生成式信息检索前沿进展与挑战

范意兴[1,2], **唐钰葆**[1,2], **陈建贵**[1,2], **张儒清**[1,2], **郭嘉丰**[1,2]
1. 中国科学院计算技术研究所网络数据科学与技术重点实验室，北京, 100190
2. 中国科学院大学，北京, 100190
{fanyixing, tangyubao21b,chenjiangui18z, zhangruqing, guojiafeng}@ict.ac.cn

摘要

信息检索（Information Retrieval, IR）旨在从大规模的语料集合中找到与用户查询相关的信息，已经成为人们解决日常工作和生活中问题的最重要工具之一。现有的IR系统主要依赖于“索引-召回-重排”的框架，将复杂的检索任务建模成多阶段耦合的搜索过程。这种解耦建模的方式，一方面提升了系统检索的效率，使得检索系统能够轻松应对数十亿的语料集合；另一方面也加重了系统架构的复杂性，无法实现端到端联合优化。为了应对这个问题，近年来研究人员开始探索利用一个统一的模型建模整个搜索过程，并提出了新的生成式信息检索范式，这种新的范式将整个语料集合编码到检索模型中，可以实现端到端优化，消除了检索系统对于外部索引的依赖。当前，生成式检索已经成为IR领域热门研究方向之一，研究人员提出了不同的方案来提升检索的效果，考虑到这个方向的快速进展，本文将对生成式信息检索进行系统的综述，包括基础概念，文档标识符和模型容量。此外，我们还讨论了一些未解决的挑战以及有前景的研究方向，希望能激发和促进更多关于这些主题的未来研究。

关键词： 信息检索 ； 检索模型 ； 生成式检索

Challenges and Advances in Generative Information Retrieval

Yixing Fan[1,2], **Yubao Tang**[1,2], **Jiangui Chen**[1,2], **Ruqing Zhang**[1,2], **Jiafeng Guo**[1,2]
1. CAS Key Lab of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190
2. University of Chinese Academy of Sciences, Beijing 100190
{fanyixing, tangyubao21b,chenjiangui18z, zhangruqing, guojiafeng}@ict.ac.cn

Abstract

Information retrieval (IR) aims to seek relevant information in response to user queries. Existing IR systems mainly rely on the “index-retrieve-then-rank” framework, which models the complex retrieval tasks as a multi-stage search process. Such a decoupling process improves the efficiency of the system, making it possible for retrieval system to handle billions of documents. However, it also increase the complexity of the search architecture, making it difficult to achieve end-to-end optimization. To address this issue, researchers have begun to explore a new paradigm of generative information retrieval. This new paradigm encodes the entire corpus into the search model, enabling end-to-end optimization and eliminating the dependence on external indices. Currently, generative information retrieval has become a hot research direction in IR, and researchers have proposed different solutions to improve retrieval effectiveness. Given the rapid progress in this direction, this article provides a systematic review of

generative information retrieval, including basic concepts, document identifiers, model architectures, and model capacity. In addition, we also discuss some unresolved challenges and promising research directions, hoping to inspire and promote future research on these topics.

Keywords: Information Retrieval , Retrieval Model , Generative Information Retrieval

1 引言

信息检索（Information retrieval, IR）在众多领域中起着重要的作用，包括网络搜索(Chapelle et al., 2011)、问答系统(Karpukhin et al., 2020; Lee et al., 2019)、对话系统(Chen et al., 2017)等任务。IR的核心是从海量文档集合中根据用户查询快速查找满足用户信息需求的文档，为了保证检索的效率，大多数现有的IR系统(Ma et al., 2021b; Ma et al., 2021c)通常采用多阶段分步检索的流水线架构，即“索引-召回-重排序”。具体而言，(i) **索引：**创建文档集中的文档表示、索引以及存储结构； (ii) **召回：**从全部文档集中快速找回与查询潜在相关的小批量文档，形成初始的候选文档集，侧重于检索结果的召回率；以及 (iii) **重排序：**基于召回的候选文档集进一步计算文档与查询相关性，对候选文档进行精排，侧重于检索结果的准确率。 这种流水线架构具有多方面的优势：(i) 高效性：它利用索引实现高效检索，适用于大规模文档集合； (ii) 灵活性：该架构允许在检索模型和排序策略上进行灵活定制； (iii) 可解释性：每个检索阶段都可以进行分析和评估； (iv) 可扩展性：该框架适用于动态增长的文档集合，能够高效存储、组织和检索持续更新的数据。 凭借这些优势，该框架在学术界和工业界都广泛的研究和应用。

尽管这种流水线架构在信息检索中已经证明了其有效性(Chapelle et al., 2011; Lee et al., 2019; Ma et al., 2021b; Ma et al., 2021c)，但它也存在一些固有的局限：首先，流水线框架的目标要求在大规模文档集中搜索数百万个全局文档；其次，多个解耦后的搜索组件通常是独立设计和优化的，缺乏端到端优化的能力；此外，流水线框架是“文档模型”，通常对每个文档进行独立评分，忽视了整个文档集中可用的全局信息；最后，索引阶段需要大量的存储空间来存储预计算的索引，这在可扩展性方面可能带来重大挑战。

鉴于这些局限性，Goole的研究人员(Metzler et al., 2021) 最先提出了一种全新的检索范式，称之为基于模型的信息检索（Model-based IR），以取代长期以来的“索引-检索-排序”架构。这种范式旨在通过一个统一的模型来替代传统方法中涉及的索引、召回和重排序模块，从而彻底改变检索的流程。受到这一蓝图的启发，生成式检索模型（Generative Information Retrieval, GIR）(Metzler et al., 2021)被提出来实现基于模型的信息检索理念，该模型可以根据给定的查询直接生成相关文档的标识符（Document identifiers, DocIDs），从而避免了资源密集型的索引过程。具体而言，生成式检索将检索任务形式化为序列到序列（Sequence-to-sequence, Seq2seq）的生成问题，其核心就是构建一个Seq2seq生成模型。在训练阶段，生成模型将文档内容映射到语义唯一标志符（DocID）实现文档的索引；而在推断阶段，生成模型将查询映射到对应的文档ID以完成文档检索。这里，模型训练通常使用最大似然估计（MLE）损失函数来优化索引和检索两个任务，即：

$$\begin{aligned}\mathcal{L}_{MLE}(\theta) &= \mathcal{L}_{MLE}^{indexing}(\theta) + \mathcal{L}_{MLE}^{retrieval}(\theta) \\ &= \sum_{d_i \in \mathcal{D}} \log P(r_i | GR_\theta(d_i)) + \sum_{q_j \in \mathcal{Q}} \log P(r_j | GR_\theta(q_j)),\end{aligned} \tag{1}$$

其中，$\mathcal{D}$ 表示给定的语料库，$\mathcal{Q}$ 表示查询集合，r_i 或r_j 表示文档d_i 或输入查询q_j 的目标DocID，θ 表示生成模型$GR_\theta(\cdot)$ 的模型参数。这种新的生成式检索范式带来了几个重要的优势：首先，通过解码DocID i，搜索空间被减小为万级的词表空间，与传统的全语料文档库搜索相比，大大降低了搜索复杂性；其次，生成式检索利用统一模型$GR_\theta(\cdot)$ 来涵盖整个检索过程，实现全局性优化；再次，通过使用最大似然估计（Maximum Likelihood Estimation, MLE）损失函数$\mathcal{L}_{MLE}^{indexing}(\cdot)$ 对模型进行编码和使用MLE损失函数$\mathcal{L}_{MLE}^{retrieval}(\cdot)$ 将查询映射到相

关的DocID，生成式检索在生成过程中利用了整个语料库的知识；最后，生成式检索避免了索引相关操作，如文档表示和索引构建。因此，存储需求得到缓解，使得能够处理庞大语料库$\mathcal{D}$的可扩展检索系统而不会产生过高的存储成本。生成式检索的出现引发了广泛的研究兴趣，旨在全面了解其底层机制并发挥其全部潜力。因此，在本文中，我们将介绍生成式检索的基本概念，以及当前相关研究涉及的生成式检索重要的几个方面，分别是DocID表示、模型架构以及模型容量。

- **DocID表示**: DocID表示在生成式检索框架中起着关键作用。一个有效的DocID应该包含来自相应文档的丰富语义信息，具备简洁的特点以便于生成，并且要能够有效区分不同的文档。因此，对DocID特征的探索对于进一步改进生成式检索至关重要。

- **模型架构**: 模型架构是生成式检索的基石。因此，对生成式检索中使用的模型架构$GR_\theta(\cdot)$进行全面的考察对于揭示其潜力是不可或缺的。当前主流的生成式模型包括编码器-解码器架构和仅解码器架构两种。

- **模型容量**: 生成式检索模型的容量，通常与其参数规模相关，显著影响其性能。较大的模型容量通常具有更强的学习能力，但也可能导致额外的计算成本。直观上期望在一定范围内增加模型容量可以提高在给定语料库$\mathcal{D}$上的性能。然而，在超过一定阈值后，效果可能会趋于平稳甚至下降。因此，研究模型容量与性能之间的关系对于优化生成式检索模型并在效果和效率之间取得平衡至关重要。

本文的结构如下所示。首先，在第2节中，我们回顾传统流水式信息检索框架。然后，在第3节中，我们将介绍生成式检索的基本概念。再次，在第4节，第5节和第6节中深入探讨生成式检索中的DocID表示、模型架构以及模型容量三个方面内容。最后，在第7节中讨论当前研究中的挑战和潜在研究方向，并在第8节对本文进行总结。

2 传统流水线检索架构

在正式介绍生成式信息检索之前，我们简要回顾传统“索引-召回-重排序”三步骤的流水线框架，这种架构被广泛应用于现有实际检索系统中(Ma et al., 2021b; Ma et al., 2021c)。该架构通过依次执行索引构建、文档找回和重排序的过程，为信息检索任务提供了系统化的方法。在索引构建阶段，主要是对文档进行离线的表征计算以及索引构建和存储；召回阶段则是基于索引库进行查询，利用索引结构特性从大规模文档集合中快速检索一组可能与用户信息需求匹配的候选文档。最后，在重排序阶段，对查询与召回阶段得到的候选文档进行更加精细化的相关性计算，目的是将最相关的文档排在列表的前面。根据当前文档表示以及索引结构的不同，现有检索框架可以分为两种主要类型(Guo et al., 2022)，即稀疏检索框架和稠密检索框架。

- **稀疏检索**通常基于倒排索引来构建文档的索引存储，它将文档与词关联起来形成文档列表，通过查询词可以快速定位词是否出现在文档集中。一般来说，这类方法利用词项频率和位置等词项的特征来计算文档得分。代表性的检索方法如TF-IDF和BM25已经在实践中被广泛采用，为了增强语义匹配能力，研究人员也探索了将词向量应用到稀疏检索模型中(Zheng and Callan, 2015)。此外，随着预训练技术的发展，研究人员开始研究使用预训练语言模型估计倒排索引的词项权重，例如，DeepCT(Dai and Callan, 2020b)和HDCT(Dai and Callan, 2020a)利用BERT获取上下文化的词项表示，提高了检索性能。

- **稠密检索**则是将文档投影到低维稠密的向量表示，并利用近似最近邻搜索算法进行高效的检索，这类检索方法由于其在语义匹配方面的优势，近年来受到研究人员的广泛关注，并提出了各种技术来提高稠密检索模型的性能。一种常见的方法是采用难负样本挖掘(Cai et al., 2022)进行双塔模型训练，通过选择具有挑战性的负样本来提高模型的判别能力。另一种策略则是采用后交互(Lee et al., 2019)机制，在较后阶段考虑查询和文档之间的交互计算，从而实现更有效的信息融合。此外，知识蒸馏(Vakili Tahami et al., 2020)也被用于稠密检索中，将基于交互的检索模型知识通过蒸馏转移给基于双塔的检索模型，提高稠密检索的效率和效果。最近的研究表明，在大规模语料库上使用对比学习对稠密检索模型进行预训练是有效的(Wu et al., 2022)。这些方法利用预训练语言模型捕捉的丰富上下文信息，在嵌入空间中学习区分正样本（相关文档）和负样本（不相关文档），从而提高检索性能。

在信息检索的重排序阶段，已经提出了各种模型来度量给定查询与候选文档的相关性。代表性的模型包括向量空间模型(Salton et al., 1975)、概率检索模型(Robertson et al., 2009)、排序学习模型(Liu, 2009; Li, 2014)和神经排序模型(Ma et al., 2021b; Ma et al., 2021c)。向量空间模型(Salton et al., 1975)将文档和查询表示为向量，通过计算二者的相似度来评估相关性。概率检索模型(Robertson et al., 2009)使用概率框架估计文档和查询之间的相关性概率。排序学习模型(Burges, 2010)旨在学习一个将文档和查询的特征映射到它们的相关性分数的排序函数，通常利用机器学习算法根据标记的训练数据优化排序函数。神经排序模型(Liu et al., 2017; Ma et al., 2021b; Ma et al., 2021a)则利用深度学习技术学习文档和查询的表示，捕捉它们的语义相关性。

尽管流水线检索框架在实际检索系统中已经被广泛应用，然而，它本身架构的复杂性导致系统难以实现端到端的全局优化，限制了其充分发挥潜力，因此，超越流水线框架并探索替代方法至关重要。

3 生成式检索的基本概念

形式化的，假设$\mathcal{D} = \{d_1, d_2, \dots\}$ 表示一个大规模的文档语料库，其中d_i 表示一个个体文档。给定查询集合$\mathcal{Q}$ 中的查询q 和语料库$\mathcal{D}$，生成式检索模型的目标是生成一组相关文档的DocID (Tay et al., 2022)。接下来，我们将具体描述索引和检索两种基本操作模式，以及学习和推断的过程。

3.1 索引和检索策略

在生成式检索框架中，索引过程被模型训练所替代，而检索过程则被模型推断所取代。一般而言，文档检索任务被转化为单一的生成式形式，并通常采用序列到序列（Seq2Seq）的编码-解码架构，以实现索引和检索的端到端学习。当前主流工作基本都基于Transformer网络来实现编码-解码架构，比如T5(Tay et al., 2022; Wang et al., 2022; Zhuang and Ren, 2022)、BART(De Cao et al., 2020; Bevilacqua et al., 2022)。

在索引阶段，生成式检索将原来流水线架构中的物理索引转化为一个模型训练任务，该任务旨在学习文档d_i 的内容与其对应的文档ID r_i 之间的映射关系。一个广泛使用的策略是Inputs2Target (Tay et al., 2022)，它以原始文档作为输入，并以直接生成的DocID 作为输出，模型使用Teacher Forcing 策略(Hao et al., 2022) 进行训练，采用标准的交叉熵损失函数，如下所示：

$$\mathcal{L}_{MLE}^{indexing}(\theta) = \sum_{d_i \in \mathcal{D}} \log P(r_i | GR_\theta(d_i)), \tag{2}$$

其中$\mathcal{D}$ 表示给定的语料库，GR 表示生成式检索模型。

现有关于索引策略的研究可以大体可以分为两类：(i) 第一类是基于文档内容生成一个全新的ID，其中包括基于数字的DocID (Tay et al., 2022; Zhou et al., 2022b)、基于单词的DocID (De Cao et al., 2020; Chen et al., 2022; Bevilacqua et al., 2022; Chen et al., 2023) 以及基于URL的DocID(Zhou et al., 2022b)。 (ii) 第二类旨在建立从文档到相应DocID的语义映射。各种文档内容类型已被提出，以增强文档与其DocID之间的关联(Tay et al., 2022; Zhou et al., 2022b; Chen et al., 2022)，例如不同语义粒度级别的上下文（例如段落、句子和短语）(Chen et al., 2022; Zhou et al., 2022a) 和超链接信息（例如锚文本）(Chen et al., 2022)。

在检索阶段，生成式检索的目标是为给定输入查询返回一个潜在相关的候选文档的排名列表。为此，生成式检索模型利用在索引阶段微调好的GR模型，通过自回归生成一个给定输入查询$q \in \mathcal{Q}$的文档ID字符串。通常情况下，该模型使用标准的训练目标和交叉熵损失进行训练。检索任务的损失函数定义为：

$$\mathcal{L}_{MLE}^{retrieval}(\theta) = \sum_{q_j \in \mathcal{Q}} \log P(r_j | GR_\theta(q_j)), \tag{3}$$

其中$\mathcal{Q}$是查询集合，r_j是为q_j生成的DocID。候选DocID可以通过使用beam search (Koszelew and Karbowska-Chilinska, 2020)得到，从而得到一个潜在相关的文档排名列表。

3.2 学习和优化

训练生成式检索模型存在两种主要策略：(i) 第一种策略是先训练GR模型进行索引，然后再训练模型进行检索。 (ii) 第二种策略是在多任务设置中训练GR同时进行索引和检索。 实验分析表明，第二种策略在表现上优于第一种策略，尤其面向具有有限标注查询-文档对的大规模语料库检索应用(Wang et al., 2022; Tay et al., 2022)。因此，GR的常用训练策略是采用多任务学习，其形式化表示为：

$$\mathcal{L}_{MLE}(\theta) = \sum_{d_i \in \mathcal{D}} \log P(r_i | GR_\theta(d_i)) + \sum_{q_j \in \mathcal{Q}} \log P(r_j | GR_\theta(q_j)), \tag{4}$$

训练的目标是最大化生成正确的DocID的似然度，用于索引和检索任务。

值得一提的是，在检索阶段的模型训练过程中，为了解决标注数据有限的问题，一些研究采用了通过查询生成技术(Wang et al., 2022; Zhou et al., 2022b) 生成伪查询来加强查询到文档ID的相关性学习；此外，也有利用预训练任务(Chen et al., 2022) 来改进查询到文档ID的相关性关系学习。

3.3 推断

在完成生成式检索模型的训练后，可以在推断阶段以端到端的方式使用它来为给定的查询检索文档。具体而言，经过训练的模型按照从左到右、逐个标记的方式自回归地生成给定测试查询q_j的DocID字符串中的第p个标记$r_{j,p}$，直到生成一个特殊的序列结束（End-of-Sequence，EOS）标记，即，

$$r_{j,p} = GR(q_j, r_{j,0}, r_{j,1}, \ldots, r_{j,p-1}). \tag{5}$$

然而，在实际解码过程中，如果模型的解码空间为整个词汇表中的所有标记，那么生成的输出可能是一个无效DocID。为了克服这个挑战，可以采用带约束的束搜索策略(De Cao et al., 2020)，以确保每个生成的DocID都属于预定的候选集，即整个文档集合中的所有DocID。

具体而言，一般可以利用前缀树建立约束，其中节点标记为从预定义候选集中选择的标记。对于前缀树中的每个节点，其子节点表示沿着从根节点到给定节点的前缀所建立的所有可行延续。通常情况下，用于生成DocID的前缀树相对较小，可以事先计算并预加载到内存中。

4 DocID 表示

在生成式检索中，生成式检索模型通过Seq2seq模型，在给定查询和文档上下文之间建立映射关系，这些文档上下文的语义内容则时由DocID的短字符串来刻画。这里，最核心的需要是设计有效的DocID表示来捕获文档内容的潜在语义，这里要求DocID具有语义信息、简洁明了并能够有效区分不同文档。在本节中，我们介绍当前主流的不同类型DocID，分别是基于数字的DocID和基于词的DocID，以下将对这两类DocID方法进行详细描述。

4.1 基于数字的DocID表示方法

基于数字的DocID 包含了使用数字值表示DocID的方法，可以使用随机数或具有语义意义的数值来实现。在没有高质量元数据（例如唯一的、语义丰富的标题）的情况下，这些方法已被证明具有良好的性能。一般而言，基于数字的DocID表示方法可分为三种主要类型 (Tay et al., 2022)，包括原子DocID、字符串DocID 和语义结构化DocID。

- **原子DocID** 使用唯一且随机的数字表示文档。具体而言，生成式检索模型被训练为为每个不同的文档输出一个logit 值，最后，解码器的输出层大小则为隐藏层大小乘以文档数量。这种方法的主要优点是构建简单，但缺点是随着文档数量的增长，模型的容量也会增加。

- **字符串DocID** 则依赖于整数字符串来构建文档的唯一表征。其核心是通过逐步解码DocID字符串中的每个标记，从而消除大型softmax 输出空间的挑战。字符串DocID 方法与原子DocID 方法的区别在于前者使用可分词的字符串DocID，并且涉及多步解码生成，而后者使用唯一且随机的数字DocID进行单步解码生成。

- **语义结构化DocID** 将文档的语义表达压缩成一个较短的数字组合作为文档ID。其目标是要捕捉文档的语义信息，自动生成能传达其对应文档语义信息的DocID。DocID 的结构可以在每个解码步骤后有效地减少搜索空间。例如，通过k-means 聚类构建的DocID 可能会在语义上相似的文档中共享前缀。

4.2 基于词的DocID表示方法

基于词的DocID表示方法是指通过直接从原始文档或其元数据中提取DocID，或者基于文档的语义信息进行重构，从而与文档建立强大的语义联系来实现。与基于数字的方法相比，基于词的方法以更自然和易于理解的方式传达语义信息。目前，广泛使用的基于单词的DocID 方法包括基于标题、基于URL 和基于N-gram 的方法。

- **基于标题的DocID** 直接使用文档的标题作为其DocID。标题通常是整个文档的简短而丰富的摘要，提供了对文档内所含信息的宏观概述。此外，在某些知识库（如维基百科）中，标题通常是唯一的，因此作为DocID 是一个理想的选择。这种方法在知识密集型语言任务中也被证明是有效的(De Cao et al., 2020; Chen et al., 2022)。然而，缺点是并非所有文档都有高质量的标题。

- **基于URL的DocID** 将与文档对应的网页URL作为其DocID。一般来说，URL 是唯一的且易于获取，可以快速而准确地与相应的文档进行关联。然而，与基于标题的方法相比，URL 所携带的语义信息较弱，并且可能引入额外的噪音（因为URL 中可能存在无效字段）(Zhou et al., 2022b)。

- **基于N-gram的DocID** 利用文档中连续出现的N-gram 作为其DocID。N-gram 容易获取，但重复率较高，因此需要额外设计去重功能。此外，在推理阶段，无法直接使用束约束搜索，需要使用FM 索引(Chen et al., 2023)。

5 模型结构

检索模型架构的选择塑造了生成式检索的基本结构，对检索性能起着决定性作用。当前的生成式检索工作(Tay et al., 2022; De Cao et al., 2020; Wang et al., 2022; Bevilacqua et al., 2022) 使用编码器-解码器结构的生成模型作为主干模型。目前尚未有工作探索生成式检索结构中各个结构的作用。然后，模型结构作为生成式检索中最核心的部分，我们这里简单探讨一下不同的网络结构在生成式检索中的应用模式。在本节，我们重点讨论如何利用编码器-解码器和仅解码器架构实现生成式检索。

5.1 编码器-解码器架构

编码器-解码器架构是实现生成式检索的常见选择。在这个设置中，编码器接收输入查询，将其编码为上下文向量，捕捉查询的语义信息。然后，解码器在编码的查询表示基础上生成相应的DocID。具体而言，训练阶段和推理阶段的过程如下：(i) 在训练阶段，模型使用查询和相应DocID的样本进行训练。编码器对输入查询或文章进行编码，解码器以自回归的方式进行训练，生成准确的DocID。训练目标则是在给定输入查询的情况下，最大化生成目标DocID的似然估计。 (ii) 在推理阶段，编码器-解码器模型接受查询作为输入，并根据查询和相关文档之间的相关性得分生成DocID。

5.2 仅解码器架构

除了编码器-解码器架构，仅解码器架构也可以用于生成式检索任务。事实上，仅解码器架构已经在大语言模型中发挥了重要作用。在这个设置中，输入序列不会被显式地编码成固定长度的表示。相反，仅解码器的模型根据初始状态或作为输入查询提供的提示直接生成DocID。具体而言，训练阶段和推理阶段的过程如下：(i) 在训练阶段，模型根据给定的查询或者文章表示的初始状态生成正确的DocID。训练目标同样是最大化生成目标DocID的似然估计。 (ii) 在推理阶段，仅解码器模型接受提示或初始状态，并根据从训练数据中学到的模式生成DocID。

6 模型容量

生成式检索模型的容量直接影响检索模型的性能，本节重点介绍模型容量（即模型参数规模）和语料库大小之间的关系。直观地说，在一定范围内增加模型容量预计会提高在给定数据集上的性能，但超过一定阈值后，效果可能趋于平稳甚至下降。

在生成式检索中，模型大小（以参数数量衡量）和语料库大小（以文档数量衡量）是影响系统性能和可扩展性的两个重要因素。

- **模型大小**指的是生成式检索模型中可学习参数的数量，包括用于生成DocID的神经网络架构的权重和偏置。一般来说，具有更多参数的较大模型有能力捕捉更完整的内容语义以及更复杂的查询-文档相关模式。然而，较大的模型在训练和推理时也需要更多的计算资源。

- **语料库大小**指的是检索系统中可用文档的数量。较大的语料库意味着更大的搜索空间和更复杂的相关模式需要生成式检索模型进行学习。管理和处理大型语料库可能会引入与计算效率、可扩展性和资源利用相关的挑战。

6.1 内存空间：生成式检索与传统检索的外部索引

在生成式检索中，索引构建是模型训练的一个特殊情况，所有与语料库相关的信息都被编码在单个神经模型的参数中。而在传统的多阶段索引-检索-排序流程中，外部构建的查询索引与数据或信息源相关联。在这里，我们介绍单个生成式检索模型所需的内存空间与传统流水线架构中外部索引所需的内存空间之间的关系。

- **生成式检索**：所需的内存空间主要取决于生成模型本身参数的大小。生成模型通常包含在训练过程中学习的参数，例如权重和偏置。较大的模型通常需要更多的内存空间。除了模型参数之外，生成式检索在推理过程中可能还需要内存来存储中间表示，例如前缀树和FM索引。这些中间表示对于生成相关和信息丰富的DocID是必要的。

- **传统检索**：传统的检索方法，例如稀疏检索和稠密检索，通常依赖于外部索引来存储和组织文档集合。这些索引所需的内存空间则取决于文档集合的大小和使用的索引方案。

  – 稀疏检索：通常使用倒排索引，将词项映射到包含它们的文档。索引的大小取决于集合中唯一词项的数量以及每个文档的平均术语数。索引所需的内存空间随着文档集合的大小和词表中词项数量的增加而增加。
  – 稠密检索：采用向量嵌入等技术在连续向量空间中表示文档和查询。这些嵌入通常存储在索引中，例如近似最近邻索引或稠密向量索引。索引所需的内存空间取决于文档的数量和向量嵌入的维度。较大的文档集合或更高维度的嵌入将需要更多的内存空间。

在实际场景中部署生成式检索模型时，需要仔细权衡模型大小、语料库大小和计算成本之间的平衡。一方面，增加模型大小通常会带来性能的提升，较大的模型在捕捉查询-文档相关性和生成准确的DocID方面具有更强的能力。另一方面，更大的语料库能提供更丰富的信息源，使模型能够更好地理解上下文并生成更相关的响应。然而，这种性能提升是以增加计算要求为代价的。较大的模型在训练和推理过程中消耗更多的内存和计算资源，导致训练时间更长和推理成本更高。类似地，增加语料库大小会增加需要处理的数据量，导致训练和推理时间更长。此外，随着模型大小和语料库大小的持续增长，性能改进的边际效益可能变得不那么显著，而相比之下计算成本的增加更为显著。

总的来说，在实际部署生成式检索模型时，需要考虑应用程序的特定要求和约束，包括可用的计算资源、时间限制和所需的性能水平。例如，在实时响应至关重要的场景中，可能需要通过选择较小的模型大小或限制语料库大小来优先考虑计算效率。相反，如果应用程序需要高准确性和性能，为了更大的模型和语料库可能会牺牲计算成本。

7 挑战和展望

在这一节，我们讨论生成式检索的几个重要挑战，希望能够给未来研究方向提供一些有价值的建议。

7.1 生成式架构

当前的生成式检索模型还没有完全实现统一传统检索流程中的三个步骤的设想，研究重点关注在利用生成式模型来替代“索引-召回”两步，而没有覆盖重排序阶段。与此同时，尽管这些方法在一定程度上提升了性能，但仍难以超越强大的稠密检索方法甚至是稀疏检索方法（如BM25 (Robertson and Zaragoza, 2009)）。一个重要的原因也在于生成式检索不能覆盖传统检索流程的重排序阶段，在很多时候性能难以媲美传统检索，这也表明在生成式检索中仍有很大的改进空间，以实现全面而有效的排序能力。

生成式检索模型主要依赖于Transformer架构，然而，Transformer架构存在一些固有的限制，比如输入长度的限制。因此，有必要探索新型网络架构以克服这种限制。这里，可以利用诸如Longformer网络结构(Beltagy et al., 2020)、多尺度解码器(Yu et al., 2023)以及扩张注意力(Ding et al., 2023)等方法来提升模型的输入长度。

7.2 端到端DocID学习

在生成式检索中，文档标识的学习通常遵循两阶段的过程。首先，使用诸如BERT等单独的模型来辅助学习DocIDs的表示。随后，利用学习到的DocID表示建立文档/查询和DocIDs之间的映射关系。另一种可行的学习方法则是采用端到端学习，这样模型可以直接在统一的框架内同时优化DocID表示的学习和文档/查询与DocIDs之间的映射关系。这可以简化学习流程，提高整体效率，并有望进一步改善生成式检索模型的性能。然而，同时优化两个目标需要权衡二者之间的相互影响，需要仔细设计优化方法，考虑到同时学习与DocIDs生成相关的各个组成部分所涉及的复杂性和挑战性。

7.3 场景受限

当前生成式检索方法大都在文档规模受限的场景下进行验证，例如MS MARCO中的文档检索或锻炼检索、Wikipedia中的实体检索等，这类检索假设文档语料规模不大，同时文档集相对固定。然而，实际检索中文档集规模通常很大，且文档会源源不断的增加，如何应对大规模文档以及动态新增文档的表示学习与DocID生成是一个重要的挑战。

8 总结

本文对生成式信息检索进行了系统的综述，区别于现有的IR系统主要采用了“索引-召回-重排”的框架，生成式检索利用统一的模型来建模整个搜索过程，这种新型的检索架构能够实现了端到端的优化，消除了对外部索引的依赖。本文对生成式信息检索的基本概念、核心方法以及难点进行了梳理，同时，探讨了一些未解决的挑战和有前景的研究方向，希望能激发和促进未来关于生成式检索的研究。

参考文献

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen-tau Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers.

Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11:23–581.

Yinqiong Cai, Jiafeng Guo, Yixing Fan, Qingyao Ai, Ruqing Zhang, and Xueqi Cheng. 2022. Hard negatives or false negatives: Correcting pooling bias in training neural ranking models. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 118–127.

Olivier Chapelle, Shihao Ji, Ciya Liao, Emre Velipasaoglu, Larry Lai, and Su-Lin Wu. 2011. Intent-based diversification of web search results: metrics and algorithms. *Information Retrieval*.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Sigkdd Explorations*, 19(2):25–35.

Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yiqun Liu, Yixing Fan, and Xueqi Cheng. 2022. Corpusbrain: Pre-train a generative retrieval model for knowledge-intensive language tasks. In *CIKM*, pages 191–200.

Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yiqun Liu, Yixing Fan, and Xueqi Cheng. 2023. A unified generative retriever for knowledge-intensive language tasks via prompt learning. In *SIGIR*.

Zhuyun Dai and Jamie Callan. 2020a. Context-aware document term weighting for ad-hoc search. In *Proceedings of The Web Conference 2020*, pages 1897–1907.

Zhuyun Dai and Jamie Callan. 2020b. Context-aware term weighting for first stage passage retrieval. In *SIGIR*, pages 1533–1536.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. In *ICLR*.

Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, and Furu Wei. 2023. Longnet: Scaling transformers to 1, 000, 000, 000 tokens. *CoRR*, abs/2307.02486.

Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022. Semantic models for the first-stage retrieval: A comprehensive review. *TOIS*, 40(4):1–42.

Yongchang Hao, Yuxin Liu, and Lili Mou. 2022. Teacher forcing recovers reward functions for text generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 12594–12607.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*, pages 6769–6781.

Jolanta Koszelew and Joanna Karbowska-Chilinska. 2020. Beam search algorithm for anti-collision trajectory planning for many-to-many encounter situations with autonomous surface vehicles. *Sensors*, 20:4115.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *ACL*, pages 6086–6096.

Hang Li. 2014. *Learning to Rank for Information Retrieval and Natural Language Processing, Second Edition*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Shichen Liu, Fei Xiao, Wenwu Ou, and Luo Si. 2017. Cascade ranking for operational e-commerce search. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1557–1565. ACM.

Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331.

Xinyu Ma, Jiafeng Guo, and Ruqing Zhang. 2021a. B-prop: bootstrapped pre-training with representative words prediction for ad-hoc retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021b. Prop: pre-training with representative words prediction for ad-hoc retrieval. In *ACM WSDM*.

Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Yingyan Li, and Xueqi Cheng. 2021c. B-prop: bootstrapped pre-training with representative words prediction for ad-hoc retrieval. In *SIGIR*, pages 1513–1522.

Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. 2021. Rethinking search: Making domain experts out of dilettantes. *SIGIR Forum*.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Yi Tay, Vinh Q Tran, Mostafa Dehghani, Jianmo Ni, and Dara Bahri. 2022. Transformer memory as a differentiable search index.

Amir Vakili Tahami, Kamyar Ghajar, and Azadeh Shakery. 2020. Distilling knowledge for fast retrieval-based chat-bots. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in information retrieval*, pages 2081–2084.

Yujing Wang, Yingyan Hou, Haonan Wang, and Ziming Miao. 2022. A neural corpus indexer for document retrieval. *arXiv preprint arXiv:2206.02743*.

Bohong Wu, Zhuosheng Zhang, Jinyuan Wang, and Hai Zhao. 2022. Sentence-aware contrastive learning for open-domain passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1062–1074.

Lili Yu, Daniel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. 2023. MEGABYTE: predicting million-byte sequences with multiscale transformers. *CoRR*, abs/2305.07185.

Guoqing Zheng and Jamie Callan. 2015. Learning to reweight terms with distributed representations. In *SIGIR*, pages 575–584.

Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, and Ji-Rong Wen. 2022a. Dynamicretriever: A pre-training model-based ir system with neither sparse nor dense index. *arXiv preprint arXiv:2203.00537*.

Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, Peitian Zhang, and Ji-Rong Wen. 2022b. Ultron: An ultimate retriever on corpus with a model-based indexer. *arXiv preprint arXiv:2208.09257*.

Shengyao Zhuang and Houxing Ren. 2022. Bridging the gap between indexing and retrieval for differentiable search index with query generation. *arXiv preprint arXiv:2206.10128*.