相似性搜索设计的库，用于高效地管理和索引大量的嵌入向量。这个库提供了一系列的索引方法和相关工具，以支持搜索、聚类、压缩和变换向量。FAISS 的目的是在各种大小的向量集合中进行高效的搜索，包括那些可能不适合在 RAM 中完全存储的大型集合

IDF:

IDF（逆向文件频率）是一种计算词语重要性的统计方法，它的核心思想是如果一个词在少数文档中出现次数较多，但在大量文档中出现较少，则认为这个词具有很好的类别区分能力，因而重要性较高。IDF 的计算公式是总文件数目除以包含该词语的文件数目的商的对数值。

图 3：

这张图展示了来自 Med-USMILE 考试的一个示例问题，它是一个多项选择题。题目描述了一位 16 岁的女孩的行为，并询问在过去两年里最可能的诊断是什么。给出了四个选择：[A] 分裂型人格障碍，[B] 反社会人格障碍，[C] 精神分裂形障碍，[D] 自闭症谱系障碍。图中用红色高亮显示了两个检索到的短语，它们与分裂型人格障碍(SPD)相关。这些短语提供了关于 SPD 的描述，比如缺乏对社交关系的兴趣，倾向于孤独的生活方式，保密性，情感冷漠和冷淡等。这个例子说明了如何通过检索到的短语来回应提出的问题，帮助诊断和理解医学知识。

MAUVE:

MAUVE 是一个基于 PyTorch 和 HuggingFace Transformers 的库，用于测量神经文本与人类文本之间的差距。它通过比较两种文本在大型语言模型的量化嵌入空间中分布的 Kullback–Leibler (KL) 散度来计算这种差距。MAUVE 可以识别出由模型大小和解码算法引起的质量差异。MAUVE 计算两个文本分布（比如神经网络生成的文本和人类生成的文本）在量化嵌入空间中的差异，主要通过比较这两种分布的 Kullback–Leibler (KL) 散度。具体来说，它使用大型语言模型的终端隐藏状态作为特征表示，并对这些特征进行量化，以便在计算 KL 散度时使用。

krishnap25/mauve: Package to compute Mauve, a similarity score between neural text and human text. Install with `pip install mauve-text`. (github.com)