From Matching to Generation: A Survey on Generative Information Retrieval

XIAOXI LI and JIAJIE JIN, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
YUJIA ZHOU, Tsinghua University, Beijing, China
YUYAO ZHANG, PEITIAN ZHANG, YUTAO ZHU, and ZHICHENG DOU, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China

Information Retrieval (IR) systems are crucial tools for users to access information, which have long been dominated by traditional methods relying on similarity matching. With the advancement of pre-trained language models, Generative Information Retrieval (GenIR) emerges as a novel paradigm, attracting increasing attention. Based on the form of information provided to users, current research in GenIR can be categorized into two aspects: (1) *Generative Retrieval* (*GR*) leverages the generative model's parameters for memorizing documents, enabling retrieval by directly generating relevant document identifiers without explicit indexing. (2) *Reliable Response Generation* employs language models to directly generate information users seek, breaking the limitations of traditional IR in terms of document granularity and relevance matching while offering flexibility, efficiency, and creativity to meet practical needs. This article aims to systematically review the latest research progress in GenIR. We will summarize the advancements in GR regarding model training and structure, document identifier, incremental learning, and so on, as well as progress in reliable response generation in aspects of internal knowledge memorization, external knowledge augmentation, and so on. We also review the evaluation, challenges, and future developments in GenIR systems. This review aims to offer a comprehensive reference for researchers, encouraging further development in the GenIR field (Github Repository: https://github.com/RUC-NLPIR/GenIR-Survey).

CCS Concepts: • **Information systems** → **Retrieval models and ranking**;

Additional Key Words and Phrases: Generative Information Retrieval, Generative Document Retrieval, Reliable Response Generation

This work was supported by Beijing Natural Science Foundation under Grant No. L233008 and National Natural Science Foundation of China under Grant No. 62272467.
Authors' Contact Information: Xiaoxi Li, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China; e-mail: xiaoxi_li@ruc.edu.cn; Jiajie Jin, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China; e-mail: jinjiajie@ruc.edu.cn; Yujia Zhou, Tsinghua University, Beijing, China; e-mail: zhouyujia@mail.tsinghua.edu.cn; Yuyao Zhang, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China; e-mail: 2020201710@ruc.edu.cn; Peitian Zhang, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China; e-mail: namespace.pt@gmail.com; Yutao Zhu, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China; e-mail: yutaozhu94@gmail.com; Zhicheng Dou (corresponding author), Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China; e-mail: dou@ruc.edu.cn.

ACM 1558-2868/2025/5-ART83
https://doi.org/10.1145/3722552

ACM Reference format:
Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. 2025. From Matching to Generation: A Survey on Generative Information Retrieval. *ACM Trans. Inf. Syst.* 43, 3, Article 83 (May 2025), 62 pages.
https://doi.org/10.1145/3722552

1 Introduction

Information Retrieval (IR) systems are crucial for navigating the vast sea of online information in today's digital landscape. From using search engines such as Google [75], Bing [195], and Baidu [208], to engaging with **Question-Answering (QA)** or dialogue systems like ChatGPT [208] and Bing Chat [196] and discovering content via recommendation platforms like Amazon [4] and YouTube [76], IR technologies are integral to our everyday online experiences. These systems are reliable and play a key role in spreading knowledge and ideas globally.

Traditional IR systems primarily rely on sparse retrieval methods based on word-level matching. These methods, which include Boolean Retrieval [240], BM25 [236], SPLADE [64], and Uni-COIL [162], establish connections between vocabulary and documents, offering high retrieval efficiency and robust system performance. With the rise of deep learning, dense retrieval methods such as DPR [116] and ANCE [322], based on the bidirectional encoding representations from the BERT model [120], capture the deep semantic information of documents, significantly improving retrieval precision. Although these methods have achieved leaps in accuracy, they rely on large-scale document indices [56, 186] and cannot be optimized in an end-to-end way. Moreover, when people search for information, what they really need is a precise and reliable answer. This document ranking list-based IR approach still requires users to spend time summarizing their required answers, which is not ideal enough for information seeking [194].

With the development of Transformer-based pre-trained language models such as T5 [229], BART [137], and GPT [226], they have demonstrated their strong text generation capabilities. In recent years, **Large Language Models (LLMs)** have brought about revolutionary changes in the field of **AI-Generated Content (AIGC)** [18, 357]. Based on large pre-training corpora and advanced training techniques like RLHF [35], LLMs [8, 104, 208, 284] have made significant progress in natural language tasks, such as dialogue [208, 280] and QA [173, 223]. The rapid development of LLMs is transforming IR systems, giving rise to a new paradigm of **Generative Information Retrieval (GenIR)**, which achieves IR goals through generative approaches.

As envisioned by Metzler et al. [194], in order to build an IR system that can respond like a domain expert, the system should not only provide accurate responses but also include source citations to ensure the credibility of the results. To achieve this, GenIR models must possess both sufficient memorized knowledge and the ability to recall the associations between knowledge and source documents, which could be the final goal of GenIR systems. Currently, research in GenIR primarily focuses on two main patterns: (1) **Generative Retrieval (GR)**, which involves retrieving documents by generating their identifiers; and (2) *Reliable Response Generation*, which entails directly generating user-centric responses with reliability enhancement strategies. Noting that although these two methods have not yet been integrated technically, they represent two primary forms by which IR systems present information to users in generative manners: either by generating lists of **Document Identifiers (DocIDs)** or by generating reliable and user-centric responses. Figure 1 illustrates the difference between these two forms. These strategies are essential to the next generation of IR and constitute the central focus of this survey.

GR, a new retrieval paradigm based on generative models, is garnering increasing attention. This approach leverages the parametric memory of generative models to directly generate DocIDs

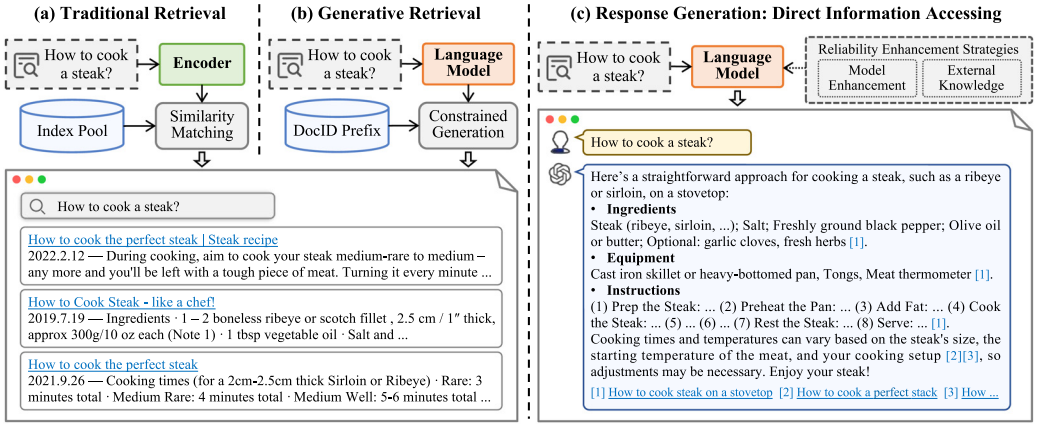

Fig. 1. Exploring IR evolution: from traditional to generative methods. This diagram illustrates the shift from traditional similarity-based document matching (a) to GenIR techniques. Current GenIR methods can be categorized into two types: GR (b), which retrieves documents by directly generating relevant DocIDs constrained by a DocID prefix tree; and response generation (c), which directly generates reliable and user-centric answers.

related to the documents [17, 279, 305, 369]. Figure 1 illustrates this transition, where traditional IR systems match queries to documents based on an indexed database (Figure 1(a)), while generative methods use language models to retrieve by directly generating relevant DocIDs (Figure 1(b)). Specifically, GR assigns a unique identifier to each document, which can be numeric-based or text-based, and then trains a GR model to learn the mapping from queries to the relevant DocIDs. This allows the model to index documents using its internal parameters. During inference, GR models use constrained beam search to limit the generated DocIDs to be valid within the corpus, ranking them based on generation probability to produce a ranked list of DocIDs. This eliminates the need for large-scale document indexes in traditional methods, enabling end-to-end training of the model.

Recent studies on GR have delved into model training and structure [6, 152, 279, 305, 363, 367, 370], DocID design [17, 263, 279, 286, 328], continual learning on dynamic corpora [79, 123, 191], downstream task adaptation [26, 27, 151], multi-modal GR [156, 177, 355], and generative recommender systems [73, 231, 302]. The progress in GR is shifting retrieval systems from matching to generation. It has also led to the emergence of workshops [10] and tutorials [277]. However, there is currently no comprehensive review that systematically organizes the research, challenges, and prospects of this emerging field.

Reliable response generation is also a promising direction in the IR field, offering user-centric and accurate answers that directly meet users' needs. Since LLMs are particularly adept at following instructions [357], capable of generating customized responses, and can even cite the knowledge sources [203, 221], making direct response generation a new and intuitive way to access information [53, 74, 239, 313, 365]. As illustrated in Figure 1, the generative approach marks a significant shift from traditional IR systems, which return a ranked list of documents (as shown in Figure 1(a) and (b)). Instead, response generation methods (depicted in Figure 1(c)) offer a more dynamic form of information access by directly generating detailed, user-centric responses, thereby providing a richer and more immediate understanding of the information need behind the users' queries.

However, the responses generated by language models may not always be reliable. They have the potential to generate irrelevant answers [84], contradict factual information [89, 103], provide

outdated data [289], or generate toxic content [92, 261]. Consequently, these limitations render them unsuitable for many scenarios that require accurate and up-to-date information. To address these challenges, the academic community has developed strategies across four key aspects: enhancing internal knowledge [15, 36, 55, 118, 131, 192, 241, 265, 283]; augmenting external knowledge [5, 112, 138, 150, 203, 243, 331]; generating responses with citation [128, 141, 155, 203, 312]; and improving personal information assistance [148, 171, 293, 325]. Despite these efforts, there is still a lack of a systematic review that organizes the existing research under this new paradigm of generative information access.

This review will systematically review the latest research progress and future developments in the field of GenIR, as shown in Figure 2, which displays the classification of research related to the GenIR system. We will introduce background knowledge in Section 2, GR technologies in Section 3, direct information accessing with generative language models in Section 4, evaluation in Section 5, current challenges and future directions in Section 6, respectively. Section 7 will summarize the content of this review. This article is the first to systematically organize the research, evaluation, challenges, and prospects of GenIR, while also looking forward to the potential and importance of GenIR's future development. Through this review, readers will gain a deep understanding of the latest progress in developing GenIR systems and how it shapes the future of information access. The main contribution of this survey is summarized as follows:

— *First comprehensive survey on GenIR.* This survey is the first to comprehensively organize the techniques, evaluation, challenges, and prospects on the emerging field of GenIR, providing a deep understanding of the latest progress in developing GenIR systems and its future in shaping information access.
— *Systematic categorization and in-depth analysis.* The survey offers a systematic categorization of research related to GenIR systems, including GR, reliable response generation. It provides an in-depth analysis of each category, covering model training and structure, DocID, and so on in GR; internal knowledge memorization, external knowledge enhancement, and so on for reliable response generation.
— *Comprehensive review of evaluation metrics and benchmarks.* The survey reviews a range of widely used evaluation metrics and benchmark datasets for accessing GenIR methods, alongside analysis on the effectiveness and weaknesses of existing GenIR methods.
— *Discussions of current challenges and future directions.* The survey identifies and discusses the current challenges faced in the GenIR field. We also provide potential solutions for each challenge and outline future research directions for building GenIR systems.

2 Background and Preliminaries

IR techniques aim at efficiently obtaining, processing, and understanding information from massive data. Technological advancements have continuously driven the evolution of these methods: from early keyword-based sparse retrieval to deep learning-based dense retrieval, and more recently, to GR, LLMs, and their augmentation techniques. Each advancement enhances retrieval accuracy and efficiency, catering to the complex and diverse query needs of users.

2.1 Traditional IR

Sparse Retrieval. In the field of traditional IR, sparse retrieval techniques implement fast and accurate document retrieval through the inverted index method. Inverted indexing technology maps each unique term to a list of all documents containing that term, providing an efficient means for IR in large document collections. Among these methods, **Term Frequency-Inverse Document**

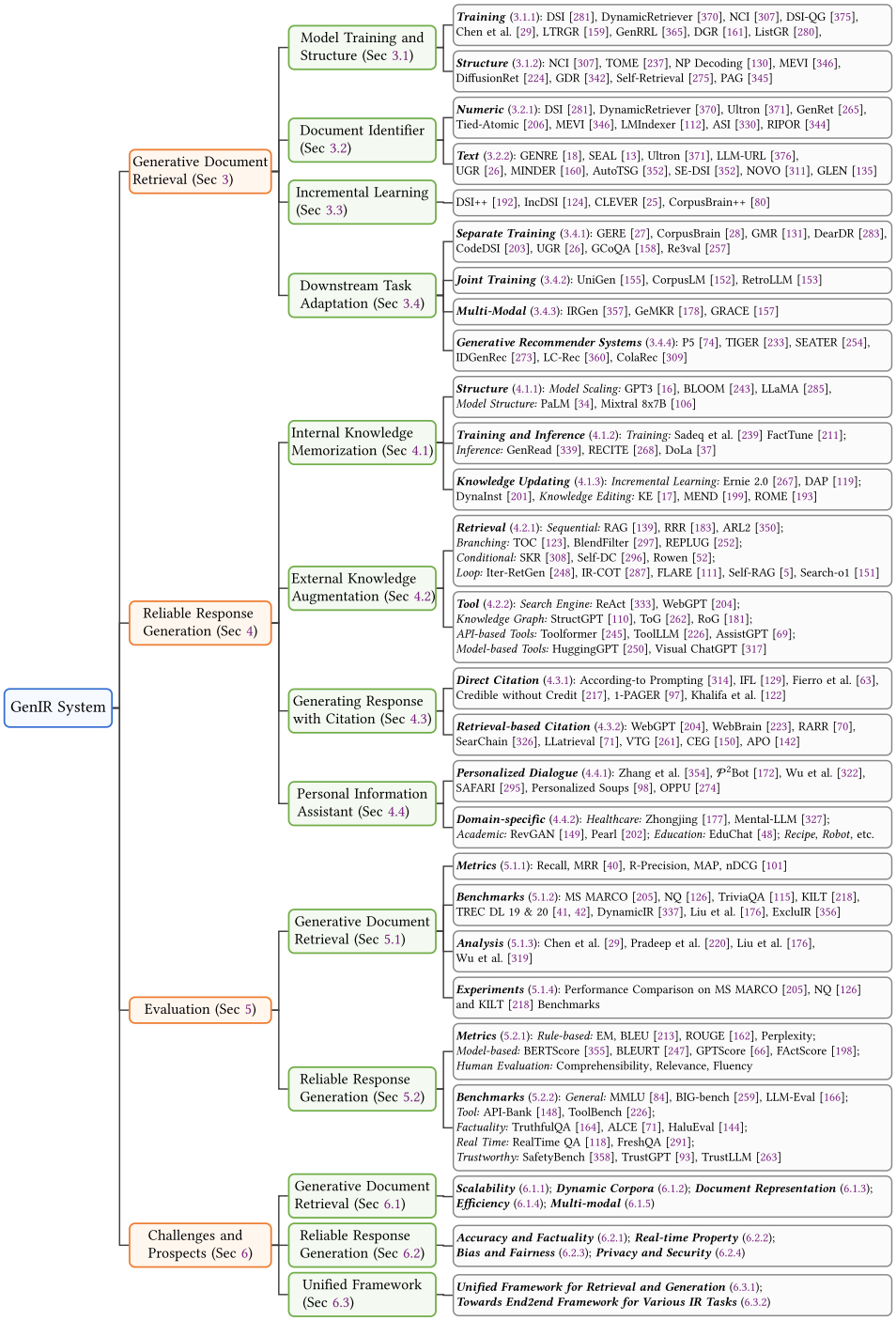

Fig. 2. Taxonomy of research on GenIR: investigating GR, reliable response generation, evaluation, challenges, and prospects. GDR, Generative Dense Retrieval.

Frequency (TF-IDF) [233] is a particularly important statistical tool used to assess the importance of a word in a document collection, thereby widely applied in various traditional retrieval systems.

The core of sparse retrieval technology lies in evaluating the relevance between documents and user queries. Specifically, given a document collection $\mathcal{D}$ and a user query q, traditional IR systems identify and retrieve information by calculating the relevance $\mathcal{R}$ between document d and query q. This relevance evaluation typically relies on the similarity measure between document d and query q, as shown below:

$$\mathcal{R}(q,d) = \sum\nolimits_{t \in q \cap d} \text{tf-idf}(t,d) \cdot \text{tf-idf}(t,q), \tag{1}$$

where t represents the terms common to both query q and document d, and tf-idf(t,d) and tf-idf(t,q) represent the TF-IDF weights of term t in document d and query q, respectively. Although sparse retrieval methods like TF-IDF [233] and BM25 [238] excel at fast retrieval, it struggles with complex queries involving synonyms, specialized terms, or context, as term matching and TF-IDF may not fully meet users' information needs [179].

Dense Retrieval. The advent of pre-trained language models like BERT [120] has revolutionized IR, leading to the development of dense retrieval methods, like DPR [116], ANCE [322], E5 [296], and SimLM [297]. Unlike traditional sparse retrieval, these methods leverage Transformer-based encoders to create dense vector representations for both queries and documents. This approach enhances the capability to grasp the underlying semantics, thereby improving retrieval accuracy.

The core of dense retrieval lies in converting documents and queries into vector representations. Given document d and query q and their vector representations $\mathbf{v}_q$, each document d is transformed into a dense vector $\mathbf{v}_d$ through a pre-trained language model, similarly, query q is transformed into vector $\mathbf{v}_q$. Specifically, we can use encoder functions $E_d(\cdot)$ and $E_q(\cdot)$ to represent the encoding process for documents and queries, respectively:

$$\mathbf{v}_d = E_d(d), \quad \mathbf{v}_q = E_q(q), \tag{2}$$

where $E_d(\cdot)$ and $E_q(\cdot)$ can be the same model or different models optimized for specific tasks.

Dense retrieval methods evaluate relevance by calculating the similarity between the query vector and document vector, which can be measured by cosine similarity, expressed as follows:

$$\mathcal{R}(q,d) = \cos(\mathbf{v}_q, \mathbf{v}_d) = \frac{\mathbf{v}_q \cdot \mathbf{v}_d}{|\mathbf{v}_q||\mathbf{v}_d|}, \tag{3}$$

where $\mathbf{v}_q \cdot \mathbf{v}_d$ represents the dot product of query vector $\mathbf{v}_q$ and document vector $\mathbf{v}_d$, and $|\mathbf{v}_q|$ and $|\mathbf{v}_d|$ respectively represent the magnitudes of the query and document vector. Finally, documents are ranked based on these similarity scores to identify the most relevant ones for the user.

2.2 GR

With the significant progress of language models, GR has emerged as a new direction in the field of IR [194, 279, 326]. Unlike traditional index-based retrieval methods, GR relies on pre-trained generative language models, such as T5 [229] and BART [137], to directly generate DocIDs relevant to the query, thereby achieving end-to-end retrieval without relying on large-scale pre-built document indices.

DocID Construction and Prefix Constraints. To facilitate GR, each document d in the corpus $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$ is assigned a unique DocID d', forming the set $\mathcal{D}' = \{d'_1, d'_2, \ldots, d'_N\}$. This mapping is typically established via a bijective function $\phi : \mathcal{D} \to \mathcal{D}'$, ensuring that:

$$\phi(d_i) = d'_i, \quad \forall d_i \in \mathcal{D}. \tag{4}$$

To enable the language model to generate only valid DocIDs during inference, we construct *prefix constraints* based on $\mathcal{D}'$. This is typically implemented using a trie (prefix tree), where each path from the root to a leaf node corresponds to a valid DocID.

Constrained Beam Search. Given a query q, the GR model aims to generate the top-k DocIDs that are most relevant to q. The language model $P(\cdot|q;\theta)$ generates DocIDs token by token, guided by the prefix constraints. At each decoding step i, only those tokens that extend the current partial sequence $d'_{<i}$ into a valid prefix of some DocIDs in $\mathcal{D}'$ are considered. Formally, the set of allowable next tokens is:

$$\mathcal{V}(d'_{<i}) = \{v \mid \exists d' \in \mathcal{D}' \text{ such that } d'_{<i}v \text{ is a prefix of } d'\}. \tag{5}$$

By employing *constrained beam search*, the model efficiently explores the space of valid DocIDs, maintaining a beam of the most probable sequences at each decoding step while adhering to the DocID prefix constraints.

Document Relevance. The relevance between the query q and a document d is quantified by the probability of generating its corresponding DocID d' given q. This is computed as:

$$\mathcal{R}(q,d) = P(d'|q;\theta) = \prod_{i=1}^{T} P(d'_i \mid d'_{<i}, q;\theta), \tag{6}$$

where T is the length of the DocID d' in tokens, d'_i is the token at position i, and $d'_{<i}$ denotes the sequence of tokens generated before position i. The constrained beam search produces a ranked list of top-k DocIDs $\{d'^{(1)}, d'^{(2)}, \ldots, d'^{(k)}\}$ based on their generation probabilities $\{\mathcal{R}(q,d^{(1)}), \mathcal{R}(q,d^{(2)}), \ldots, \mathcal{R}(q,d^{(k)})\}$. The corresponding documents $\{d^{(1)}, d^{(2)}, \ldots, d^{(k)}\}$ are then considered the most relevant to the query q.

Model Optimization. GR models are typically optimized using cross-entropy loss, which measures the discrepancy between the generated DocID sequence and the ground truth DocID. Given a query q and its corresponding DocID d', the cross-entropy loss is defined as:

$$\mathcal{L} = -\sum_{i=1}^{T} \log P(d'_i \mid d'_{<i}, q;\theta), \tag{7}$$

where T is the length of the DocID in tokens, d'_i is the token at position i, and $d'_{<i}$ denotes the sequence of tokens generated before position i. This loss function encourages the model to learn the association between query and labeled DocID sequence.

This approach allows the GR model to produce a relevance-ordered list of documents without relying on traditional indexing structures. The core of this approach lies in leveraging the language model's capability to generate DocID sequences within prefix constraints. This section discusses the simplest GR method. In Section 3, we will delve into advanced methods from multiple perspectives, including model architectures, training strategies, and DocID design, to further enhance retrieval performance across various scenarios.

2.3 LLMs

The evolution of LLMs marks a significant leap in **Natural Language Processing (NLP)**, rooted from early statistical and neural network-based language models [372]. These models, through pre-training on vast text corpora, learned deep semantic features of language, greatly enriching the understanding of text. Subsequently, generative language models, most notably the GPT series [15, 226, 227], significantly improved text generation and understanding capabilities with improved model size and number of parameters.

LLMs can be mainly divided into two categories: encoder–decoder models and decoder-only models. Encoder–decoder models, like T5 [229] and BART [137], convert input text into vector representations through their encoder, then the decoder generates output text based on these

representations. This model perspective treats various NLP tasks as text-to-text conversion problems, solving them through text generation. On the other hand, decoder-only models, like the GPT [226] and GPT-2 [227], rely entirely on the Transformer decoder, generating text step by step through the self-attention mechanism. The introduction of GPT-3 [15], with its 175 billion parameters, marked a significant milestone in this field and led to the creation of models like InstructGPT [209], Falcon [213], PaLM [33], and Llama series [58, 283, 284]. These models, all using a decoder-only architecture, trained on large-scale datasets, have shown astonishing language processing capabilities [357].

For IR tasks, LLMs play a crucial role in directly generating the exact information users seek [54, 172, 372]. This capability marks a significant step towards a new era of GenIR. In this era, the retrieval process is not solely about locating existing information but also about creating new content that meets the specific needs of users. This feature is especially advantageous in situations where users might not know how to phrase their queries or when they are in search of complex and highly personalized information, scenarios where traditional matching-based methods fall short.

2.4 Augmented Language Models

Despite the advances of LLMs, they still face significant challenges such as hallucination, particularly in complex tasks or those requiring access to long-tail or real-time information [89, 357]. To address these issues, retrieval augmentation and tool augmentation have emerged as effective strategies. Retrieval augmentation involves integrating external knowledge sources into the language model's workflow. This integration allows the model to access up-to-date and accurate information during the generation process, thereby grounding its responses in verified data and reducing the likelihood of hallucinations [138, 250, 269]. Tool augmentation, on the other hand, extends the capabilities of LLMs by incorporating specialized tools or APIs that can perform specific functions like mathematical computations, data retrieval, or executing predefined commands [224, 243, 274]. With retrieval and tool augmentations, language models can provide more precise and contextually relevant responses, thereby improving factuality and functionality in practical applications.

Moreover, due to the aforementioned issue of hallucinations, the responses generated by LLMs are often considered unreliable because users are unaware of the sources behind the generated content, making it difficult to verify its accuracy. To enhance the credibility of responses, some studies have focused on generating responses with citations [142, 203, 254]. This approach involves enabling language models to cite the source documents of their generated content, thereby increasing the trustworthiness of the responses. All these methods are effective strategies for improving both the quality and reliability of language model outputs and are essential technologies for building the next generation of GenIR systems.

3 GR: From Similarity Matching to Generating DocIDs

In recent advancements in AIGC, GR has emerged as a promising approach in the field of information retrieval, garnering increasing interest from the academic community. Figure 3 showcases a timeline of the GR methods. Initially, GENRE [17] proposed to retrieve entities by generating their unique names through constrained beam search via a pre-built entity prefix tree, achieving advanced entity retrieval performance. Subsequently, Metzler et al. [194] envisioned a model-based IR framework aiming to combine the strengths of traditional document retrieval systems and pre-trained language models to create systems capable of providing expert-quality answers in various domains.

Following their lead, a diverse range of methods including DSI [279], DynamicRetriever [368], SEAL [12], NCI [305], and so on, have been developed, with a continuously growing body of work. These methods explore various aspects such as model training and architectures, DocIDs,

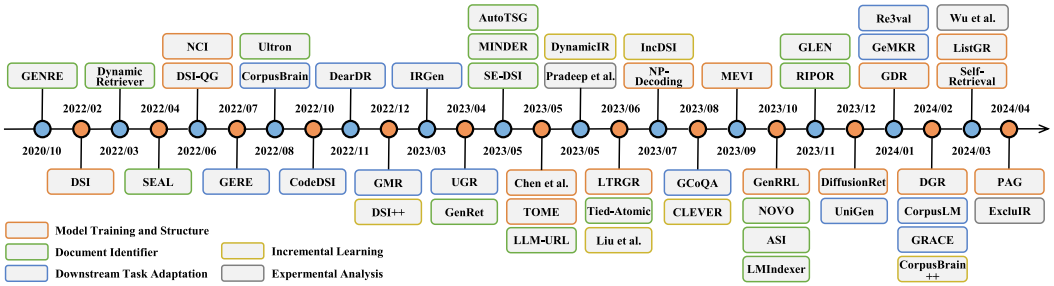

Fig. 3. Timeline of research in GR: focus on model training and structure, DocID design, incremental learning, and downstream task adaptation.

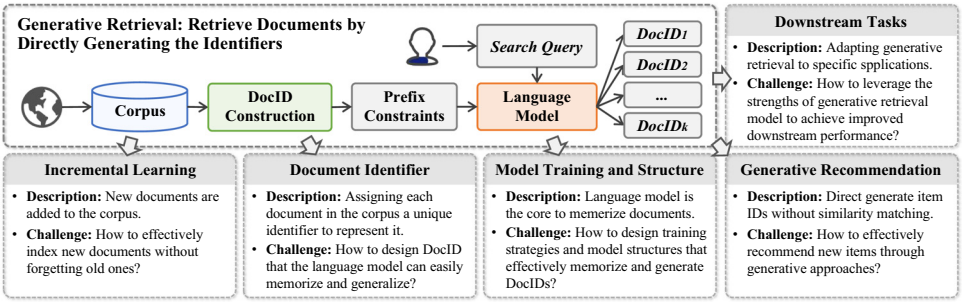

Fig. 4. A conceptual framework for a GR system, with a focus on challenges in incremental learning, identifier construction, model training and structure, and integration with downstream tasks and recommendation systems.

incremental learning, task-specific adaptation, and generative recommendations. Figure 4 presents an overview of the GR system, and we'll provide an in-depth discussion of each associated challenge in the following sections.

3.1 Model Training and Structure

One of the core components of GR is the model training and structure, aiming to enhance the model's ability to memorize documents in the corpus.

3.1.1 Model Training. To effectively train generative models for indexing documents, the standard approach is to train the mapping from queries to relevant DocIDs, based on standard **Sequence-to-Sequence (seq2seq)** training methods, as described in Equation (2). This method has been widely used in numerous GR research works, such as DSI [279], NCI [305], SEAL [12], and so on. Moreover, a series of works have proposed various model training methods tailored for GR tasks to further enhance GR retrieval performance, such as sampling documents or generating queries based on document content to serve as pseudo queries for data augmentation, or including training objectives for document ranking.

Specifically, DSI [279] proposed two training strategies: one is "indexing," that is, training the model to associate document tokens with their corresponding DocIDs, where DocIDs are pre-built based on documents in corpus, which will be discussed in detail in Section 3.2; the other is "retrieval," using labeled query-DocID pairs to fine-tune the model. Notably, DSI was the first to realize a differentiable search index based on the Transformer [288] structure, showing good performance

in web search [204] and QA [125] scenarios. Next, a series of methods propose training methods for data augmentation and improving GR model ranking ability.

Sampling Document Pieces as Pseudo Queries. In the same era, DynamicRetriever [368], also based on the encoder–decoder model, constructed a model-based IR system by initializing the encoder with a pre-trained BERT [120]. Besides, DynamicRetriever utilizes passages, sampled terms, and N-grams to serve as pseudo queries to enhance the model's memorization of DocIDs. Formally, the training methods can be summarized as follows:

$$\text{Sampled Document} : d_{s_i} \longrightarrow \text{DocID}, i \in \{1, ..., k_{d_s}\}, \tag{8}$$

$$\text{Labeled Query} : q_i \longrightarrow \text{DocID}, i \in \{1, ..., k_q\}, \tag{9}$$

where d_{s_i} and q_i denote each of the k_{d_s} sampled document text and each of the k_q labeled query for the corresponding DocID, respectively.

Generating Pseudo Queries from Documents. Following DSI, the NCI [305] model was trained using a combination of labeled query-document pairs and augmented pseudo query-document pairs. Specifically, NCI proposed two strategies: one using the DocT5Query [207] model as a query generator, generating pseudo queries for each document in the corpus through beam search; the other strategy directly uses the document as a query, as stated in Equation (8). Similarly, DSI-QG [373] also proposed using a query generator to enhance training data, establishing a bridge between indexing and retrieval in DSI. This data augmentation method has been proven in subsequent works to be an effective method to enhance the model's memorization for DocIDs, which can be expressed as follows:

$$\text{Pseudo Query} : q_{s_i} \longrightarrow \text{DocID}, i \in \{1, ..., k_{q_s}\}, \tag{10}$$

where q_{s_i} represents each of the k_{q_s} generated pseudo query for the corresponding DocID.

Improving Ranking Capability. Additionally, a series of methods focus on further optimizing the ranking capability of GR models. Chen et al. [29] proposed a multi-task distillation method to improve retrieval quality without changing the model structure, thereby obtaining better indexing and ranking capabilities. Meanwhile, LTRGR [158] introduced a ranking loss to train the model in ranking paragraphs. Subsequently, Zhou et al. [363] proposed GenRRL, which improves ranking quality through reinforcement learning with relevance feedback, aligning token-level DocID generation with document-level relevance estimation. Moreover, Li et al. [160] introduced DGR, which enhances GR through knowledge distillation. Specifically, DGR uses a cross-encoder as a teacher model, providing fine-grained passage ranking supervision signals, and then optimizes the model with a distilled RankNet loss. ListGR [278] defined positional conditional probabilities, emphasizing the importance of the generation order of each DocID in the list. In addition, ListGR employs relevance calibration that adjusts the generated list of DocIDs to better align with the labeled ranking list. See Table 1 for a detailed comparison of GR methods.

3.1.2 Model Structure. Basic GR models mostly use pre-trained encoder–decoder structured generative models, such as T5 [229] and BART [137], fine-tuned for the DocID generation task. To better adapt to the GR task, researchers have proposed a series of specifically designed model structures [129, 222, 235, 273, 305, 340, 344].

Model Decoding Methods. For the semantic structured DocID proposed by DSI [279], NCI [305] designed a prefix-aware weight-adaptive decoder. By adjusting the weights at different positions of DocIDs, this decoder can capture the semantic hierarchy of DocIDs. To allow the GR model to utilize both own parametric knowledge and external information, NP-Decoding [129] proposed using non-parametric contextualized word embeddings (as external memory) instead of traditional word embeddings as the input to the decoder. Additionally, PAG [343] proposed a planning-ahead

Table 1. Comparisons of Representative GR Methods, Focusing on DocID, Training Data Augmentation, and Training Objective

Method	DocID			Training Data Augmentation		Training Objective		
	State	Data Type	Order	Sample Doc	Doc2Query	Seq2seq	DocID	Ranking
GENRE [17]	Static	Text	Sequence	-	-	✓	-	-
DSI [279]	Static	Numeric	Sequence	✓	-	✓	-	-
DynamicRetriever [368]	Static	Numeric	Sequence	✓	-	✓	-	-
SEAL [12]	Static	Text	Sequence	✓	-	✓	-	-
DSI-QG [373]	Static	Numeric	Sequence	-	✓	✓	-	-
NCI [305]	Static	Numeric	Sequence	✓	✓	✓	-	-
Ultron [369]	Static	Numeric/Text	Sequence	✓	✓	✓	-	-
CorpusBrain [27]	Static	Text	Sequence	✓	-	✓	-	-
GenRet [263]	Learnable	Numeric	Sequence	-	✓	✓	✓	-
AutoTSG [350]	Static	Text	Set	-	✓	✓	-	-
SE-DSI [276]	Static	Text	Sequence	✓	-	✓	-	-
Chen et al. [29]	Static	Numeric	Sequence	✓	✓	✓	-	✓
LLM-URL [374]	Static	Text	Sequence	-	-	-	-	-
MINDER [159]	Static	Text	Sequence	-	✓	✓	-	-
LTRGR [158]	Static	Text	Sequence	-	✓	✓	-	✓
NOVO [309]	Learnable	Text	Set	✓	-	-	✓	-
GenRRL [363]	Static	Text	Sequence	-	✓	✓	-	✓
LMIndexer [111]	Learnable	Numeric	Sequence	-	✓	✓	✓	-
ASI [328]	Learnable	Numeric	Sequence	-	✓	✓	✓	-
RIPOR [342]	Learnable	Numeric	Sequence	-	✓	✓	✓	✓
GLEN [134]	Learnable	Text	Sequence	-	✓	✓	✓	✓
DGR [160]	Static	Text	Sequence	-	✓	✓	-	✓
ListGR [278]	Static	Numeric	Sequence	-	✓	✓	-	✓

Generative retrieval, Document Representation, Training Data Augmentation, and Training Objective

generation approach, which first decodes the set-based DocID to approximate document-level scores, and then continues to decode the sequence-based DocID on this basis.

Combining Generative and Dense Retrieval Methods. Combining seq2seq generative models with dual-encoder retrieval models, MEVI [344] utilizes **Residual Quantization (RQ)** [188] to organize documents into hierarchical clusters, enabling efficient retrieval of candidate clusters and precise document retrieval within those clusters. Similarly, Generative Dense Retrieval [340] proposed to first broadly match queries to document clusters, optimizing for interaction depth and memory efficiency, and then perform precise, cluster-specific document retrieval, boosting both recall and scalability.

Utilizing Multiple Models. TOME [235] proposed to decompose the GR task into two stages, first generating text paragraphs related to the query through an additional model, then using the GR model to generate the URL related to the paragraph. DiffusionRet [222] proposed to first use a diffusion model (SeqDiffuSeq [339]) to generate a pseudo-document from a query, where the generated pseudo-document is similar to real documents in length, format, and content, rich in semantic information; then, it employs another generative model to perform retrieval based on N-grams, similar to the process used by SEAL [12], leveraging an FM-Index [61] for generating N-grams found in the corpus. Self-Retrieval [273] fully integrated indexing, retrieval, and evaluation into a single LLM. It generates natural language indices and document segments and performs self-evaluation to score and rank the generated documents.

3.2 Design of DocIDs

Another essential component of GR is document representation, also known as DocIDs, which act as the target outputs for the GR model. Accurate document representations are crucial as they

enable the model to more effectively memorize document information, leading to enhanced retrieval performance. Table 1 provides a detailed comparison of the states, data types, and order of DocIDs across numerous GR methods. In the following sections, we will explore the design of DocIDs from two categories: numeric-based identifiers and text-based identifiers.

3.2.1 Numeric-Based Identifiers. An intuitive method to represent documents is by using a single number or a series of numbers, referred to as DocIDs. Existing methods have designed both static and learnable DocIDs.

Static DocIDs. Initially, DSI [279] introduced three numeric DocIDs to represent documents: (1) Unstructured Atomic DocID: a unique integer identifier is randomly assigned to each document, containing no structure or semantic information. (2) Naively Structured String DocID: treating random integers as divisible strings, implementing character-level DocID decoding to replace large softmax output layers. (3) Semantically Structured DocID: introducing semantic structure through hierarchical k-means method, allowing semantically similar documents to share prefixes in their identifiers, effectively reducing the search space. Concurrently, DynamicRetriever [368] also built a model-based IR system based on unstructured atomic DocID. Subsequently, Ultron [369] encoded documents into a latent semantic space using BERT [120], and compressed vectors into a smaller semantic space via **Product Quantization (PQ)** [72, 101], preserving semantic information. Each document's PQ code serves as its semantic identifier. MEVI [344] clusters documents using RQ [188] and utilizes dual-tower and seq2seq model embeddings for a balanced performance in large-scale document retrieval.

Learnable DocIDs. Unlike previous static DocIDs, GenRet [263] proposed learnable document representations, transforming documents into DocIDs through an encoder, then reconstructs documents from DocIDs using a decoder, trained to minimize reconstruction error. Furthermore, it used progressive training and diversity clustering for optimization. To ensure that DocID embeddings can reflect document content, Tied-Atomic [205] proposed to link document text with token embeddings and employs contrastive loss for DocID generation. LMIndexer [111] and ASI [328] learned optimal DocIDs through semantic indexing, with LMIndexer using a reparameterization mechanism for unified optimization, facilitating efficient retrieval by aligning semantically similar documents under common DocIDs. ASI extends this by establishing an end-to-end retrieval framework, incorporating semantic loss functions and reparameterization to enable joint training. Furthermore, RIPOR [342] treats the GR model as a dense encoder to encode document content. It then splits these representations into vectors via RQ [188], creating unique DocID sequences. Furthermore, RIPOR implements a prefix-guided ranking optimization, increasing relevance scores for prefixes of pertinent DocIDs through margin decomposed pairwise loss during decoding.

In summary, numeric-based document representations can utilize the embeddings of dense retrievers, obtaining semantically meaningful DocID sequences through methods such as k-means, PQ [101], and RQ [188]; they can also combine encoder–decoder GR models with bi-encoder DR models to achieve complementary advantages [205, 344].

3.2.2 Text-Based Identifiers. Text-based DocIDs have the inherent advantage of effectively leveraging the strong capabilities of pre-trained language models and offering better interpretability.

Document Titles. The most straightforward text-based identifier is the document title, which requires each title to uniquely represent a document in the corpus, otherwise, it would not be possible to accurately retrieve a specific document. The Wikipedia corpus used in the KILT [216] benchmark, due to its well-regulated manual annotation, has a unique title corresponding to each document. Thus, GENRE [17], based on the title as DocID and leveraging the generative model BART [137] and pre-built DocID prefix, achieved superior retrieval performance across 11 datasets in KILT. Following GENRE, GERE [26], CorpusBrain [27], Re3val [255], and CorpusBrain++ [79]

also based their work on title DocIDs for Wikipedia-based tasks. Notably, LLM-URL [376] directly generated URLs using ChatGPT prompts, achieving commendable performance after removing invalid URLs. However, in the web search scenario [204], document titles in the corpus often have significant duplication and many meaningless titles, making it unfeasible to use titles alone as DocIDs. Thus, Ultron [369] effectively addressed this issue by combining URLs and titles as DocIDs, identifying documents through keywords in web page URLs and titles.

Sub-Strings of Documents. To increase the flexibility of DocIDs, SEAL [12] proposed a sub-string identifier, representing documents with any N-grams within them. Using FM-Index (a compressed full-text sub-string index) [61], SEAL could generate N-grams present in the corpus to retrieve all documents containing those N-grams, scoring and ranking documents based on the frequency of N-grams in each document and the importance of N-grams. Following SEAL, various GR models [25, 158–160] also utilized sub-string DocIDs and FM-Index during inference. For a more comprehensive representation of documents, MINDER [159] proposed multi-view identifiers, including generated pseudo queries from document content via DocT5Query [207], titles, and sub-strings. This multi-view DocID was also used in LTRGR [158] and DGR [160].

Term Sets. Unlike the sequential DocIDs described earlier, AutoTSG [350] proposed a term set-based document representation, using keywords extracted from titles and content, rather than predefined sequences, allowing for retrieval of the target document as long as the generated term set is included in the extracted keywords. Recently, PAG [343] also constructed DocIDs based on sets of key terms, disregarding the order of terms, which is utilized for approximating document relevance in decoding.

Learnable DocIDs. Text-based identifiers can also be learnable. Similarly based on term sets, NOVO [309] proposed learnable continuous N-grams constituting term-set DocIDs. Through denoising query modeling, the model learned to generate queries from documents with noise, thereby implicitly learning to filter out document N-grams more relevant to queries. NOVO also improves the document's semantic representation by updating N-gram embeddings. Later, GLEN [134] uses dynamic lexical DocIDs and follows a two-phase index learning strategy. First, it assigns DocIDs by extracting keywords from documents using self-supervised signals. Then, it refines DocIDs by integrating query-document relevance through two loss functions. During inference, GLEN ranks documents using DocID weights without additional overhead.

3.3 Incremental Learning on Dynamic Corpora

Prior studies have focused on GR from static document corpora. However, in reality, the documents available for retrieval are continuously updated and expanded. To address this challenge, researchers have developed a range of methods to optimize GR models for adapting to dynamic corpora.

Optimizer and Document Rehearsal. At first, DSI++ [191] aims to address the incremental learning challenges encountered by DSI [279]. DSI++ modifies the training by optimizing flat loss basins through the Sharpness-Aware Minimization optimizer, stabilizing the learning process of the model. It also employs DocT5Query [207] to generate pseudo queries for documents in the existing corpus as training data augmentation, mitigating the forgetting issue of GR models.

Constrained Optimization. Addressing the scenario of real-time addition of new documents, such as news or scientific literature IR systems, IncDSI [123] views the addition of new documents as a constrained optimization problem to find optimal representations for the new documents. This approach aims to (1) ensure new documents can be correctly retrieved by their relevant queries, and (2) maintain the retrieval performance of existing documents unaffected.

Incremental PQ (IPQ). CLEVER [24], based on PQ [101], proposes IPQ for generating PQ codes as DocIDs for documents. Compared to traditional PQ methods, IPQ designs two adaptive thresholds to update only a subset of centroids instead of all, maintaining the indices of updated centroids

constant. This method reduces computational costs and allows the system to adapt flexibly to new documents.

Fine-Tuning Adapters for Specific Tasks. CorpusBrain++ [79] introduces the KILT++ benchmark for continuously updated KILT [216] tasks and designs a dynamic architecture paradigm with a backbone-adapter structure. By fixing a shared backbone model to provide basic retrieval capabilities while introducing task-specific adapters to incrementally learn new documents for each task, it effectively avoids catastrophic forgetting. During training, CorpusBrain++ generates pseudo queries for new document sets and continues to pre-train adapters for specific tasks.

3.4 Downstream Task Adaption

GR methods, apart from addressing retrieval tasks individually, have been tailored to various downstream generative tasks. These include fact verification [282], entity linking [85], open-domain QA [125], dialogue [50], slot filling [136], among others, as well as knowledge-intensive tasks [216], code [178], conversational QA [3], and multi-modal retrieval scenarios [164], demonstrating superior performance and efficiency. These methods are discussed in terms of separate training, joint training, and multi-modal GR.

3.4.1 Separate Training. For fact verification tasks [282], which involve determining the correctness of input claims, GERE [26] proposed using an encoder–decoder-based GR model to replace traditional indexing-based methods. Specifically, GERE first utilizes a claim encoder to encode input claims and then generates document titles related to the claim through a title decoder to obtain candidate sentences for corresponding documents.

Knowledge-Intensive Language Tasks (KILT). For KILT [216], CorpusBrain [27] introduced three pre-training tasks to enhance the model's understanding of query-document relationships at various granularities: Internal Sentence Selection, Leading Paragraph Selection, and Hyperlink Identifier Prediction. Similarly, UGR [25] proposed using different granularities of N-gram DocIDs to adapt to various downstream tasks, unifying different retrieval tasks into a single generative form. UGR achieves this by letting the GR model learn prompts specific to tasks, generating corresponding document, passage, sentence, or entity identifiers.

Furthermore, DearDR [281] utilizes remote supervision and self-supervised learning techniques, using Wikipedia page titles and hyperlinks as training data. The model samples sentences from Wikipedia documents as input and trains a self-regressive model to decode page titles or hyperlinks, or both, without the need for manually labeled data. Re3val [255] proposes a retrieval framework combining generative reordering and reinforcement learning. It first reorders retrieved page titles using context information obtained from a dense retriever, then optimizes the reordering using the REINFORCE algorithm to maximize rewards generated by constrained decoding.

Multi-Hop Retrieval. In multi-hop retrieval tasks, which require iterative document retrievals to gather adequate evidence for answering a query, GMR [130] proposed to employ language model memory and multi-hop memory to train a GR model, enabling it to memorize the target corpus and simulate real retrieval scenarios through constructing pseudo multi-hop query data, achieving dynamic stopping and efficient performance in multi-hop retrieval tasks.

Code Retrieval. CodeDSI [202] is an end-to-end generative code search method that directly maps queries to pre-stored code samples' DocIDs instead of generating new code. Similar to DSI [279], it includes indexing and retrieval stages, learning to map code samples and real queries to their respective DocIDs. CodeDSI explores different DocID representation strategies, including direct and clustered representation, as well as numerical and character representations.

Conversational QA. GCoQA [157] is a GR method for conversational QA systems that directly generates DocIDs for passage retrieval. This method focuses on key information in the dialogue

context at each decoding step, achieving more precise and efficient passage retrieval and answer generation, thereby improving retrieval performance and overall system efficiency.

3.4.2 Joint Training. The methods in the previous section involve separately training generative retrievers and downstream task generators. However, due to the inherent nature of GR models as generative models, a natural advantage lies in their ability to be jointly trained with downstream generators to obtain a unified model for retrieval and generation tasks.

Multi-Decoder Structure. UniGen [154] proposes a unified generation framework to integrate retrieval and QA tasks, bridging the gap between query input and generation targets using connectors generated by LLMs. UniGen employs shared encoders and task-specific decoders for retrieval and QA, introducing iterative enhancement strategies to continuously improve the performance of both tasks.

Multi-Task Training. Later, CorpusLM [151] introduces a unified language model that integrates GR, closed-book generation, and **Retrieval-Augmented Generation (RAG)** to handle various knowledge-intensive tasks. The model adopts a multi-task learning approach and introduces ranking-guided DocID decoding strategies and continuous generation strategies to improve retrieval and generation performance. In addition, CorpusLM designs a series of auxiliary DocID understanding tasks to deepen the model's understanding of DocID semantics.

3.4.3 Multi-Modal GR. GR methods can also leverage multi-modal data such as text, images, and so on, to achieve end-to-end multi-modal retrieval.

Tokenizing Images to DocID Sequences. At first, IRGen [355] transforms image retrieval problems into generative problems, predicting relevant discrete visual tokens, i.e., image identifiers, through a seq2seq model given a query image. IRGen proposed a semantic image tokenizer, which converts global image features into short sequences capturing high-level semantic information.

Advanced Model Training and Structure. Later, GeMKR [177] combines LLMs' generation capabilities with visual-text features, designing a generative knowledge retrieval framework. It first guides multi-granularity visual learning using object-aware prefix tuning techniques to align visual features with LLMs' text feature space, achieving cross-modal interaction. GeMKR then employs a two-step retrieval process: generating knowledge clues closely related to the query and then retrieving corresponding documents based on these clues. GRACE [177] achieves generative cross-modal retrieval method by assigning unique identifier strings to images and training multi-modal LLMs [7] to memorize the association between images and their identifiers. The training process includes (1) learning to memorize images and their corresponding identifiers, and (2) learning to generate the target image identifiers from textual queries. GRACE explores various types of image identifiers, including strings, numbers, semantic and atomic identifiers, to adapt to different memory and retrieval requirements.

3.4.4 Generative Recommender Systems. Recommendation systems, as an integral part of the information retrieval, are currently undergoing a paradigm shift from discriminative models to generative models. Generative recommendation systems do not require the computation of ranking scores for each item followed by database indexing, but instead accomplish item recommendations through the direct generation of IDs. In this section, several seminal works, including P5 [73], GPT4Rec [145], TIGER [231], SEATER [252], IDGenRec [271], LC-Rec [358], and ColaRec [307], are summarized to outline the development trends in generative recommendations.

P5 [73] transforms various recommendation tasks into different natural language sequences, designing a universal, shared framework for recommendation completion. This method, by setting unique training objectives, prompts, and prediction paradigms for each recommendation domain's downstream tasks, serves well as a backbone model, accomplishing various recommendation tasks

through generated text. In GR, effective indexing identifiers have been proven to significantly enhance the performance of generative methods. Similarly, TIGER [231] initially learns a residual quantized autoencoder to generate semantically informative indexing identifiers for different items. It then trains a transformer-based encoder-decoder model with this semantically informative indexing identifier sequence to generate item identifiers for recommending the next item based on historical sequences.

Focusing solely on semantic information and overlooking the collaborative filtering information under the recommendation context might limit the further development of generative models. Therefore, after generating semantic indexing identifiers similar to TIGER using a residual quantized autoencoder with uniform semantic mapping, LC-Rec [358] also engages in a series of alignment tasks, including sequential item prediction, explicit index-language alignment, and recommendation-oriented implicit alignment. Based on the learned item identifiers, it integrates semantic and collaborative information, enabling LLMs to better adapt to sequence recommendation tasks.

IDGenRec [271] innovatively combines generative recommendation systems with LLMs by using human language tokens to generate unique, concise, semantically rich and platform-agnostic textual identifiers for recommended items. The framework includes a text ID generator trained on item metadata with a diversified ID generation algorithm, and an alternating training strategy that optimizes both the ID generator and the LLM-based recommendation model for improved performance and accuracy in sequential recommendations. SEATER [252] designs a balanced k-ary tree-structured indexes, using a constrained k-means clustering method to recursively cluster vectors encoded from item texts, obtaining equal-length identifiers. Compared to the method proposed by DSI [279], this balanced k-ary tree index maintains semantic consistency at every level. It then trains a Transformer-based encoder-decoder model and enhances the semantics of each level of indexing through contrastive learning and multi-task learning. ColaRec [307] integrates collaborative filtering signals and content information by deriving generative item identifiers from a pre-trained recommendation model and representing users via aggregated item content. Then it uses an item indexing generation loss and contrastive loss to align content-based semantic spaces with collaborative interaction spaces, enhancing the model's ability to recommend items in an end-to-end framework.

4 Reliable Response Generation: Direct Information Accessing with Generative Language Models

The rapid advancement of LLMs has positioned them as a novel form of IR system, capable of generating reliable responses directly aligned with users' informational needs. This not only saves the time users would otherwise spend on collecting and integrating information but also provides personalized, user-centric answers tailored to individual users.

However, challenges remain in creating a grounded system that delivers faithful answers, such as hallucination, prolonged inference time, and high operational costs. This section will outline strategies for constructing a faithful GenIR system by: (1) optimizing the GenIR model internally, (2) enhancing the model with external knowledge, (3) increasing accountability, and (4) developing personalized information assistants.

4.1 Internal Knowledge Memorization

To develop a user-friendly and reliable IR system, the generative model should be equipped with comprehensive internal knowledge. Optimization of the backbone generative model can be categorized into three aspects: structural enhancements, training strategies, and inference techniques. The overview of this section is shown in the green part of Figure 5.

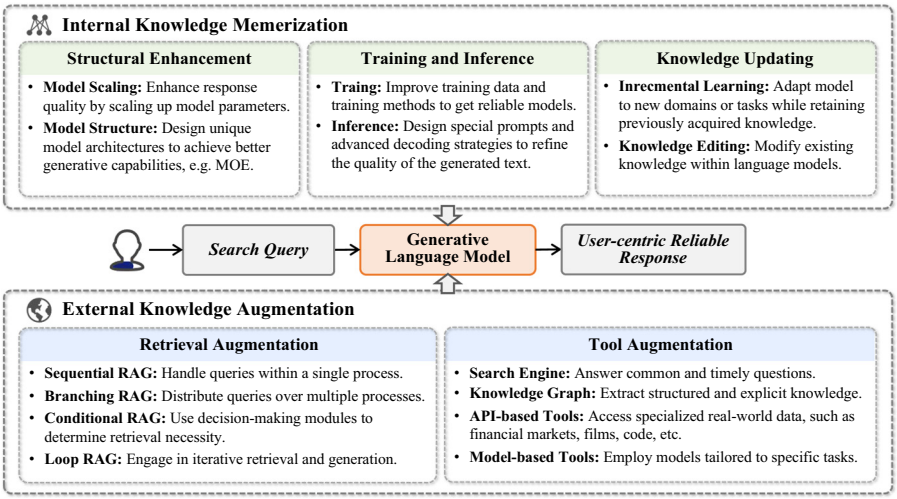

Fig. 5. An illustration of strategies for enhancing language models to generate user-centric and reliable responses, including model internal knowledge memorization and external knowledge augmentation.

4.1.1 Model Structure. With the advent of generative models, various methods have been introduced to improve model structure and enhance generative reliability. We aim to discuss the crucial technologies contributing to this advancement in this subsection.

(1) *Model Scaling.* Model parameter scaling is a pivotal factor influencing performance. Contemporary language models predominantly employ the Transformer architecture, and scaling both the model parameters and the training data enhances the model's capacity to retain knowledge and capabilities [115]. For instance, in the GPT [2, 15, 226, 227] series and LLaMA [283, 284] family, larger models tend to perform better on diverse downstream tasks, including few-shot learning, language understanding, and generation [33]. Additionally, scaling the model contributes to improved instruction-following capabilities [225], enabling a more adept comprehension of user intent and generating responses that better satisfy user requests.

(2) *Model Integration.* Model integration is an effective method to enhance the reliability of generated outputs by capitalizing on the diverse strengths of various models. The predominant approach is the Mixture of Experts [95], which utilizes a gating mechanism to selectively activate sections of network parameters during inference, greatly increasing the effective parameters without inflating inference costs [57, 60, 105, 135]. This method also boasts impressive scalability, with efficacy augmented alongside the expanding parameter volume and the number of expert models [37]. Alternatively, the LLM-Blender framework [106] employs a ranker and a fuser to combine answers from various LLMs, including black-box models, but faces high deployment costs.

4.1.2 Training and Inference. In the model training stage, methods to enhance the reliability of answers can be categorized into two aspects: training data optimization and training methods optimization.

(1) *Training Data Optimization.* The quality of training data substantially affects the reliability of model outputs. Noise, misinformation, and incomplete information can disrupt the learning process, leading to hallucinations and other issues. To address this, Gunasekar et al. [78] used GPT-3.5 to artificially create textbooks filled with examples and language descriptions as training data, resulting in significant improvements on downstream tasks after minor fine-tuning. LIMA [361] used dialogues from community forums to construct a small-scale fine-tuning dataset, enhancing the

model's conversation capabilities during the alignment phase. To reduce redundancies in crawled internet data, Lee et al. [131] combined suffix arrays [187], and MinHash [14] to approximate matching and deduplicate the training dataset, reducing direct reproduction from the same source.

(2) *Training Methods Optimization*. Beyond conventional training methods, additional techniques have been proposed to improve the factuality of model outputs. MixCL [262] incorporates contrastive learning into the training objective, using an external knowledge base to identify correct snippets and reduce the probability of generating incorrect tokens, thus enhancing model reliability. CaliNet [55] utilizes a contrastive method to assess erroneous knowledge learned by the model and fine-tunes the parameters of the FFN layer to rectify these errors. FactTune [210] incorporates factuality assessment during the RLHF phase, using automatic evaluation methods like FactScore [197] to rank outputs and employing DPO [228] to teach the model factuality preference.

Apart from enhancing the internal knowledge reliability during training, the inference stage significantly impacts the reliability of answers. The overall inference process consists of user input and the model's token decoding, and approaches to increase generation reliability can be divided into prompt engineering and decoding strategy.

(3) *Prompt Engineering*. Prompting methods play a vital role in guiding the model. A well-designed prompt can better promote the model's internal capabilities to provide more accurate answers. The **Chain-of-Thought (CoT)** [311] prompting method guides the model to explicitly decompose the question into a reasoning chain during decoding, improving response accuracy by grounding the final answer on accurate intermediate steps. Further, CoT-SC [304] samples multiple answers and chooses the most consistent one as the final answer. The Tree of Thoughts [330] expands CoT's single reasoning path to multiple paths, synthesizing their outcomes to arrive at the final answer. The Chain-of-Verification [48] introduces a self-reflection mechanism where the LLM generates a draft response, then validates each statement for factual inaccuracies, correcting errors to enhance factual accuracy. Additionally, methods like RECITE [266] and GenRead [337] prompt the model to output relevant internal knowledge fragments, which are then used to bolster the QA process.

(4) *Decoding Strategy*. Decoding strategies are another critical factor influencing the reliability of model-generated responses. An appropriate decoding method can maintain the reliability and diversity of a model's response. Nucleus Sampling [132] samples within a set probability range for tokens, ensuring better diversity while balancing variety and reliability. Building on this, Factual-Nucleus Sampling [133] employs a dynamic, decaying threshold for token sampling, ensuring later tokens are not influenced by earlier less factual tokens. Wan et al. [290] proposed a faithfulness-aware decoding method to enhance the faithfulness of the beam-search approach by incorporating a Ranker to reorder generated sequences and a lookahead method to avoid unfaithful tokens.

Apart from directly modifying the decoding method, several studies influence the decoding distribution by leveraging hidden layer information. DoLa [36] uses distributional differences between hidden and output layers to prioritize newly learned factual knowledge or key terms, increasing their generation likelihood. Inference-Time Intervention [146] identifies attention heads strongly correlated with response correctness, adjusts their orientations, and moderates their activation, achieving more truthful generation with minimal model interference. Shi et al. [249] proposed CAD, comparing output distributions before and after adding extra information, reducing reliance on the model's own knowledge to avoid conflicts leading to inaccuracies.

4.1.3 Knowledge Updating. In real-life scenarios, information is constantly evolving, and therefore, the GenIR system needs to continuously acquire the latest knowledge to meet users'

information needs. Since the model's knowledge storage is limited, knowledge updating is necessary to ensure more reliable generated responses. In this section, we will discuss existing methods for knowledge updating from two perspectives: incremental learning and knowledge editing.

(1) *Incremental Learning*. Incremental learning refers to the ability of machine learning models to continuously learn new skills and tasks while retaining previously acquired knowledge [299, 301, 319, 349]. In the GenIR system, it is crucial to enable the language model to memorize the latest information while preventing the forgetting of previous knowledge.

One approach is *Incremental Pre-Training*, which does not rely on supervised data but continues pre-training on continuously updated corpora to alleviate catastrophic forgetting. For example, Baidu proposed the ERNIE 2.0 framework [265], enhancing language understanding through continuous multi-task learning. Jang et al. [99] introduced **Continual Knowledge Learning (CKL)** to explore how LLMs update and retain knowledge amidst rapidly changing information, creating benchmarks like FUAR. Cossu et al. [38] studied continual pre-training for language and vision, finding that self-supervised or unsupervised methods are more effective in retaining previous knowledge compared to supervised learning. Additionally, Ke et al. [118] proposed Domain Adaptive Pre-training to improve the model's adaptability to new domains while preventing forgetting using techniques like soft masking and contrastive learning. For domain-specific model construction, Xie et al. [321] introduced FinPythia-6.9B, an efficient continual pre-training method specifically designed for large-scale language models in the financial domain.

On the other hand, *Incremental Fine-tuning* utilizes only labeled data for training. Progressive Prompts [234] appends new soft prompts for each new task, facilitating knowledge transfer and reducing forgetting. DynaInst [200] enhances lifelong learning in pre-trained language models through parameter regularization and experience replay, employing dynamic instance and task selection for efficient learning under resource constraints. Jang et al. [98] challenge traditional multi-task prompt fine-tuning by refining expert models on individual tasks. Suhr et al. [258] introduce a feedback-driven continual learning approach for instruction-following agents, where natural language feedback is converted into immediate rewards via contextual bandits to optimize learning. O-LoRA [303] achieves superior continual learning by training new tasks in orthogonal low-rank subspaces, significantly minimizing task interference. Peng et al. [214] propose a scalable language model that dynamically adjusts parameters based on task requirements, effectively preventing the forgetting of previously learned tasks.

(2) *Knowledge Editing*. Knowledge editing refers to the process of modifying and updating existing knowledge within language models [190, 301], distinct from incremental learning that focuses on adapting to new domains or tasks. By editing the weights or layers of a model, knowledge editing methods can correct erroneous facts and incorporate new knowledge, making it important before deploying GenIR systems. There are primarily three paradigms for internal knowledge editing within language models: adding trainable parameters, locate-then-edit, and meta-learning.

One method of *Adding Trainable Parameters* is by integrating new single neurons (patches) in the final **Feed-Forward Neural Network (FFN)** layer, as in T-Patcher [93] and CaliNet [55], which serve as trainable parameters to adjust the model's behavior. Alternatively, discrete code-book modules are introduced in the middle layers of the language model, as in GRACE [82], to adjust and correct information.

Moreover, the *Locate-then-Edit* method first identifies the parameters corresponding to specific knowledge and then updates these targeted parameters directly. Common techniques involve identifying key-value pairs in the FFN matrix, known as "knowledge neurons," and updating them [44]. Techniques like ROME [192] use causal mediation analysis to pinpoint areas needing editing, and MEMIT [193] builds on ROME to implement synchronized editing in various scenarios.

Table 2. Comparison of Representative Reliable Response Generation Methods, Considering Model Configurations, Specializations, and Evaluations

Method	Model Configuration			Target Domain	
	Backbone	Parameters	Trained	Capability	Evaluation Task
GPT-3 [15]	Transformer	175B	✓	General	General Tasks (LM, QA, Reasoning, ...)
Llama-3.1 [58]	Transformer	8B/70B/405B	✓	General	General Tasks
Mistral [104]	Transformer	7B/22B/123B	✓	General	General Tasks
PaLM [33]	Transformer	540B	✓	General	General Tasks
FactTune [210]	Llama-2	7B	✓	Factuality	Domain-Specific QA
GenRead [337]	InstructGPT	175B	×	Factuality	Knowledge-Intensive Tasks
DoLa [36]	LLaMA	7B$\check{6}$5B	×	Factuality	Multi-Choice QA, Open-Ended Generation
RAG [138]	BART	400M	✓	Factuality	Knowledge-Intensive Tasks
REPLUG [250]	GPT-3	175B	×	Factuality	LM, Multi-Choice QA, ODQA
FLARE [110]	GPT-3	175B	×	Factuality	Knowledge-Intensive Tasks
Self-RAG [5]	Llama-2	7B/13B	✓	Factuality	ODQA, Reasoning, Fact Check.
IR-CoT [285]	GPT-3/Flan-T5	175B/11B	×	Factuality	Multi-Hop QA
ReAct [331]	PaLM	540B	×	Tools	Multi-Hop QA, Fact Check., Decision Making
StructGPT [109]	GPT-3/GPT-3.5	175B/-	×	Tools	KG-Based QA, Table-Based QA, Text-to-SQL
ToolFormer [243]	GPT-J	6B	✓	Tools	LM, Math, QA, Temporal Tasks
ToolLLM [224]	LLaMA	7B	✓	Tools	Tool Use
HuggingGPT [248]	GPT-3.5	-	×	Tools	Various Complex AI Tasks
According to [312]	GPT-3/Flan-T5/...	175B/11B/...	×	Accountability	ODQA
IFL [128]	GPT-J	6B	✓	Accountability	Long-Form QA
WebGPT [203]	GPT-3	175B	✓	Accountability	Long-Form QA
WebBrain [221]	BART	400M	✓	Accountability	Long-Form QA
RARR [69]	PaLM	540B	×	Accountability	ODQA, Reasoning, Conversational QA
SearChain [324]	GPT-3.5	-	×	Accountability	Knowledge-Intensive Tasks
P2Bot [171]	Transformer	-	✓	Personalization	Personalized Dialogue
P-Soups [97]	Tulu	7B	✓	Personalization	Personalized Dialogue
OPPU [272]	Llama-2	7B	✓	Personalization	Language Model Personalization Tasks
Zhongjing [11]	Ziya-LLaMA	13B	✓	Healthcare	Chinese Medical Dialogue
Mental-LLM [325]	Alpaca/GPT-3.5/...	7B/-/...	✓/×	Healthcare	Mental Health Reasoning Tasks
Edu-Chat [47]	LLaMA	13B	✓	Education	ODQA, Education Tasks

LM, Language Modeling; ODQA, Open-Domain QA.

Methods such as PMET [153] employ attention mechanisms for editing, while BIRD [181] introduces a bidirectional inverse relation modeling approach.

Meta-Learning, another paradigm, uses hyper-networks to generate the necessary updates for model editing. Knowledge Editor [16] predicts weight updates for each data point using a hyper-network. MEND [198], by taking low-order decomposition of gradients as input, learns to rapidly edit language models to enhance performance. Additionally, MALMEN [268] separates the computations of hyper-networks and language models, facilitating the editing of multiple facts under a limited memory budget. These meta-learning mechanisms enable models to swiftly adapt to new knowledge and tasks. A detailed comparison of representative reliable response generation methods is provided in Table 2.

4.2 External Knowledge Augmentation

Although LLMs have demonstrated significant effectiveness in response generation, issues such as susceptibility to hallucinations, difficulty handling in-domain knowledge, and challenges with knowledge updating persist. Augmenting the model's generative process with external knowledge sources can serve as an effective way to tackle these issues. Based on the form of external knowledge employed, these approaches can be classified into retrieval augmentation and tool augmentation. The blue area in Figure 5 provides an overview of this section.

4.2.1 Retrieval Augmentation. RAG enhances the response of generative models by combining them with a retrieval mechanism [94, 138, 366]. By querying a large collection of documents, information that is relevant to the input query can be fetched and integrated into the input of the generative model. RAG enables generative models to be grounded in existing reliable knowledge, significantly improving the reliability of model generation. Typically, an RAG method involves a retriever and a generator. Based on the interaction flow between these two, RAG methods can be divided into four categories [71].

(1) *Sequential RAG.* Sequential RAG operates on a linear progression, where the retriever first retrieves relevant information and the generator utilizes this information to directly complete the response generation process.

The basic form of sequential RAG is a "Retrieve-Read" framework [182], where early works perform joint [13, 80, 138] or separate [94] training of retriever and generator but require costly pre-training. In-Context RALM [232] addresses this by directly using retrieved documents as input, leveraging the model's in-context learning without additional training.

With the widespread adoption of LLMs, most subsequent works are built on the foundation of a frozen generator. AAR [338] fine-tunes a general retriever to adapt to the information acquisition preferences of the generative model. LLM-embedder [351] uses rewards produced by LLM to train an embedding model dedicated to retrieval augmentation. ARL2 [348] leverages LLM to annotate relevance scores in the training set and trains a retriever using contrastive learning.

Several works introduce pre-retrieval and post-retrieval processes [71] into the sequential pipeline to enhance the overall efficiency. In the pre-retrieval process, the RRR model [182] introduces a rewriter module before the retriever, trained using the generator's feedback to enable the retrieval system to provide more suitable information for generation.

In the post-retrieval process, information compressors are proposed to filter out irrelevant content from documents, avoiding misleading the generator's response [42, 113, 169]. RECOMP [323] uses both abstractive and extractive compressors to generate concise summaries of retrieved documents. LLMLingua [108] retains important tokens by calculating token importance based on the **Perplexity (PPL)** provided by the generative model. LongLLMLingua [107] introduces query-aware compression and reranks retrieved documents based on importance scores to alleviate the "loss in the middle" phenomenon [169]. PRCA [327] employs reinforcement learning to train a text compressor adaptable to black-box LLMs and various retrievers, serving as a versatile plug-in.

(2) *Branching RAG.* In the Branching RAG framework, the input query is processed across multiple pipelines, and each pipeline may involve the entire process in the sequential pipeline. The outputs from all pipelines are merged to form the final response, allowing for finer-grained handling of the query or retrieval results.

In the pre-retrieval stage, TOC [122] uses few-shot prompting to recursively decompose complex questions into clear sub-questions in a tree structure, retrieving relevant documents for each and generating a comprehensive answer. BlendFilter [295] enhances the original query using prompts with internal and external knowledge, retrieves related documents with the augmented queries, and merges them for a comprehensive response.

In the post-retrieval stage, REPLUG [250] processes each retrieved document with the query through the generator separately and combines the resulting probability distributions to form the final prediction. GenRead [337] prompts LLM to generate related documents and merges them with retrieved documents from the retriever as input, enhancing content coverage.

(3) *Conditional RAG.* The Conditional RAG framework adapts to various query types through distinct processes, improving the system's flexibility. Since there can be knowledge conflict between the knowledge from retrieved documents and the generator's own knowledge, RAG's effectiveness isn't consistent across all scenarios. To address this, common conditional RAG methods

include a decision-making module that determines whether to engage the retrieval process for each query.

SKR [306] trains a binary classifier on a dataset of questions LLMs can or cannot answer, determining at inference whether to use retrieval. Training labels are obtained by prompting the model to assess if external knowledge is needed. Self-DC [294] uses the model's confidence score to decide on retrieval necessity, categorizing queries into unknown, uncertain, and known, with unknown queries processed through sequential RAG and uncertain ones decomposed into sub-questions. Rowen [51] introduces a multilingual detection module that perturbs the original question and measures response consistency to decide on retrieval.

(4) *Loop RAG.* Loop RAG involves deep interactions between the retriever and generator components. Owing to multi-turn retrieval and generation processes, accompanied by comprehensive interactions, it excels at handling complex and diverse input queries, yielding superior results in response generation.

ITER-RETGEN [246] introduces an iterative framework alternating between RAG and generation-augmented retrieval, repeating this process to produce the final answer. IR-COT [285] follows a similar procedure to ITER-RETGEN but the iteration pauses based on the model's own generative process. FLARE [110] conducts concurrent retrieval during generation, evaluating the need for retrieval at each new sentence based on the LLM's confidence score, dynamically supplementing information to enhance content reliability. COG [127] models generation as continual retrieval and copying of segments from an external corpus, with the generator producing conjunctions to maintain fluency. Self-RAG [5] adds special tokens into the vocabulary, allowing the generator to decide on retrieval, document importance, and whether to perform a critique.

Some works focus on deconstructing complex inquiries into sub-questions, addressing these individually to produce a more dependable response. Press et al. [219] guide LLM to decompose complex questions into sub-questions, answer each using retrieved results, and synthesize the answers; RET-Robust [336] builds upon this by incorporating a **Natural Language Inference (NLI)** model to verify retrieved documents support the sub-question answers, reducing misinformation.

4.2.2 Tool Augmentation. Although retrieval-augmented techniques have significantly improved upon the blind spots of a generator's self-knowledge, these methods struggle with the rapid and flexible update of information since they rely on the existence of information within an external corpus of documents. Tool augmentation, on the other hand, excels in addressing this issue by invoking various tools that allow for the timely acquisition and usage of the latest data, including finance, news, and more. Moreover, tool augmentation expands the scope of responses a model can offer, such as language translation, image generation, and other tasks, to more comprehensively meet users' IR needs.

There are four categories of tools that can be utilized to construct a more reliable IR system:

(1) *Search Engine.* Common search engine tools like Google Search and Bing Search help answer frequent and time-sensitive queries effectively. Self-Ask [219] initially decomposes complex questions into multiple sub-questions, then uses a search engine to answer each sub-question, and finally generates a comprehensive answer to the complex question. ReAct [331] embeds search engine calls into the model's reasoning process, allowing the generative model to determine when to make calls and what queries to input for more flexible reasoning. New Bing can automatically search relevant information from Bing based on user input, yielding reliable and detailed answers, including citation annotations in the generated content.

Some works have also built advanced conversational systems based on tools like search engines. Internet-Augmented Generation [124] enhances the quality of conversational replies by using search engines during conversations. LaMDA [280] and BlenderBot [251] combine search engines with

conversational agents, constantly accessing internet information to enrich conversation factualness. WebGPT [203] and WebCPM [223] directly teach models to perform human-like browser operations by generating commands such as Search, Click, and Quote, facilitating the automated retrieval and acquisition of information.

(2) ***Knowledge Graph (KG)***. Compared to search engines, KGs are particularly useful for extracting structured, explicit knowledge. Relevant knowledge from a KG can be extracted and used as a prompt input to enhance the generative process [260]. StructGPT [109] introduces an iterative reading-and-reasoning framework where the model can access a KG through a well-designed interface, continually acquiring information and reasoning until an answer is obtained. RoG [180] generates plausible reasoning paths from a KG, executes them in parallel, and integrates outcomes for a final answer; ToG [260] allows the model to explore entities and links without pre-planning paths, continuously assessing reasoning feasibility.

(3) *API-Based Tools*. An important part of the tools is the real-world APIs, which enable the model to obtain information from specific data sources, such as real-time stock information, movie services, code interpreters, and so on. However, the multitude and diversity of APIs, coupled with the adherence to certain operational protocols, make the teaching of API usage to models a focal point of this area.

Toolformer [243] trains language models in a self-supervised manner to automatically call APIs when needed, using prompts to generate API calls, executing them, and filtering ineffective ones to form the dataset. Training with standard language modeling objectives yields models that can autonomously invoke APIs across tasks without losing language modeling capabilities. RestGPT [256] formulates a framework prompting LLMs to invoke RESTful APIs, comprising an online planner, an API selector, and an executor. ToolLLM [224] uses a large corpus of scraped APIs to build a dataset for fine-tuning. Gorilla [212] introduces an information retriever providing the model with reference API documentation, facilitating retrieval-based information utilization during fine-tuning. ToolkenGPT [81] incorporates each tool as a new token into the vocabulary, enabling the model to invoke APIs during inference as naturally as generating text.

Beyond learning to invoke APIs, CREATOR [220] prompts models to write code based on actual problems as new tool implementations, with generated tools functioning through a code interpreter and demonstrating impressive results on complex tasks.

Some works additionally support multi-modal inputs, broadening the application scope of the models. AssistGPT [68] offers a framework including modules like Planner, Executor, Inspector, and Learner, utilizing language and code for intricate inference. ViperGPT [267] feeds CodeX with user queries and visual API information to generate Python code invoking APIs, successfully completing complex visual tasks.

(4) *Model-Based Tools*. With the swift expansion of diverse AI communities (i.e., Huggingface, ModelScope, GitHub), various types of AI models have become readily accessible for use, serving as a pivotal tool in enhancing GR systems. These AI models encompass a wide array of tasks, each accompanied by comprehensive model descriptions and usage examples.

HuggingGPT [248] employs ChatGPT as a controller to deconstruct user queries into tasks, determining which models to invoke for execution. Similarly, Visual ChatGPT [315] integrates a visual foundation model with LLMs, leveraging ChatGPT as a prompt manager to mobilize visual foundation models like BLIP [144] and ControlNet [347], adept at processing image-based requests efficiently compared to multi-modal models.

4.3 Generating Response with Citation

To build a reliable GenIR system, generating responses with citations is a promising approach [87, 167, 194]. Citations allow users to clearly understand the source of each piece of knowledge

in the response, enhancing trust and facilitating widespread adoption. Existing methods can be divided into directly generating responses with citations and using a retrieval module to enhance the generated content. Refer to the green portion in Figure 6 for an overview of this section.

4.3.1 Direct Generating Response with Citation. This method uses the model's intrinsic memory to generate source citations without relying on a retrieval module.

(1) *Model Intrinsic Knowledge.* Leveraging the capabilities of the language model itself, according-to prompting [312] guides LLMs to more accurately cite information from pre-training data by adding phrases like "according to Wikipedia" in the prompts.

To improve citation quality, **Iterative Feedback Learning (IFL)** [128] employs a critique model to assess and provide feedback on generated text, iteratively enhancing LLMs' citation accuracy, content correctness, and fluency. Additionally, Fierro et al. [62] introduce a plan-based approach using a series of questions as a blueprint for content generation, with abstract and extractive attribution models, showing that planning improves citation quality.

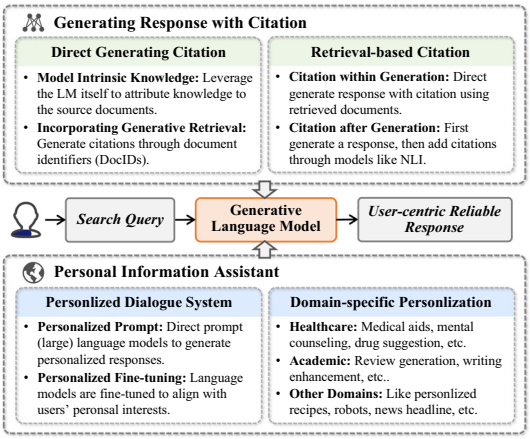

Fig. 6. Generating response with citation and personal information assistant are also crucial approaches for building a reliable and user-centric GenIR system.

(2) *Incorporating GR.* As envisioned by Metzler et al. [194], allowing the model to directly generate responses with citations is a promising approach for building an expert-level reliable IR system. Users receive reliable responses tailored to their needs without searching through returned documents. Moreover, the cited document is generated by the model through the GR approach described in Section 3, directly producing corresponding DocIDs.

Utilizing GR, 1-PAGER [96] combines answer generation and evidence retrieval by generating N-gram DocIDs through constrained decoding using FM-Index [61], enabling step-by-step corpus partitioning, document selection, and response generation. This method matches retrieval-then-read methods in accuracy and surpasses closed-book QA models by attributing predictions to specific evidence, offering a new scheme for integrating retrieval into seq2seq generation.

Recently, Khalifa et al. [121] propose a source-aware training method where models learn to associate DocIDs with knowledge during pre-training and provide supporting citations during instruction tuning, effectively achieving knowledge attribution and enhancing LLM verifiability.

4.3.2 Retrieval-Based Response with Citation. To enhance the accuracy of citations, several methods have been developed based on retrieval techniques to fetch relevant documents, thereby improving the quality of responses with embedded citations.

(1) *Citation within Generation.* Following retrieval, models directly generate responses that include citations. Initially, systems like WebGPT [203], LaMDA [280], and WebBrain [221] utilized web pages or Wikipedia to construct large-scale pre-training datasets, teaching models how to generate responses with citations.

Subsequently, more advanced strategies for citation generation were proposed. For instance, **Search-in-the-Chain (SearChain)** [324] first generates a reasoning chain (**Chain-of-Query [CoQ]**) via LLM prompts, then interacts with each CoQ node using retrieval for verification and

completion, ultimately generating the reasoning process and marking citations at each inference step.

LLatrieval [155] suggests continuously improving retrieval results through iterative updating, verifying whether retrieved documents support the generated answers until satisfaction. AGREE [333] uses a NLI model to verify consistency between LLM-generated answers and retrieved documents, employing a Test-Time Adaptation strategy that allows LLMs to actively search and cite current information during generation, enhancing response accuracy and reliability. VTG [259] integrates an evolved memory system and a dual-layer validator for generating verifiable text, combining long-term and short-term memories to adapt to dynamic content, and uses an NLI model to evaluate logical support between claims and evidence.

Based on the **Graph of Thoughts (GoT)**, HGOT [59] improves context learning in retrieval-augmented settings by constructing a hierarchical GoT, leveraging the LLM's planning capabilities to break down complex queries into smaller sub-queries and introducing a scoring mechanism to assess the quality of retrieved paragraphs.

Employing reinforcement learning, Huang et al. [86] introduce a fine-grained reward mechanism to train language models, allocating specific rewards for each generated sentence and citation to teach models accurate external source citation. This approach uses rejection sampling and reinforcement learning algorithms to enhance citation-inclusive text generation through localized reward signals. APO [141] reimagines attributive text generation as a preference learning problem, automatically generating preference data pairs to reduce annotation costs, and uses progressive preference optimization and experience replay to reinforce preference signals without overfitting or text degradation.

(2) *Citation after Generation.* This approach involves models first generating a response, then adding citations through models like NLI. RARR [69] improves attributability by automatically finding external evidence for the language model's output and post-editing to correct content while preserving the original output, enhancing attribution capabilities without altering the existing model. PURR [23] employs an unsupervised learning method where LLMs generate text noise themselves, then trains an editor to eliminate this noise, improving attribution performance and significantly speeding up generation. CEG [149] searches for supporting documents related to generated content and uses an NLI-based citation generation module to ensure each statement is supported by citations. "Attribute First, then Generate" [254] decomposes the generation process, first selecting relevant source text details and then generating based on these details, achieving localized attributability with each sentence supported by a clear source, greatly reducing manual fact-checking workload.

4.4 Personal Information Assistant

The core of the GenIR system is the user, so understanding user intent is crucial. Researchers have explored various methods like personalized search [300, 362, 371], dialogue [171, 183, 352], and recommender [45, 168, 359] systems to explore users' interests. Specifically, personalized information assistants aim to better understand users' personalities and preferences, generating personalized responses to better meet their information needs. This section reviews the progress in research on personalized dialogue and domain-specific personalization. An overview of this section is provided in the blue area of Figure 6.

4.4.1 Personalized Dialogue System. To better understand user needs, researchers have explored two main approaches: personalized prompt design and model fine-tuning.

(1) *Personalized Prompt.* For personalized prompt design, Liu et al. [168] and Dai et al. [45] input users' interaction and rating history into ChatGPT [208] for in-context learning, effectively

generating personalized responses. LaMP [238] enhances the language model's personalized output by retrieving personalized history from user profiles. Using long-term history, Christakopoulou et al. [34] design prompts describing users' long-term interests, needs, and goals for input into LLMs. BookGPT [359] uses LLM prompts, interactive querying methods, and result verification frameworks to obtain personalized book recommendations. PerSE [292] infers preferences from several reviews by a specific reviewer and provides personalized evaluations for new story inputs.

Using prompt rewriting, Li et al. [139] propose a method combining supervised and reinforcement learning to better generate responses from frozen LLMs. Similarly, Chen et al. [30] rewrites user input prompts using extensive user text-to-image interaction history to align better with expected visual outputs.

(2) *Personalized Fine-Tuning*. This line of work focuses on fine-tuning models for personalized response generation. Zhang et al. [352] introduced the Persona-Chat dataset with 5 million personas to train models for more personalized dialogues. Mazaré et al. [189] created a dataset of over 700 million conversations extracted from Reddit, demonstrating the effectiveness of training dialogue models on large-scale personal profiles. $\mathcal{P}^2$Bot [171] generates personalized and consistent dialogues by simulating the perception of personalities between conversation participants. DHAP [183] designs a novel Transformer structure to automatically learn implicit user profiles from dialogue history without explicit personal information. Wu et al. [320] propose a generative segmentation memory network to integrate diverse personal information. Fu et al. [66] developed a variational approach to model the relationship between personal memory and knowledge selection, with a bidirectional learning mechanism.

Using reinforcement learning, Cheng et al. [31] collected a domain-specific preference dataset and proposed a three-stage reward model learning scheme, including base model training, general preference fine-tuning, and customized preference fine-tuning. Jang et al. [97] developed "Personalized Soups," first optimizing multiple policy models with different preferences using PPO [244], then dynamically combining parameters during inference.

Using retrieval-enhanced methods, LAPDOG [90] retrieves relevant information from story documents to enhance personal profiles and generate better personalized responses. SAFARI [293] leverages LLMs' planning and knowledge integration to generate responses consistent with character settings. Inspired by writing education, Li et al. [140] proposed a multi-stage, multi-task framework including retrieval, ranking, summarization, synthesis, and generation to teach LLMs personalized responses. For subjective tasks, [314] studied the superior performance of personalized fine-tuning in subjective text perception tasks compared to non-personalized models.

To achieve a personalized information assistant for every user, OPPU [272] uses personalized PEFT [52] to store user-specific behavioral patterns and preferences, showing superior performance. For multi-modal scenarios, PMG [247] proposes a personalized multi-modal generation method that transforms user behavior into natural language, allowing LLMs to understand and extract user preferences.

4.4.2 Domain-Specific Personalization. Understanding users' personalized information needs, the GenIR system has broad applications across various domains such as healthcare, academia, education, and recipes.

(1) *Healthcare.* In AI-assisted healthcare, personalization plays a crucial role. Liu et al. [174] utilize few-shot tuning to process time-series physiological and behavioral data. Zhang et al. [345] implement medical diagnosis identification and diagnostic assistance using prompts from ChatGPT [208] and GPT-4 [2]. Yang et al. [11] propose an LLM for traditional Chinese medicine called Zhongjing, based on LLaMA [283], undergoing pre-training, supervised fine-tuning, and RLHF [35]. Abbasian et al. [1] introduce an open source LLM-based conversational health agent framework called

openCHA, which collects necessary information through specific actions and generates personalized responses. MedAgents [275] propose a multidisciplinary collaboration framework where LLM-based agents engage in multi-round cooperative discussions to enhance expertise and reasoning.

For mental healthcare, Mental-LLM [325] presents a framework using LLMs to predict mental health from social media text data, with prompting-based and fine-tuning methods for real-time monitoring of issues like depression and anxiety. Lai et al. [126] introduce Psy-LLM, a psychological consultation aid combining pre-trained LLMs with real psychologist Q&As and psychological articles.

For medication suggestions, Liu et al. [176] propose PharmacyGPT, a framework for generating personalized patient groups, formulating medication plans, and predicting outcomes.

(2) *Academic*. In the academic domain, RevGAN [148] can automatically generate controllable and personalized user reviews based on users' emotional tendencies and stylistic information. For writing assistants, Porsdam et al. [217] explore personalized enhancement of academic writing using LLMs like GPT-3 [15], showing higher quality after training with authors' published works. Similarly, to address the lack of personalized outputs in LLMs, Mysore et al. [201] propose Pearl, a personalized LLM writing assistant trained on users' historical documents and develop a KL divergence training objective for retrievers.

(3) *Education*. Cui et al. [43] propose an adaptive and personalized exercise generation method that adjusts difficulty to match students' progress by combining knowledge tracing and controlled text generation. EduChat [47] learns education-specific functionalities through pre-training on educational corpora and fine-tuning on customized instructions, addressing delayed knowledge updates and lack of expertise in LLMs.

(4) *Other Domains*. For recipe generation tasks, Majumder et al. [185] propose a personalized generation model based on users' historical recipe consumption, enhancing personalization. For personalized headline generation, Zhang et al. [346] simulate users' interests based on browsing history to generate news headlines. Salemi et al. [238] propose the LaMP benchmark, including personalized generation tasks like news headline, academic title, email subject, and tweet rewriting. Additionally, for personalized assistance with home cleaning robots, TidyBot [316] uses LLMs to generalize from user examples to infer user preference rules.

5 Evaluation

This section will provide a range of evaluation metrics and benchmarks for GenIR methods, along with analysis and discussions on their performance.

5.1 Evaluation for GR

5.1.1 Metrics. In this section, we discuss several core metrics for evaluating GR methods. These metrics provide different perspectives on the effectiveness of a GR system, including its accuracy, efficiency, and the relevance of its results. Specifically, we consider Recall, R-Precision, **Mean Reciprocal Rank (MRR)**, **Mean Average Precision (MAP)**, and **Normalized Discounted Cumulative Gain (nDCG)**. Each metric captures unique aspects of retrieval performance, allowing for a comprehensive assessment of the system's capabilities.

- *Recall* measures the proportion of relevant documents retrieved by the search system, reflecting its ability to find all relevant items.
- *R-Precision* evaluates the precision at a rank position corresponding to the number of relevant documents, balancing precision and recall at a specific cutoff.
- *MRR* captures the average rank position of the first relevant document, emphasizing the system's ability to return relevant results early in the ranking.

— *MAP* calculates the average precision across multiple queries, considering the exact positions of all relevant documents and providing a comprehensive measure of retrieval accuracy.
— *nDCG* takes into account not only the relevance of the documents returned but also their positions in the result list, reflecting both the quality and the ordering of the results.

For detailed mathematical formulations of these metrics, please refer to Appendix A.1.

5.1.2 Benchmarks. Evaluating the effectiveness of GR methods relies on high-quality and challenging benchmark datasets.

MS MARCO [204] is a large-scale dataset designed for machine reading comprehension, retrieval, and QA tasks in web search environments. It contains millions of documents and passages derived from real user queries, providing a realistic benchmark for assessing GR systems in complex search scenarios.

Natural Questions (NQ) [125] is a QA dataset introduced by Google, utilizing Wikipedia as its primary corpus. It includes a vast number of natural user queries and their corresponding answers, making it suitable for evaluating the retrieval performance of GR systems in addressing real-world informational needs.

KILT [216] is a comprehensive benchmark integrating multiple categories of knowledge-intensive tasks such as fact checking, entity linking, slot filling, open-domain QA, and dialogue. Utilizing Wikipedia as its corpus, KILT aims to evaluate the effectiveness of IR systems in handling complex language tasks that require extensive background knowledge.

TREC Deep Learning Track 2019 and 2020 [40, 41] focus on leveraging deep learning techniques to enhance IR efficiency, primarily through document and passage ranking tasks. These tracks utilize the MS MARCO dataset to emulate real-world search queries, providing a standardized environment for benchmarking various retrieval methodologies.

DynamicIR. For dynamic corpora, DynamicIR [335] proposes a task framework based on StreamingQA [166] benchmark for evaluating IR models within dynamically updated corpora. Through experimental analysis, DynamicIR revealed that GR systems are superior in adapting to evolving knowledge, handling temporally informed data, and are more efficient in terms of memory, indexing time, and FLOPs compared to dense retrieval systems.

ExcluIR. For exclusionary retrieval tasks, where users explicitly indicate in their queries that they do not want certain information, ExcluIR [354] provides a set of resources. This includes an evaluation benchmark and a training set to help retrieval models understand and process exclusionary queries.

For detailed descriptions and comprehensive information about benchmark datasets, please refer to Appendix A.2.

5.1.3 Analysis. In addition to the benchmarks and metrics for evaluating the performance of GR methods, there is a series of works that have conducted detailed analyses and discussions to study the behavior of GR models.

Understanding GR. To understand the performance of DSI [279] in text retrieval, Chen et al. [28] examine uniqueness, completeness, and relevance ordering. These respectively reflect the system's ability to distinguish between different documents, retrieve all relevant documents, and accurately rank documents by relevance. Experimental analysis finds that DSI excels in remembering the mapping from pseudo queries to DocIDs, indicating a strong capability to recall specific DocIDs from particular queries. However, the study also pointed out DSI's deficiency in distinguishing relevant documents from random ones, negatively impacting its retrieval effectiveness.

Exploring the connection between generative and dense retrieval, Nguyen and Yates [205] demonstrate that they can be considered as bi-encoders in dense retrieval. Specifically, the authors

analyze the computation of dot products during the GR process, which is similar to the calculation of dot products between query vectors and document vectors in dense retrieval. Following this, Wu et al. [317] revisit GR from the perspective of multi-vector dense retrieval, revealing a common framework in computing document-query relevance between the two methods. This work also analyzes their differences in document encoding and alignment strategies, further confirming through experiments the phenomenon of term matching in the alignment matrices and their commonalities in retrieval.

Large-Scale Experimental Analysis. Later, Pradeep et al. [218] conduct the first comprehensive experimental study on GR techniques over large document sets, such as the 8.8M MS MARCO passages. It was found that among all the techniques examined, using generated pseudo queries to augment training data remains the only effective method on large document corpus. The strongest result in the experiments was achieved by using a training task that only utilized synthetic queries to Naive DocIDs, expanding the model to T5-XL (3B parameters) to achieve an MRR@10 of 26.7. Surprisingly, increasing the parameters to T5 XXL (11B) in the same setup did not improve performance but rather led to a decline. These findings suggest that more research and in-depth analysis are needed in the GR field, and possibly additional improvements to the paradigm, to fully leverage the power of larger language models.

Out-of-Distribution (OOD) *Perspective.* For OOD robustness of GR models, Liu et al. [175] investigate three aspects: query variations, new query types, and new tasks. Their study showed that all types of retrieval models suffer from performance drops with query variations, indicating sensitivity to query quality and structure. However, when dealing with new query types and tasks, GR models showed different levels of adaptability, with pre-training enhancing their flexibility. The research highlights the critical need for OOD robustness in GR models for dealing with ever-changing real-world information sources.

5.1.4 Experiments. Analyzing experimental results is essential for understanding the performance of different GR models. This section provides a comprehensive evaluation of current GR models on widely used benchmark tests and examines their applicability and limitations in scenarios such as web search, QA, and knowledge-intensive tasks. The overall results are presented in Tables 3 and 4.

Experimental Settings. Our evaluation is based on the MS MARCO [204], NQ [125], and KILT [216] benchmarks, which are commonly used datasets for existing GR methods. For the MS MARCO dataset, following previous works [263, 350, 369], we use the MS MARCO 300K subset, which contains 320k documents, 360k training instances, and 772 testing instances. For the NQ dataset, following [134, 263, 279, 305, 350], we use the NQ320K subset, which, after deduplication based on titles, contains 109k documents, 320k training instances, and 7,830 testing instances. For the KILT benchmark, we use the standard development sets. Detailed statistics are available in previous works [27, 216].

Regarding evaluation metrics, we employ Recall@{1, 10, 100} and MRR@{10, 100} for the MS MARCO and NQ datasets, and R-Precision for the KILT benchmark. In our comparisons, we include not only existing representative GR methods but also sparse retrieval methods such as BM25 [236] and SPLADEv2 [63], which are based on bag-of-words representations, and dense retrieval methods like DPR [116], GTR [206], RAG [138], and MT-DPR [184], which rely on dense embeddings.

Due to variations in datasets, corpus sizes, and evaluation metrics across different methods, alignment is necessary for a fair comparison. For the methods evaluated in our experiments, we primarily use results reported in existing papers. For methods where settings are not aligned, we provide results based on our own implementations.

Results on MS MARCO and NQ Datasets. MS MARCO and NQ are among the most widely used benchmarks for evaluating GR methods, particularly in the contexts of web search and QA.

Table 3. Overall Retrieval Performance on the MS MARCO (300K) and NQ (320K) Datasets

Model	Doc Rep.	MS MARCO					NQ				
		R@1	R@10	R@100	M@10	M@100	R@1	R@10	R@100	M@10	M@100
Sparse and Dense Retrieval											
BM25 [263]	Bag-of-words	0.196	0.591	0.861	0.313	0.325	0.297	0.603	0.821	-	0.402
SPLADEv2 [350]	Bag-of-words	0.328	0.779	0.956	0.443	0.452	0.624	0.873	0.954	0.726	0.731
DPR [263]	Dense Vector	0.271	0.764	0.948	0.424	0.433	0.502	0.777	0.909	-	0.489
GTR-Base [263]	Dense Vector	0.332	**0.793**	**0.960**	0.484	0.485	0.560	0.844	0.937	-	0.662
GR											
GENRE [350]	Title	0.266	0.579	0.751	0.361	0.368	0.591	0.756	0.814	0.653	0.656
DSI [350]	Semantic ID	0.257	0.538	0.692	0.339	0.346	0.533	0.715	0.816	0.594	0.598
DSI-QG [263, 369]	Semantic ID	0.288	0.623	-	0.385	-	0.631	0.807	0.880	-	0.695
NCI [263]	Semantic ID	0.301	0.643	0.851	0.408	-	0.659	0.852	0.924	-	0.731
SEAL [263]	Sub-string	0.259	0.686	0.879	0.393	0.402	0.570	0.800	0.914	-	0.655
Ultron [350]	Title+URL	0.304	0.676	0.794	0.432	0.437	0.654	0.854	0.911	0.726	0.729
GenRet [263]	Learnable	-	-	-	-	-	0.681	0.888	0.952	-	0.759
MINDER [350]	Multi-view	0.289	0.728	0.916	0.431	0.435	0.627	0.869	0.933	0.709	0.713
LTRGR[a]	Multi-view	0.327	0.759	0.929	0.463	0.469	0.644	0.879	0.941	0.721	0.726
GLEN [134]	Learnable	-	-	-	-	-	0.691	0.860	-	-	0.754
TSGen [350]	Term Set	**0.384**	0.781	0.931	**0.502**	**0.505**	**0.708**	0.889	0.948	**0.771**	**0.774**
NOVO [309]	Term Set	-	-	-	-	-	0.693	**0.897**	**0.959**	-	0.767
DGR[a]	Multi-view	0.359	0.779	0.934	0.498	0.504	0.682	0.887	0.949	0.759	0.764

The best results are in bold, and the second-best are underlined.
[a]Results from our own implementation, while other results are consistent with those reported in existing papers.

Table 3 presents a detailed comparison of various GR models against traditional sparse and dense retrieval methods on these datasets.

(1) Overall Performance Comparison. Overall, GR methods demonstrate competitive performance compared to sparse and dense retrieval baselines. Specifically, on the MS MARCO dataset, GR models such as TSGen and DGR achieve Recall@1 scores of 0.384 and 0.359, respectively, surpassing dense methods like DPR (0.271) and being comparable to SPLADEv2 (0.328). On the NQ dataset, GR models also show strong performance, with TSGen attaining the highest Recall@1 of 0.708, outperforming both SPLADEv2 (0.624) and DPR (0.502).

(2) Term Set DocID Methods. Analyzing models that utilize term set-based DocIDs, such as TSGen and NOVO, reveals that these methods excel in both datasets. TSGen leads with the highest Recall@1 and MRR@10 on MS MARCO and NQ, respectively, indicating robust retrieval capabilities. NOVO also performs exceptionally well on the NQ dataset, achieving the second-best Recall@1 and MRR@10, demonstrating the effectiveness of term set representations in capturing relevant document information.

(3) Multi-View DocID Methods. Multi-view approaches, exemplified by MINDER, LTRGR, and DGR, show consistent improvements over several metrics. For instance, LTRGR achieves the highest Recall@10 on MS MARCO (0.759) and maintains strong performance across other metrics and on the NQ dataset. These results suggest that leveraging multi-view DocIDs, ranking and distillation training methods enhances retrieval effectiveness by capturing diverse aspects of the documents.

(4) Learnable DocID Methods. Learnable DocID models, such as GenRet and GLEN, exhibit mixed performance. While GenRet shows competitive Recall@1 on NQ (0.681), it does not report results on MS MARCO. GLEN achieves the highest MRR@100 on NQ (0.754) but lags behind in other metrics. This indicates that learnable DocID approaches may benefit from further refinement to consistently outperform other representation methods across different datasets.

Table 4. Overall Retrieval Performance on the KILT Benchmark

Model	Doc Rep.	FC	Entity Linking			Slot Filling		Open Domain QA				Dial.
		FEVER	AY2	WnWi	WnCw	TREx	zsRE	NQ	HoPo	TQA	ELI5	WoW
Sparse and Dense Retrieval												
BM25 [184]	Bag-of-words	0.501	0.035	-	-	0.586	0.664	0.258	0.440	0.294	-	0.275
RAG [216]	Dense Vector	0.635	0.774	0.490	0.467	0.293	0.654	0.603	308	0.493	0.104	0.467
MT-DPR [184]	Dense Vector	0.747	0.838	-	-	0.692	0.772	0.615	0.442	0.620	-	0.397
GR												
BART[a]	Semantic ID	0.003	0.001	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000
BART [27]	Title	0.819	0.892	0.676	0.623	0.752	0.911	0.586	0.487	0.676	0.121	0.510
T5 [216]	Title	-	0.866	0.474	0.465	-	-	-	-	-	-	-
GENRE [17]	Title	**0.847**	**0.928**	**0.877**	**0.706**	**0.797**	0.948	0.643	0.518	0.711	**0.135**	**0.563**
SEAL[a]	Sub-string	0.826	0.866	0.809	0.651	0.704	0.919	**0.658**	**0.565**	**0.715**	0.124	0.527
CorpusBrain [27]	Title	0.821	0.908	0.723	0.662	0.776	**0.983**	0.591	0.501	0.688	0.129	0.538

The best results are in bold, and the second-best are underlined.
[a]Results from our own implementation, while other results are consistent with those reported in existing papers.

(5) Other DocID Methods. Other methods like GENRE, DSI, NCI, SEAL, and Ultron, generally underperform compared to term set and multi-view DocID methods. For example, on the MS MARCO dataset, GENRE achieves a Recall@1 of 0.266 and an MRR@10 of 0.361, which are significantly lower than TSGen (Recall@1 = 0.384, MRR@10 = 0.502) and LTRGR (Recall@1 = 0.327, MRR@10 = 0.463). The lower performance of methods utilizing simpler DocID designs (e.g., titles, semantic IDs) highlights the need for more sophisticated or alternative DocID strategies to effectively capture key information for high-quality retrieval across different scenarios.

Results on KILT Benchmark. The KILT benchmark provides a comprehensive evaluation across various knowledge-intensive tasks, utilizing a large-scale Wikipedia corpus comprising 5.9 million documents. Overall results are shown in Table 4.

(1) Overall Performance Comparison. GR methods generally outperform traditional sparse and dense retrieval approaches in most tasks. Notably, GENRE achieves the highest scores in several categories, including FEVER (0.847), AY2 (0.928), WnWi (0.877), and WnCw (0.706), outperforming the best sparse method BM25 and dense methods like MT-DPR.

(2) Title DocID Methods. Models utilizing title-based DocIDs consistently perform well on the KILT benchmark. For instance, GENRE and BART achieve FEVER scores of 0.847 and 0.821, respectively. This superior performance can be attributed to the fact that Wikipedia document titles accurately represent the key entities within each document, making the task of predicting titles relatively straightforward. Moreover, these models effectively leverage the pre-trained knowledge embedded within language models, enhancing their ability to generalize and retrieve relevant documents based on titles.

(3) Sub-String DocID Methods. Methods based on sub-string DocIDs also demonstrate strong performance on the KILT benchmark, particularly in QA tasks. SEAL achieves the highest QA scores across several categories, including NQ (0.658), HoPo (0.565), TQA (0.715), and WoW (0.527). The ability of sub-string DocID methods to capture meaningful fragments of the documents likely contributes to their high accuracy in retrieving precise information necessary for answering questions effectively.

(4) DSI-Based Numeric DocID Methods. In contrast, methods employing numeric Semantic DocIDs based on hierarchical k-means clustering [279] exhibit significantly diminished performance on the KILT benchmark. The BART model, which uses Semantic IDs and trained with just labeled queries, records scores close to zero across all tasks (e.g., FEVER: 0.003, AY2: 0.001). This decline is

primarily due to the substantial increase in corpus size, and <query, document> pairs in training data cover only a small fraction of the entire document set. Consequently, these models struggle to generalize beyond the training pairs, just "memorizing" DocIDs without capturing the broader diversity of the corpus. This observation aligns with findings from [218], which reported similar challenges of DSI [279] when scaling to an 8.8 million passage corpus in the MS MARCO benchmark.

5.2 Evaluation for Response Generation

5.2.1 Metrics. Evaluating the quality of generated responses includes aspects such as accuracy, fluency, relevance, and so on. In this section, we'll introduce the main metrics for evaluating reliable response generation, categorized into rule-based, model-based, and human evaluation metrics.

(1) *Rule-Based Metrics.* Exact Match is a straightforward evaluation method requiring the model's output to be completely identical to the reference answer at the word level. This full character-level matching is stringent, often used in tasks requiring precise and concise answers, such as QA systems, e.g., NQ [125], TriviaQA [114], SQuAD [230], and so on. It simply calculates the ratio of perfectly matched instances to the total number of instances.

For the generation of longer text sequences, BLEU [211] is a common metric initially used to evaluate the quality of machine translation. It compares the similarity between the model's output and a set of reference texts by calculating the overlap of n-grams, thereby deriving a score. This method assumes that high-quality generation should have a high lexical overlap with the labeled answer. Optimized from BLEU, METEOR [9] is an alignment-based metric that considers not only exact word matches but also synonyms and stem matches. Additionally, METEOR introduces considerations for word order and syntactic structure to better assess the fluency and consistency of the generated text.

ROUGE [161] is also a commonly used metric for evaluating longer texts, by measuring the extent of overlap in words, sentences, n-grams, and so forth, between the generated text and a collection of reference texts. It focuses on recall, meaning it evaluates how much of the information in the reference text is covered by the generated text. ROUGE comes in various forms, including ROUGE-N, which evaluates based on n-gram overlap, and ROUGE-L, which considers the longest common subsequence, catering to diverse evaluation requirements.

PPL is a metric for evaluating the performance of language models, defined as the exponentiation of the average negative log-likelihood, reflecting the model's average predictive ability for a given corpus of text sequences. The lower the PPL, the stronger the model's predictive ability. Specifically, given a sequence of words $W = w_1, w_2, \ldots, w_N$, where N is the total number of words in the sequence, PPL can be expressed as:

$$\mathrm{PPL}(W) = \exp\left\{-\frac{1}{N}\sum_{i=1}^{N}\log p(w_i|w_{<i})\right\}, \tag{11}$$

where $p(w_i|w_{<i})$ represents the pre-trained language model's probability of predicting the ith word w_i given the previous words $w_{<i}$.

(2) *Model-Based Metrics.* With the rise of pre-trained language models, a series of model-based evaluation metrics have emerged. These metrics utilize neural models to capture the deep semantic relationships between texts.

Unlike traditional rule-based metrics, BERTScore [353] utilizes the contextual embeddings of BERT [120] to capture the deep semantics of words, evaluating the similarity between candidate and reference sentences through the cosine similarity of embeddings. BERTScore employs a greedy matching strategy to optimize word-level matching and uses optional IDF weighting to emphasize important words, ultimately providing a comprehensive evaluation through a combination of recall,

precision, and F1 score. BERTScore captures not only surface lexical overlap but also a deeper understanding of the semantic content of sentences.

Similarly based on BERT [120], BLEURT [245] designed multiple pre-training tasks, enhancing the model's ability to recognize textual differences with millions of synthetic training pairs. These pre-training tasks include automatic evaluation metrics (such as BLEU [211], ROUGE [161], and BERTScore [353]), back-translation likelihood, textual entailment, and so on. Each task provides different signals to help the model learn how to evaluate the quality of text generation.

BARTScore [341], based on the pre-trained seq2seq generative model BART [137], treats the evaluation of generated text as a text generation problem. Specifically, BARTScore determines the quality of text based on the transition probability between the generated text and reference text. BARTScore does not require additional parameters or labeled data and can flexibly evaluate generated text from multiple perspectives (such as informativeness, fluency, factuality, etc.) and further enhance evaluation performance through text prompts or fine-tuning for specific tasks.

FActScore [197] focuses on the factual accuracy of each independent information point in long texts. It calculates a score representing factual accuracy by decomposing the text into atomic facts and verifying whether these facts are supported by reliable knowledge sources. This method provides a more detailed evaluation than traditional binary judgments and can be implemented efficiently and accurately through human evaluation and automated models (combining retrieval and powerful language models).

GPTScore [65] is a flexible, multi-faceted evaluation tool that allows users to evaluate text using natural language instructions without the need for complex training processes or costly annotations. GPTScore constructs an evaluation protocol dynamically through task specification and aspect definition and utilizes the zero-shot capability of pre-trained language models to evaluate text quality, optionally using demonstration samples to improve evaluation accuracy.

(3) *Human Evaluation Metrics.* Human evaluation is an important method for assessing the performance of language models, especially in complex tasks where automated evaluation tools struggle to provide accurate assessments. Compared to rule-based and model-based metrics, human evaluation is more accurate and reliable in real-world applications. This evaluation method requires human evaluators (such as experts, researchers, or everyday users) to provide comprehensive assessments of the model-generated content based on their intuition and knowledge.

Human evaluation measures the quality of language model outputs by integrating multiple assessment criteria, following [22]: Accuracy [253] primarily evaluates the correctness of information and its correspondence with facts; Relevance [360] focuses on whether the model's output is pertinent to the specific context and user query; Fluency [287] examines whether the text is coherent, natural, and facilitates smooth communication with users; Safety [102] scrutinizes whether the content may lead to potential adverse consequences or harm. These indicators collectively provide a comprehensive assessment of the model's performance in real-world settings, ensuring its effectiveness and applicability.

However, human evaluation also faces numerous challenges, primarily including high costs and time consumption, difficulty in controlling evaluation quality, inconsistency in evaluation dimensions, issues of consistency due to evaluators' subjectivity, and the need for professional evaluators for specific tasks. These problems limit the widespread application of human evaluation and the comparability of results [21].

5.2.2 Benchmarks and Analysis. In this section, we explore various benchmarks for evaluating the performance of language models in generating reliable responses. These benchmarks assess language understanding, factual accuracy, reliability, and the ability to provide timely information.

(1) *General Evaluation.* To comprehensively assess the language models' understanding capabilities across a wide range of scenarios, MMLU [83] utilizes a multiple-choice format covering 57 different tasks, from basic mathematics to American history, computer science, and law. This benchmark spans evaluations in humanities, social science, and science, technology, engineering, and mathematics, providing a comprehensive and challenging test. It has been widely used in the evaluation of LLMs in recent years [104, 283, 284].

Furthermore, BIG-bench [257] introduces a large-scale and diverse benchmark designed to measure and understand the capabilities and limitations of LLMs across a broad range of tasks. Including 204 tasks contributed by 450 authors from 132 institutions, it covers areas such as linguistics, mathematics, and common sense reasoning. It focuses on tasks beyond the capabilities of language models, exploring how model performance and societal biases evolve with scale and complexity.

LLM-Eval [165] offers a unified multi-dimensional automatic evaluation method for open-domain dialogue of LLMs, eliminating the need for manual annotation. The performance of LLM-Eval across various datasets demonstrates its effectiveness, efficiency, and adaptability, improving over existing evaluation methods. The research also analyzes the impact of different LLMs and decoding strategies on the evaluation outcomes, underscoring the importance of selecting suitable LLMs and decoding strategies.

For Chinese, C-Eval [91] aims to comprehensively evaluate LLMs' advanced knowledge and reasoning capabilities in the Chinese context. It is based on a multiple-choice format, covering four difficulty levels and 52 different academic fields from secondary school to professional levels. C-Eval also introduces C-Eval Hard, a subset containing highly challenging subjects to test the models' advanced reasoning capabilities. Through evaluating state-of-the-art English and Chinese LLMs, C-Eval reveals areas where current models still fall short in handling complex tasks, guiding the development and optimization of Chinese LLMs.

(2) *Tool Evaluation.* To assess the ability of language models to utilize tools, API-Bank [147] provides a comprehensive evaluation framework containing 73 APIs and 314 tool usage dialogs, along with a rich training dataset of 1,888 dialogs covering 1,000 domains to improve LLMs' tool usage capabilities. Experiments show that different LLMs perform variably in tool usage, highlighting their strengths and areas for improvement.

Later, ToolBench [224] developed a comprehensive framework including a dataset and evaluation tools to facilitate and assess the ability of LLMs to use over 16,000 real-world APIs. It enhances reasoning capabilities by automatically generating diverse instruction and API usage scenario paths, introducing a decision tree based on depth-first search. ToolBench significantly enhances LLMs' performance in executing complex instructions and in their ability to generalize to unseen APIs. ToolLLaMA, an LLM fine-tuned from LLaMA [283], exhibits remarkable zero-shot capabilities and performance comparable to state-of-the-art LLMs like ChatGPT [208].

(3) *Factuality Evaluation.* TruthfulQA [163] measures the truthfulness of language models in answering questions. This benchmark consists of 817 questions covering 38 categories, including health, law, finance, and politics. This evaluation reveals that, even in optimal conditions, the truthfulness of model responses only reaches 58%, in stark contrast to human performance at 94%. Moreover, they proposed an automated evaluation metric named GPT-judge, which classifies the truthfulness of answers by fine-tuning the GPT-3 [16] model, achieving 90-96% accuracy in predicting human evaluations.

HaluEval [143] is a benchmark for evaluating LLM illusions, constructed using a dataset containing 35K illusion samples, employing a combination of automated generation and manual annotation. This provides effective tools and methods for assessing and enhancing LLMs' capabilities in identifying and reducing illusions. For Chinese scenarios, HalluQA [32] designs 450

meticulously selected adversarial questions to assess the illusion phenomenon in Chinese LLMs, covering multiple domains and reflecting Chinese culture and history, identifying two main types of illusions: imitative falsehoods and factual errors.

To evaluate the ability of LLMs to generate answers with cited text, ALCE [70] builds an end-to-end system for retrieving relevant text passages and generating answers with citations. ALCE contains three datasets, covering different types of questions, and evaluates the generated text's quality from "fluency," "correctness," and "citation quality" dimensions, combining human evaluation to verify the effectiveness of the evaluation metrics. The experimental results show that while LLMs excel at generating fluent text, there is significant room for improvement in ensuring content factual correctness and citation quality, especially on the ELI5 dataset where the best model was incomplete in citation support half of the time.

(4) *Real-Time Evaluation.* RealTime QA [117] created a dynamic question-and-answer platform that regularly releases questions and evaluates systems weekly to ask and answer questions about the latest events or information. It challenges the static assumption of traditional QA datasets aiming for immediate application. Experiments based on LLMs like GPT-3 and T5 found that models could effectively update their generated results based on newly retrieved documents. However, when the retrieved documents failed to provide sufficient information, models tended to return outdated answers.

Furthermore, FreshQA [289] evaluates LLMs' performance in challenges involving time-sensitive and erroneous premise questions by creating a new benchmark containing questions of this nature. Evaluating various open and closed-source LLMs revealed significant limitations in handling questions involving rapidly changing knowledge and erroneous premises. Based on these findings, the study proposed a simple in-context learning method, FreshPrompt, significantly improving LLMs' performance on FreshQA by integrating relevant and up-to-date information sourced from search engines into the prompt.

(5) *Safety, Ethic, and Trustworthiness.* To comprehensively evaluate the safety of LLMs, SafetyBench [356] implements an efficient and accurate evaluation of LLMs' safety through 11,435 multiple-choice questions covering 7 safety categories in multiple languages (Chinese and English). The diversity of question types and the broad data sources ensure rigorous testing of LLMs in various safety-related scenarios. Comparing the performance of 25 popular LLMs, SafetyBench revealed GPT-4's significant advantage and pointed out the areas where current models need improvements in safety to promote the rapid development of safer LLMs.

For ethics, TrustGPT [92] aims to assess LLMs' ethical performance from toxicity, bias, and value alignment, three key dimensions. The benchmark uses pre-defined prompt templates based on social norms to guide LLMs in generating content and employs multiple metrics to quantitatively assess the toxicity, bias, and value consistency of these contents. Experimental analysis revealed that even the most advanced LLMs still have significant issues and potential risks in these ethical considerations.

For trustworthiness, TrustLLM [261] explores principles and benchmarks including truthfulness, safety, fairness, robustness, privacy, and machine ethics across six dimensions. Extensive experiments, including assessing 16 mainstream LLMs' performance on 30 datasets, found that trustworthiness usually positively correlates with functional effectiveness. While proprietary models typically outperform open-source models in trustworthiness, some open-source models like Llama2 showed comparable high performance.

These benchmarks provide important tools and metrics for evaluating and improving the capabilities of language models, contributing to the development of more accurate, reliable, safe, and timely GenIR systems. For further understanding of the evaluation works, [22, 46, 89, 291] offer more detailed introductions.

6 Challenges and Prospects

This section discusses the key challenges faced in the fields of GR and reliable response generation, as well as potential directions for future research.

6.1 Challenges on GR

6.1.1 Scalability Issues. As extensively studied by [218], GR demonstrates significantly lower retrieval accuracy compared to dense retrieval when handling million-level document corpora in web search scenarios. Merely increasing the model size does not yield stable performance improvements. However, GR outperforms dense retrieval in document collections smaller than 300K, posing a question: What impedes GR methods from scaling to large document sizes? This issue encompasses several aspects.

Training Data. Current LLMs are pre-trained on huge datasets ranging from hundreds of billions to several trillion tokens, covering vast knowledge sources such as the internet, books, and news articles, consuming substantial computational power [357]. They are then extensively fine-tuned with high-quality, human-annotated data to achieve substantial generalization capabilities [137, 209, 229, 283]. In contrast, GR models often begin with a pre-trained language model and are fine-tuned on labeled data comprising <query, DocID> pairs, which does not sufficiently prepare them to fully grasp GR tasks. For numeric-based DocIDs, the models, having not encountered these numbers in their pre-training phase, tend to rote memorize the DocIDs seen during training, struggling to predict unseen ones effectively. Similarly, if text-based DocIDs fail to precisely represent the documents, the model also tends to rote learning.

A potential solution is to create a large-scale pre-training dataset for GR on a general corpus, possibly including a variety of common DocIDs such as URLs, titles, and numerical sequences. We can utilize instructions to distinguish generation targets for various DocIDs. Then we can pre-train a GR model from scratch, the model can understand GR across diverse domains. This method could bridge the gap between language model pre-training data and GR tasks, enhancing the generalization ability of GR models across different corpora.

Training Method. As described in Section 3.1.1, existing training methods explore various training objectives, including seq2seq training, learning DocID, and ranking capabilities. Other methods involve knowledge distillation [29], reinforcement learning [363], and so on. Is there a better training method to enable GR models to master generating DocID ranking lists? For example, RLHF [35] has been effectively used to train LLMs [209, 284], though at a high cost. Exploring RLHF in the GR field is also worthwhile.

Model Structure. As discussed in Section 3.1.2, most current GR models are based on encoder–decoder Transformers structures [279, 305, 369], such as T5 [229] and BART [137]. Some GR methods like CorpusLM [151] have experimented with a decoder-only structure of the LLM Llama2 [284], requiring more training computational power but not significantly improving performance. Research is needed to determine which structure is more suitable for GR. Additionally, whether increasing model and data size could lead to emergent phenomena similar to those observed in LLMs [242, 310] is also a promising research direction.

6.1.2 Handling Dynamic Corpora. Real-world applications often involve dynamically changing corpora, such as the web and news archives, where incremental learning is essential. However, for language models, indexing new documents inevitably leads to forgetting old ones, posing a challenge for GR systems. Existing methods like DSI++ [191], IncDSI [123], CLEVER [24], and CorpusBrain++ [79] propose solutions such as experience replay, constrained optimization, IPQ, and continual generative pre-training frameworks to address incremental learning issues. Yet, these

methods have their specific applicable scenarios, and more effective and universally applicable incremental learning strategies remain a key area for exploration.

6.1.3 DocID. Accurately representing a document with high-quality DocIDs is crucial for GR. For example, the KILT dataset based on the Wikipedia corpus, which includes 5.9 million documents, demonstrates optimistic retrieval performance for GR methods using titles as DocIDs [17, 27, 151]. This is because each document in Wikipedia has a unique manually annotated title that represents the core entity discussed in that page. However, in the web search scenario, such as in the MS MARCO dataset [204], many documents lack a unique title, are overlapping, and the titles do not accurately represent the core content of the documents. Thus, GR performance significantly declines in the MS MARCO corpus of 8.8 million passages.

Therefore, how to construct high-quality titles (or other types of DocIDs) in general corpora, similar to those in Wikipedia, that not only accurately represent documents but also are lightweight, is a critical factor for implementing GR methods and warrants in-depth research.

Text or Numeric? As discussed in Section 3.2, current methods include text-based and numeric-based DocIDs, each with their advantages and disadvantages. Text-based DocIDs effectively leverage the linguistic capabilities of pre-trained generative language models and offer better interpretability. Numeric-based DocIDs can utilize dense retriever embeddings to obtain semantic DocID sequences; they can also complement dense retrievers to achieve synergistic benefits.

However, to ensure good generalization ability of GR models without extensive pre-training, it is essential to utilize the inherent pre-trained parameters of the model. Coherent textual DocIDs can naturally leverage this aspect, but they also need to capture key document semantics and maintain linguistic sequence characteristics. Numeric DocIDs, however, do not offer this advantage. Thus, as mentioned in Section 6.1.1, extensive pre-training is necessary to enable models to fully understand the meanings behind these numerical strings, which is a costly endeavor.

Do We Need a Unique ID for Each Document? Most current GR methods use a unique DocID to uniquely identify a document. However, as the number of documents in a corpus increases, maintaining a unique DocID becomes increasingly challenging. Even if a unique DocID is maintained, it is difficult to differentiate significantly from other DocIDs semantically, leading to reduced retrieval precision. Some methods, such as using sub-string as DocIDs [12, 25], have proven effective. These methods utilize the FM-Index [61] to ensure the generated sub-string exists in the corpus and use the number of generated sub-strings in different documents to rank documents, demonstrating good performance and generalization ability.

However, since this method is based on FM-Index, its inference latency is high, which is an issue that needs addressing. Furthermore, exploring other more efficient alternatives to FM-Index and even considering not using constrained search but freely generating a DocID sequence followed by a lightweight matching and scoring module to efficiently return a document ranking list are also worthy of exploration.

6.1.4 Efficiency Concerns. Current GR methods generally rely on constrained beam search to generate multiple DocID sequences during inference, resulting in high latency. This is particularly severe when returning 100 or more documents, with latencies reaching several hundred milliseconds [305], which is unacceptable for low-latency IR systems. Therefore, designing more efficient inference methods is crucial. To reduce inference latency, the length of the DocID sequence should not be too long; 16 tokens or fewer is an efficient range. This necessitates designing DocIDs that are precise and concise enough to represent documents while maintaining performance and improving efficiency. Additionally, developing more efficient decoding strategies is a valuable research direction for the future.

6.1.5 Multi-Modal GR. Existing multi-modal GR models aim to retrieve images by converting each image in the collection into a unique sequence that serves as its identifier. A language model is then employed to predict these image identifiers, enabling effective image retrieval. However, there are still potential areas for future optimization: (1) *Image Representation*: Developing advanced image representation techniques is essential for enhancing the performance of multi-modal GR. These techniques should capture the key features of an image within its identifier sequence. (2) *End-to-end Training*: Existing methods perform image representation and image identifier prediction separately for GR. Exploring how to train these two tasks in a fully end-to-end manner is also worth investigating. (3) *Extend to Additional Modalities*: Current multi-modal GR methods predominantly focus on text and image modalities. Expanding these approaches to incorporate additional modalities such as audio and video presents a valuable research opportunity.

6.2 Challenges on Reliable Response Generation

6.2.1 Improving Accuracy and Factuality. In GenIR systems, ensuring content accuracy and factuality is crucial. To achieve this, as mentioned in Section 4, there are two main areas of improvement.

Internal Knowledge Memorization. Firstly, training stronger generative models is critical for building reliable GenIR systems. Various commercial LLMs continue to progress, utilizing vast training data and computational resources, but exploring better model structures is also worthwhile. Recent research such as Retentive Networks [264], Mamba [77], and others have shown potential to challenge the performance and efficiency of Transformers [288]. However, whether these can scale and truly surpass Transformer-based LLMs in generation quality is still an open question. Moreover, what types of training data and methods can consistently produce models capable of generating high-quality, reliable text also deserve thorough investigation and summary. The mechanisms by which language models recall knowledge during inference are not yet clear and need to be fully understood to better serve user information needs.

External Knowledge Enhancement. As described in Section 4.2.1, RAG is an effective method widely applied in LLMs. However, there is still room for improvement. (1) For example, whether inserting retrieved documents directly into generative models via prompts is the best method, or if there are better ways, such as inputting embeddings [334], needs exploration. (2) Additionally, whether models can autonomously decide whether to perform retrieval [270, 294], and when in the generation process to perform it [110]. (3) Third, in dialogue scenarios, enhancing RAG models to better utilize long conversational history is also worth further exploration [199].

Tool-augmented generation, as discussed in Section 4.2.2, is also a popular method for endowing LLMs with fine-grained world knowledge and performing complex tasks. Recent research has raised questions, such as "Should tools always be used?" [308]. More specifically, whether the performance improvements brought by using tools justify the extra computational costs incurred during model training or the inference costs during testing. Existing work mainly focuses on task accuracy, but studying the cost-effectiveness of these methods is also a valuable topic.

6.2.2 Real-Time Properties of GenIR Systems. Timeliness is critical for GenIR systems, as well as traditional IR systems, to provide users with the most up-to-date information. However, since the knowledge of pre-trained generative models is fixed after training, methods like retrieval and tool augmentation are needed to acquire new external knowledge. Research on real-time knowledge acquisition remains limited, making it a valuable area for investigation.

Moreover, continually relying on outdated knowledge from language models is inadequate, as models cannot comprehend the significance of given contexts or backgrounds in the current era, thus reducing the reliability of the generated content. Therefore, updating the information in

language models while avoiding the forgetting of existing knowledge, such as through continual learning [299, 318], knowledge editing [190, 301, 332], and so on, is a topic worth further exploring.

6.2.3 Bias and Fairness. Since LLMs are often trained on large, unfiltered datasets, GenIR systems may propagate stereotypes and biases present in the data regarding race, culture, and other aspects [67]. Researchers have explored various methods to enhance the fairness of generated content during training data selection, training methods, generation techniques, and rewriting phases. However, biases have not been eradicated and require a thorough understanding of the mechanisms by which generative models produce biases, to design methods to solve them and build fair GenIR systems that further the practical application of GenIR.

6.2.4 Privacy and Security. Firstly, the content generated by GenIR systems risks plagiarism [49, 119]. Studies such as [20, 88] indicate that pre-trained language models can reproduce large segments of their training data, leading to inadvertent plagiarism and causing academic dishonesty or copyright issues. On one hand, legal regulations regarding the copyright of AIGC will gradually emerge and evolve. On the other hand, technical research aimed at reducing plagiarism by generative models, such as generating text with correct citations [87, 170, 194], is a promising research direction for reliable GenIR that has received increasing attention in recent years.

Moreover, due to the unclear mechanisms of memory and generation in pre-trained language models, GenIR systems inevitably return unsafe content. For example, studies [19, 20, 364] show that when attacked, LLMs may return private information of users seen in training data. Therefore, understanding the mechanisms by which LLMs recall training data and designing effective defense mechanisms to enhance security are crucial for the widespread use of GenIR systems. Additionally, developing effective detection methods for content generated by LLMs is essential for enhancing the security of GenIR systems [329].

6.3 Unified Framework

This article discusses two mainstream forms of GenIR: GR and reliable response generation. However, each approach has its advantages and limitations. GR still returns a list of documents, whereas the reliable response generation model itself cannot effectively capture document-level relationships. Therefore, integrating these two approaches is a promising research direction.

6.3.1 Unified Framework for Retrieval and Generation. Given that both GR and downstream generation tasks can be based on generative language models, could a single model perform both retrieval and generation tasks? Indeed, it could.

Current attempts, such as UniGen [154], use a shared encoder and two decoders for GR and QA tasks, respectively, and show superior performance on small-scale retrieval and QA datasets. However, they struggle to generalize across multiple downstream tasks and to integrate with powerful LLMs. Additionally, CorpusLM [151] uses a multi-task training approach to obtain a universal model for GR, QA, and RAG. Yet, merely merging training data does not significantly improve retrieval and generation performance, and CorpusLM remains limited to the Wikipedia corpus. Facing a broader internet corpus presents significant challenges.

In the future, can we construct a **Large Search Model (LSM)** that allows an LLM to have the capability to generate DocIDs and reliable responses autonomously? Even LSM could decide when to generate DocIDs to access the required knowledge before continuing generation. Unlike the LSM defined in [298], which unifies models beyond the first-stage retrieval (such as re-ranking, snippet, and answer models), we aim to integrate the first-stage retrieval as well, enabling the LSM to fully understand the meaning of retrieval and its connection with various downstream generation tasks.

6.3.2 Towards End-to-End Framework for Various IR Tasks. Metzler et al. [194] envisioned an expert-level corpus model that not only possesses linguistic capabilities but also understands document-level DocIDs and knows the sources of its own knowledge. Such a model could not only solve the issue of hallucinations common in traditional language models but could also generate texts with references pointing to the source documents, thus achieving a reliable end-to-end GenIR model. By understanding DocIDs and knowledge sources, this end-to-end system could also perform additional IR tasks, such as returning the main content of a document given its DocID or returning other related document DocIDs, as well as enabling multi-lingual and multi-modal retrieval.

Current methods, as discussed in this GenIR survey, primarily focus on GR and response generation as separate entities. GR models excel at comprehending DocIDs at the document-level, while downstream models demonstrate powerful task generation capabilities. However, existing methods face challenges when it comes to effectively integrating these two generative abilities, limiting the overall performance and effectiveness of the GenIR system. The integration of these generative abilities in a seamless and efficient manner remains a key challenge in the field.

In the future, we can design training methods that align knowledge and DocIDs and construct high-quality training datasets for generating answers with references, to train such an end-to-end GenIR model. Achieving this goal remains challenging and requires the collaborative efforts of researchers to contribute to building the next generation of GenIR systems.

7 Conclusion

In this survey, we explore the latest research developments, evaluations, current challenges, and future directions in GenIR. We discuss two main directions in the GenIR field: GR and reliable response generation. Specifically, we systematically review the progress of GR covering model training, DocID design, incremental learning, adaptability to downstream tasks, multi-modal GR, and generative recommendation systems; as well as advancements in reliable response generation in terms of internal knowledge memorization, external knowledge enhancement, generating responses with citations, and personal information assistance. Additionally, we have sorted out the existing evaluation methods and benchmarks for GR and response generation. We organize the current limitations and future directions of GR systems, addressing scalability, handling dynamic corpora, document representation, and efficiency challenges. Furthermore, we identify challenges in reliable response generation, such as accuracy, real-time capabilities, bias and fairness, privacy, and security. We propose potential solutions and future research directions to tackle these challenges. Finally, we also envision a unified framework, including unified retrieval and generation tasks, and even building an end-to-end framework capable of handling various IR tasks. Through this review, we hope to provide a comprehensive reference for researchers in the GenIR field to further promote the development of this area.

References

[1] Mahyar Abbasian, Iman Azimi, Amir M. Rahmani, and Ramesh C. Jain. 2023. Conversational health agents: A personalized LLM-powered agent framework. arXiv:2310.02374. Retrieved from https://arxiv.org/abs/2310.02374

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. arXiv:2303.08774. Retrieved from https://arxiv.org/abs/2303.08774

[3] Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. TopiOCQA: Open-domain conversational question answering with topic switching. *Transactions of the Association for Computational Linguistics* 10 (2022), 468–483. DOI: https://doi.org/10.1162/TACL_A_00471

[4] Amazon. 2023. Amazon. Retrieved from https://www.amazon.com

[5] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. arXiv:2310.11511. Retrieved from https://arxiv.org/abs/2310.11511

[6] Arian Askari, Chuan Meng, Mohammad Aliannejadi, Zhaochun Ren, Evangelos Kanoulas, and Suzan Verberne. 2024. Generative retrieval with few-shot indexing. arXiv:2408.02152. Retrieved from https://arxiv.org/abs/2408.02152

[7] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Yitzhak Gadre, Shiori Sagawa, et al. 2023. OpenFlamingo: An open-source framework for training large autoregressive vision-language models. arXiv:2308.01390. Retrieved from https://arxiv.org/abs/2308.01390

[8] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. arXiv:2309.16609. Retrieved from https://arxiv.org/abs/2309.16609

[9] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005*. Association for Computational Linguistics, 65–72. Retrieved from https://aclanthology.org/W05-0909/

[10] Garbiel Bénédict, Ruqing Zhang, and Donald Metzler. 2023. Gen-IR@SIGIR 2023: The first workshop on generative information retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 3460–3463.

[11] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of Thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, 17682–17690.

[12] Michele Bevilacqua, Giuseppe Ottaviano, Patrick S. H. Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. In *Proceedings of the Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022 (NeurIPS '22)*, 31668–31683. Retrieved from http://papers.nips.cc/paper_files/paper/2022/hash/cd88d62a2063fdaf7ce6f9068fb15dcd-Abstract-Conference.html

[13] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the International Conference on Machine Learning (ICML '22)*. Proceedings of Machine Learning Research, Vol. 162, PMLR, 2206–2240. Retrieved from https://proceedings.mlr.press/v162/borgeaud22a.html

[14] A. Z. Broder. 1997. On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, 21–29. DOI: https://doi.org/10.1109/SEQUEN.1997.666900

[15] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proceedings of the Advances in Neural Information Processing Systems 33*, pp. 1877–1901.

[16] Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP '21), Virtual Event*. Association for Computational Linguistics, 6491–6506. DOI: https://doi.org/10.18653/V1/2021.EMNLP-MAIN.522

[17] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *Proceedings of the 9th International Conference on Learning Representations (ICLR '21), Virtual Event*. OpenReview.net. Retrieved from https://openreview.net/forum?id=5k8F6UU39V

[18] Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun. 2023. A comprehensive survey of AI-Generated Content (AIGC): A history of generative AI from GAN to ChatGPT. arXiv:2303.04226. Retrieved from https://arxiv.org/abs/2303.04226

[19] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Security Symposium (USENIX Security '19)*. USENIX Association, 267–284. Retrieved from https://www.usenix.org/conference/usenixsecurity19/presentation/carlini

[20] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, et al. 2021. Extracting training data from large language models. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security '21)*. USENIX Association, 2633–2650. Retrieved from https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting

[21] Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. arXiv:2006.14799. Retrieved from https://arxiv.org/abs/2006.14799

[22] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. arXiv:2307.03109. Retrieved from https://arxiv.org/abs/2307.03109

[23] Anthony Chen, Panupong Pasupat, Sameer Singh, Hongrae Lee, and Kelvin Guu. 2023. PURR: Efficiently editing language model hallucinations by denoising language model corruptions. arXiv:2305.14908. Retrieved from https://arxiv.org/abs/2305.14908

[24] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023. Continual learning for generative retrieval over dynamic corpora. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 306–315.

[25] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yiqun Liu, Yixing Fan, and XueqiCheng. 2023. A unified generative retriever for knowledge-intensive language tasks via prompt learning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. ACM, 1448–1457. DOI: https://doi.org/10.1145/3539618.3591631

[26] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022. GERE: Generative evidence retrieval for fact verification. arXiv:2204.05511. Retrieved from https://arxiv.org/abs/2204.05511

[27] 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 *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. ACM, 191–200. DOI: https://doi.org/10.1145/3511808.3557271

[28] Xiaoyang Chen, Yanjiang Liu, Ben He, Le Sun, and Yingfei Sun. 2023. Understanding differential search index for text retrieval. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, 10701–10717. DOI: https://doi.org/10.18653/V1/2023.FINDINGS-ACL.681

[29] Xiaoyang Chen, Yanjiang Liu, Ben He, Le Sun, and Yingfei Sun. 2023. Understanding differential search index for text retrieval. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, 10701–10717. DOI: https://doi.org/10.18653/V1/2023.FINDINGS-ACL.681

[30] Zijie Chen, Lichao Zhang, Fangsheng Weng, Lili Pan, and Zhenzhong Lan. 2023. Tailored visions: Enhancing text-to-image generation with personalized prompt rewriting. arXiv:2310.08129. Retrieved from https://arxiv.org/abs/2310.08129

[31] Pengyu Cheng, Jiawen Xie, Ke Bai, Yong Dai, and Nan Du. 2023. Everyone deserves a reward: Learning customized human preferences. arXiv:2309.03126. Retrieved from https://arxiv.org/abs/2309.03126

[32] Qinyuan Cheng, Tianxiang Sun, Wenwei Zhang, Siyin Wang, Xiangyang Liu, Mozhi Zhang, Junliang He, Mianqiu Huang, Zhangyue Yin, Kai Chen, et al. 2023. Evaluating hallucinations in Chinese large language models. arXiv:2310.03368. Retrieved from https://arxiv.org/abs/2310.03368

[33] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research* 24 (2023), 240:1–240:113. Retrieved from http://jmlr.org/papers/v24/22-1144.html

[34] Konstantina Christakopoulou, Alberto Lalama, Cj Adams, Iris Qu, Yifat Amir, Samer Chucri, Pierce Vollucci, Fabio Soldo, Dina Bseiso, Sarah Scodel, et al. 2023. Large language models for user interest journeys. arXiv:2305.15498. Retrieved from https://arxiv.org/abs/2305.15498

[35] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 4299–4307. Retrieved from https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html

[36] Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2023. DoLa: Decoding by contrasting layers improves factuality in large language models. arXiv:2309.03883. Retrieved from https://arxiv.org/abs/2309.03883

[37] Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake A. Hechtman, Trevor Cai, Sebastian Borgeaud, et al.. 2022. Unified scaling laws for routed language models. arXiv:2202.01169. Retrieved from https://arxiv.org/abs/2202.01169

[38] Andrea Cossu, Tinne Tuytelaars, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, and Davide Bacciu. 2022. Continual pre-training mitigates forgetting in language and vision. arXiv:2205.09357. Retrieved from https://arxiv.org/abs/2205.09357

[39] Nick Craswell. 2009. Mean reciprocal rank. In *Encyclopedia of Database Systems*. Springer, Boston, MA, 1703. DOI: https://doi.org/10.1007/978-0-387-39940-9_488

[40] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. arXiv:2102.07662. Retrieved from https://arxiv.org/abs/2102.07662

[41] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. arXiv:2003.07820. Retrieved from https://arxiv.org/abs/2003.07820

[42] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for RAG systems. arXiv:2401.14887. Retrieved from https://arxiv.org/abs/2401.14887

[43] Peng Cui and Mrinmaya Sachan. 2023. Adaptive and personalized exercise generation for online language learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*

(ACL '23). Association for Computational Linguistics, 10184–10198. DOI: https://doi.org/10.18653/V1/2023.ACL-LONG.567

[44] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL '22)*. Association for Computational Linguistics, 8493–8502. DOI: https://doi.org/10.18653/V1/2022.ACL-LONG.581

[45] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering ChatGPT's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23)*. ACM, 1126–1132. DOI: https://doi.org/10.1145/3604915.3610646

[46] Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. 2024. Unifying bias and unfairness in information retrieval: A survey of challenges and opportunities with large language models. arXiv:2404.11457. Retrieved from https://arxiv.org/abs/2404.11457

[47] Yuhao Dan, Zhikai Lei, Yiyang Gu, Yong Li, Jianghao Yin, Jiaju Lin, Linhao Ye, Zhiyan Tie, Yougen Zhou, Yilei Wang, et al. 2023. EduChat: A large-scale language model-based chatbot system for intelligent education. arXiv:2308.02773. Retrieved from https://arxiv.org/abs/2308.02773

[48] Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-Verification reduces hallucination in large language models. arXiv:2309.11495. Retrieved from https://arxiv.org/abs/2309.11495

[49] Joseph Dien. 2023. Generative artificial intelligence as a plagiarism problem. *Biological Psychology* 181 (2023), 108621.

[50] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of Wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations*.

[51] Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, and Xueqi Cheng. 2024. Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models. arXiv:2402.10612. Retrieved from https://arxiv.org/abs/2402.10612

[52] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence* 5, 3 (2023), 220–235. DOI: https://doi.org/10.1038/S42256-023-00626-4

[53] Guanting Dong, Xiaoshuai Song, Yutao Zhu, Runqi Qiao, Zhicheng Dou, and Ji-Rong Wen. 2024. Toward general instruction-following alignment for retrieval-augmented generation. arXiv:2410.09584. Retrieved from https://arxiv.org/abs/2410.09584

[54] Guanting Dong, Yutao Zhu, Chenghao Zhang, Zechen Wang, Zhicheng Dou, and Ji-Rong Wen. 2024. Understand what LLM needs: Dual preference alignment for retrieval-augmented generation. arXiv:2406.18676. Retrieved from https://arxiv.org/abs/2406.18676

[55] Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. In *Findings of the Association for Computational Linguistics (EMNLP '22)*. Association for Computational Linguistics, 5937–5947. DOI: https://doi.org/10.18653/V1/2022.FINDINGS-EMNLP.438

[56] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss library. arXiv:2401.08281. Retrieved from https://arxiv.org/abs/2401.08281

[57] Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2021. GLaM: Efficient scaling of language models with mixture-of-experts. arXiv:2112.06905. Retrieved from https://arxiv.org/abs/2112.06905

[58] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 herd of models. arXiv:2407.21783. Retrieved from https://arxiv.org/abs/2407.21783

[59] Yihao Fang, Stephen W. Thomas, and Xiaodan Zhu. 2024. HGOT: Hierarchical graph of thoughts for retrieval-augmented in-context learning in factuality evaluation. arXiv:2402.09390. Retrieved from https://arxiv.org/abs/2402.09390

[60] William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research* 23, 1, Article 120 (Jan. 2022), 39 pages.

[61] Paolo Ferragina and Giovanni Manzini. 2000. Opportunistic data structures with applications. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS '00)*. IEEE Computer Society, 390–398. DOI: https://doi.org/10.1109/SFCS.2000.892127

[62] Constanza Fierro, Reinald Kim Amplayo, Fantine Huot, Nicola De Cao, Joshua Maynez, Shashi Narayan, and Mirella Lapata. 2024. Learning to plan and generate text with citations. arXiv:2404.03381. Retrieved from https://arxiv.org/abs/2404.03381

[63] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse lexical and expansion model for information retrieval. arXiv:2109.10086. Retrieved from https://arxiv.org/abs/2109.10086
[64] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), Virtual Event*. ACM, 2288–2292. DOI: https://doi.org/10.1145/3404835.3463098
[65] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. GPTScore: Evaluate as you desire. arXiv:2302.04166. Retrieved from https://arxiv.org/abs/2302.04166
[66] Tingchen Fu, Xueliang Zhao, Chongyang Tao, Ji-Rong Wen, and Rui Yan. 2022. There are a thousand hamlets in a thousand people's eyes: Enhancing knowledge-grounded dialogue with personal memory. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL '22)*. Association for Computational Linguistics, 3901–3913. DOI: https://doi.org/10.18653/V1/2022.ACL-LONG.270
[67] Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md. Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2023. Bias and fairness in large language models: A survey. arXiv:2309.00770. Retrieved from https://arxiv.org/abs/2309.00770
[68] Difei Gao, Lei Ji, Luowei Zhou, Kevin Qinghong Lin, Joya Chen, Zihan Fan, and Mike Zheng Shou. 2023. AssistGPT: A general multi-modal assistant that can plan, execute, inspect, and learn. arXiv:2306.08640. Retrieved from https://arxiv.org/abs/2306.08640
[69] Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y. Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL '23)*. Association for Computational Linguistics, 16477–16508. DOI: https://doi.org/10.18653/V1/2023.ACL-LONG.910
[70] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, 6465–6488. Retrieved from https://aclanthology.org/2023.emnlp-main.398
[71] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. arXiv:2312.10997. Retrieved from https://arxiv.org/abs/2312.10997
[72] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2014. Optimized product quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 4 (2014), 744–755. DOI: https://doi.org/10.1109/TPAMI.2013.240
[73] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5). In: *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys '22)*. ACM, 299–315. DOI: https://doi.org/10.1145/3523227.3546767
[74] Lukas Gienapp, Harrisen Scells, Niklas Deckers, Janek Bevendorff, Shuai Wang, Johannes Kiesel, Shahbaz Syed, Maik Frobe, Guide Zucoon, Benno Stein, et al. 2023. Evaluating generative ad hoc information retrieval. arXiv:2311.04694. Retrieved from https://api.semanticscholar.org/CorpusID:265050661
[75] Google. 2023. Google. Retrieved from https://www.google.com
[76] Google. 2023. YouTube. Retrieved from https://www.youtube.com
[77] Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. arXiv:2312.00752. Retrieved from https://arxiv.org/abs/2312.00752
[78] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. arXiv:2306.11644. Retrieved from https://arxiv.org/abs/2306.11644
[79] Jiafeng Guo, Changjiang Zhou, Ruqing Zhang, Jiangui Chen, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024. CorpusBrain++: A continual generative pre-training framework for knowledge-intensive language tasks. arXiv:2402.16767. Retrieved from https://arxiv.org/abs/2402.16767
[80] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the International Conference on Machine Learning*. PMLR, 3929–3938.
[81] Shibo Hao, Tianyang Liu, Zhen Wang, and Zhiting Hu. 2023. ToolkenGPT: Augmenting frozen language models with massive tools via tool embeddings. In *Proceedings of the Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023 (NeurIPS '23)*, 45870–45894. Retrieved from http://papers.nips.cc/paper_files/paper/2023/hash/8fd1a81c882cd45f64958da6284f4a3f-Abstract-Conference.html
[82] Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with GRACE: Lifelong model editing with discrete key-value adaptors. In *Proceedings of the Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023 (NeurIPS '23)*, 47934–47959. Retrieved from http://papers.nips.cc/paper_files/paper/2023/hash/95b6e2ff961580e03c0a662a63a71812-Abstract-Conference.html

[83] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *Proceedings of the 9th International Conference on Learning Representations (ICLR '21), Virtual Event*. OpenReview.net. Retrieved from https://openreview.net/forum?id=d7KBjmI3GmQ

[84] William R. Hersh. 2023. Search still matters: Information retrieval in the era of generative AI. arXiv:2311.18550. Retrieved from https://arxiv.org/abs/2311.18550

[85] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, 782–792.

[86] Chengyu Huang, Zeqiu Wu, Yushi Hu, and Wenya Wang. 2024. Training language models to generate text with citations via fine-grained rewards. arXiv:2402.04315. Retrieved from https://arxiv.org/abs/2402.04315

[87] Jie Huang and Kevin Chen-Chuan Chang. 2023. Citation: A key to building responsible and accountable large language models. arXiv:2307.02185. Retrieved from https://arxiv.org/abs/2307.02185

[88] Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are large pre-trained language models leaking your personal information? In *Findings of the Association for Computational Linguistics: EMNLP 2022*. Association for Computational Linguistics, 2038–2047. DOI: https://doi.org/10.18653/V1/2022.FINDINGS-EMNLP.148

[89] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. arXiv:2311.05232. Retrieved from https://arxiv.org/abs/2311.05232

[90] Qiushi Huang, Shuai Fu, Xubo Liu, Wenwu Wang, Tom Ko, Yu Zhang, and LilianH. Y. Tang. 2023. Learning retrieval augmentation for personalized dialogue generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, 2523–2540. DOI: https://doi.org/10.18653/V1/2023.EMNLP-MAIN.154

[91] Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. 2023. C-Eval: A multi-level multi-discipline Chinese evaluation suite for foundation models. In *Proceedings of the Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023 (NeurIPS '23)*, 62991–63010. Retrieved from http://papers.nips.cc/paper_files/paper/2023/hash/c6ec1844bec96d6d32ae95ae694e23d8-Abstract-Datasets_and_Benchmarks.html

[92] Yue Huang, Qihui Zhang, Philip S. Yu, and Lichao Sun. 2023. TrustGPT: A benchmark for trustworthy and responsible large language models. arXiv:2306.11507. Retrieved from https://arxiv.org/abs/2306.11507

[93] Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-Patcher: One mistake worth one neuron. In *Proceedings of the 11th International Conference on Learning Representations (ICLR '23)*. OpenReview.net. Retrieved from https://openreview.net/pdf?id=4oYUGeGBPm

[94] Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. arXiv:2007.01282. Retrieved from https://arxiv.org/abs/2007.01282

[95] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation* 3, 1 (1991), 79–87.

[96] Palak Jain, Livio Soares, and Tom Kwiatkowski. 2023. 1-PAGER: One pass answer generation and evidence retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 14529–14543. DOI: https://doi.org/10.18653/V1/2023.FINDINGS-EMNLP.967

[97] Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2023. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. arXiv:2310.11564. Retrieved from https://arxiv.org/abs/2310.11564

[98] Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2023. Exploring the benefits of training expert language models over instruction tuning. In *Proceedings of the International Conference on Machine Learning*. PMLR, 14702–14729.

[99] Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. 2021. Towards continual knowledge learning of language models. arXiv:2110.03215. Retrieved from https://arxiv.org/abs/2110.03215

[100] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems* 20, 4 (2002), 422–446. DOI: https://doi.org/10.1145/582415.582418

[101] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. 2011. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 1 (2011), 117–128. DOI: https://doi.org/10.1109/TPAMI.2010.57

[102] Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset. In *Proceedings of the Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information*

Processing Systems 2023 (NeurIPS '23), 24678–24704. Retrieved from http://papers.nips.cc/paper_files/paper/2023/hash/4dbb61cb68671edc4ca3712d70083b9f-Abstract-Datasets_and_Benchmarks.html

[103] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys* 55, 12 (2023), 1–38.

[104] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. arXiv:2310.06825. Retrieved from https://arxiv.org/abs/2310.06825

[105] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. arXiv:2401.04088. Retrieved from https://arxiv.org/abs/2401.04088

[106] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-Blender: Ensembling large language models with pairwise comparison and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL '23)*, 14165–14178.

[107] Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. arXiv:2310.06839. Retrieved from https://arxiv.org/abs/2310.06839

[108] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 13358–13376. DOI: https://doi.org/10.18653/v1/2023.emnlp-main.825

[109] Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. 2023. StructGPT: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 9237–9251. DOI: https://doi.org/10.18653/v1/2023.emnlp-main.574

[110] Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, 7969–7992. Retrieved from https://aclanthology.org/2023.emnlp-main.495

[111] Bowen Jin, Hansi Zeng, Guoyin Wang, Xiusi Chen, Tianxin Wei, Ruirui Li, Zhengyang Wang, Zheng Li, Yang Li, Hanqing Lu, et al. 2023. Language models as semantic indexers. arXiv:2310.07815. Retrieved from https://arxiv.org/abs/2310.07815

[112] Jiajie Jin, Yutao Zhu, Xinyu Yang, Chenghao Zhang, and Zhicheng Dou. 2024. FlashRAG: A modular toolkit for efficient retrieval-augmented generation research. arXiv:2405.13576. Retrieved from https://arxiv.org/abs/2405.13576

[113] Jiajie Jin, Yutao Zhu, Yujia Zhou, and Zhicheng Dou. 2024. BIDER: Bridging knowledge inconsistency for efficient retrieval-augmented LLMs via key supporting evidence. arXiv:2402.12174. Retrieved from https://arxiv.org/abs/2402.12174

[114] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 1601–1611.

[115] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. arXiv:2001.08361. Retrieved from https://arxiv.org/abs/2001.08361

[116] 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 *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, 6769–6781.

[117] Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. 2023. RealTime QA: What's the answer right now? In *Proceedings of the Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023 (NeurIPS '23)*, 49025–49043. Retrieved from http://papers.nips.cc/paper_files/paper/2023/hash/9941624ef7f867a502732b5154d30cb7-Abstract-Datasets_and_Benchmarks.html

[118] Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. arXiv:2302.03241. Retrieved from https://arxiv.org/abs/2302.03241

[119] Krishnaram Kenthapadi, Himabindu Lakkaraju, and Nazneen Rajani. 2023. Generative ai meets responsible AI: Practical challenges and opportunities. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5805–5806.

[120] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 4171–4186.

[121] Muhammad Khalifa, David Wadden, Emma Strubell, Honglak Lee, Lu Wang, Iz Beltagy, and Hao Peng. 2024. Source-Aware Training Enables Knowledge Attribution in Language Models. Retrieved from https://api.semanticscholar.org/CorpusID:268819100

[122] Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. 2023. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 996–1009. DOI: https://doi.org/10.18653/v1/2023.emnlp-main.63

[123] Varsha Kishore, Chao Wan, Justin Lovelace, Yoav Artzi, and Kilian Q. Weinberger. 2023. IncDSI: Incrementally updatable document retrieval. In *Proceedings of the International Conference on Machine Learning (ICML '23)*. Proceedings of Machine Learning Research, Vol. 202, PMLR, 17122–17134. Retrieved from https://proceedings.mlr.press/v202/kishore23a.html

[124] Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 8460–8478. DOI: https://doi.org/10.18653/v1/2022.acl-long.579

[125] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural Questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics* 7 (2019), 453–466.

[126] Tin Lai, Yukun Shi, Zicong Du, Jiajie Wu, Ken Fu, Yichao Dou, and Ziqi Wang. 2023. Psy-LLM: Scaling up global mental health psychological services with AI-based large language models. arXiv:2307.11991. Retrieved from https://arxiv.org/abs/2307.11991

[127] Tian Lan, Deng Cai, Yan Wang, Heyan Huang, and Xian-Ling Mao. 2023. Copy is all you need. In *Proceedings of the 11th International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=CROlOA9Nd8C

[128] Dongyub Lee, Taesun Whang, Chanhee Lee, and Heuiseok Lim. 2023. Towards reliable and fluent large language models: Incorporating feedback learning loops in QA systems. arXiv:2309.06384. Retrieved from https://arxiv.org/abs/2309.06384

[129] Hyunji Lee, JaeYoung Kim, Hoyeon Chang, Hanseok Oh, Sohee Yang, Vladimir Karpukhin, Yi Lu, and Minjoon Seo. 2023. Nonparametric decoding for generative retrieval. In *Findings of the Association for Computational Linguistics (ACL '23)*. Association for Computational Linguistics, 12642–12661. DOI: https://doi.org/10.18653/V1/2023.FINDINGS-ACL.801

[130] Hyunji Lee, Sohee Yang, Hanseok Oh, and Minjoon Seo. 2022. Generative multi-hop retrieval. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP '22)*. Association for Computational Linguistics, 1417–1436. DOI: https://doi.org/10.18653/V1/2022.EMNLP-MAIN.92

[131] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL '22)*. Association for Computational Linguistics, 8424–8445. DOI: https://doi.org/10.18653/V1/2022.ACL-LONG.577

[132] Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. In *Proceedings of the Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022 (NeurIPS '22)*, 34586–34599. Retrieved from http://papers.nips.cc/paper_files/paper/2022/hash/df438caa36714f69277daa92d608dd63-Abstract-Conference.html

[133] Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Factuality enhanced language models for open-ended text generation. arXiv:2206.04624. Retrieved from https://arxiv.org/abs/2206.04624

[134] Sunkyung Lee, Minjin Choi, and Jongwuk Lee. 2023. GLEN: Generative retrieval via lexical index learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, 7693–7704. Retrieved from https://aclanthology.org/2023.emnlp-main.477

[135] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. GShard: Scaling giant models with conditional computation and automatic sharding. arXiv:2006.16668. Retrieved from https://arxiv.org/abs/2006.16668

[136] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the Computational Natural Language Learning (CoNLL)*, 333–342.

[137] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the Association for Computational Linguistics (ACL)*, 7871–7880.

[138] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020 (NeurIPS '20), Virtual*, 9459–9474. Retrieved from https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[139] Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, and Michael Bendersky. 2023. Automatic prompt rewriting for personalized text generation. arXiv:2310.00152. Retrieved from https://arxiv.org/abs/2310.00152

[140] Cheng Li, Mingyang Zhang, Qiaozhu Mei, Yaqing Wang, Spurthi Amba Hombaiah, Yi Liang, and Michael Bendersky. 2023. Teach LLMs to personalize—An approach inspired by writing education. arXiv:2308.07968. Retrieved from https://arxiv.org/abs/2308.07968

[141] Dongfang Li, Zetian Sun, Baotian Hu, Zhenyu Liu, Xinshuo Hu, Xuebo Liu, and Min Zhang. 2024. Improving attributed text generation of large language models via preference learning. arXiv:2403.18381. Retrieved from https://arxiv.org/abs/2403.18381

[142] Dongfang Li, Zetian Sun, Xinshuo Hu, Zhenyu Liu, Ziyang Chen, Baotian Hu, Aiguo Wu, and Min Zhang. 2023. A survey of large language models attribution. arXiv:2311.03731. Retrieved from https://arxiv.org/abs/2311.03731

[143] Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, 6449–6464. Retrieved from https://aclanthology.org/2023.emnlp-main.397

[144] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the International Conference on Machine Learning*. PMLR, 12888–12900.

[145] Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni. 2023. GPT4Rec: A generative framework for personalized recommendation and user interests interpretation. arXiv:2304.03879. Retrieved from https://arxiv.org/abs/2304.03879

[146] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-Time Intervention: Eliciting truthful answers from a language model. In *Proceedings of the 37th Conference on Neural Information Processing Systems*. Retrieved from https://openreview.net/forum?id=aLLuYpn83y

[147] Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. API-Bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, 3102–3116. Retrieved from https://aclanthology.org/2023.emnlp-main.187

[148] Pan Li and Alexander Tuzhilin. 2019. Towards controllable and personalized review generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP '19)*. Association for Computational Linguistics, 3235–3243. DOI: https://doi.org/10.18653/V1/D19-1319

[149] Weitao Li, Junkai Li, Weizhi Ma, and Yang Liu. 2024. Citation-enhanced generation for LLM-based chatbots. arXiv:2402.16063. Retrieved from https://arxiv.org/abs/2402.16063

[150] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models. arXiv:2501.05366. Retrieved from https://arxiv.org/abs/2501.05366

[151] Xiaoxi Li, Zhicheng Dou, Yujia Zhou, and Fangchao Liu. 2024. CorpusLM: Towards a unified language model on corpus for knowledge-intensive tasks. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*. Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (Eds.), ACM, 26–37. DOI: https://doi.org/10.1145/3626772.3657778

[152] Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yongkang Wu, Zhonghua Li, Qi Ye, and Zhicheng Dou. 2024. RetroLLM: Empowering large language models to retrieve fine-grained evidence within generation. arXiv:2412.11919. Retrieved from https://arxiv.org/abs/2412.11919

[153] Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024. PMET: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, 18564–18572.

[154] Xiaoxi Li, Yujia Zhou, and Zhicheng Dou. 2024. UniGen: A unified generative framework for retrieval and question answering with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, 8688–8696.

[155] Xiaonan Li, Changtai Zhu, Linyang Li, Zhangyue Yin, Tianxiang Sun, and Xipeng Qiu. 2023. LLatrieval: LLM-verified retrieval for verifiable generation. arXiv:2311.07838. Retrieved from https://arxiv.org/abs/2311.07838

[156] Yongqi Li, Wenjie Wang, Leigang Qu, Liqiang Nie, Wenjie Li, and Tat-Seng Chua. 2024. Generative cross-modal retrieval: Memorizing images in multimodal language models for retrieval and beyond. arXiv:2402.10805. Retrieved from https://arxiv.org/abs/2402.10805

[157] Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023. Generative retrieval for conversational question answering. *Information Processing & Management* 60, 5 (2023), 103475. DOI: https://doi.org/10.1016/J.IPM.2023.103475

[158] Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023. Learning to rank in generative retrieval. arXiv:2306.15222. Retrieved from https://arxiv.org/abs/2306.15222

[159] Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023. Multiview identifiers enhanced generative retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL '23)*. Association for Computational Linguistics, 6636–6648. DOI: https://doi.org/10.18653/V1/2023.ACL-LONG.366

[160] Yongqi Li, Zhen Zhang, Wenjie Wang, Liqiang Nie, Wenjie Li, and Tat-Seng Chua. 2024. Distillation enhanced generative retrieval. arXiv:2402.10769. Retrieved from https://arxiv.org/abs/2402.10769

[161] Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. Retrieved from https://api.semanticscholar.org/CorpusID:964287

[162] Jimmy Lin and Xueguang Ma. 2021. A few brief notes on DeepImpact, COIL, and a conceptual framework for information retrieval techniques. arXiv:2106.14807. Retrieved from https://arxiv.org/abs/2106.14807

[163] Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL '22)*. Association for Computational Linguistics, 3214–3252. DOI: https://doi.org/10.18653/V1/2022.ACL-LONG.229

[164] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *Proceedings of the 13th European Conference on Computer Vision (ECCV '14), Part V*. Lecture Notes in Computer Science, Vol. 8693, Springer, 740–755. DOI: https://doi.org/10.1007/978-3-319-10602-1_48

[165] Yen-Ting Lin and Yun-Nung Chen. 2023. LLM-Eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. arXiv:2305.13711. Retrieved from https://arxiv.org/abs/2305.13711

[166] Adam Liska, Tomás Kociský, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, et al. 2022. StreamingQA: A benchmark for adaptation to new knowledge over time in question answering models. In *Proceedings of the International Conference on Machine Learning (ICML '22)*. Proceedings of Machine Learning Research, Vol. 162, PMLR, 13604–13622. Retrieved from https://proceedings.mlr.press/v162/liska22a.html

[167] Robert Litschko, Max Müller-Eberstein, Rob van der Goot, Leon Weber-Genzel, and Barbara Plank. 2023. Establishing trustworthiness: Rethinking tasks and model evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, 193–203. DOI: https://doi.org/10.18653/V1/2023.EMNLP-MAIN.14

[168] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is ChatGPT a good recommender? A preliminary study. arXiv:2304.10149. Retrieved from https://arxiv.org/abs/2304.10149

[169] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. arXiv:2307.03172. Retrieved from https://arxiv.org/abs/2307.03172

[170] Nelson F. Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 7001–7025. DOI: https://doi.org/10.18653/V1/2023.FINDINGS-EMNLP.467

[171] Qian Liu, Yihong Chen, Bei Chen, Jian-Guang Lou, Zixuan Chen, Bin Zhou, and Dongmei Zhang. 2020. You impress me: Dialogue generation via mutual persona perception. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL '20)*. Association for Computational Linguistics, 1417–1427. DOI: https://doi.org/10.18653/V1/2020.ACL-MAIN.131

[172] Wenhan Liu, Xinyu Ma, Yutao Zhu, Ziliang Zhao, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. 2024. Sliding windows are not the end: Exploring full ranking with long-context large language models. arXiv:2412.14574. Retrieved from https://arxiv.org/abs/2412.14574

[173] Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. WebGLM: Towards an efficient web-enhanced question answering system with human preferences. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*. ACM, 4549–4560. DOI: https://doi.org/10.1145/3580305.3599931

[174] Xin Liu, Daniel McDuff, Geza Kovacs, Isaac R. Galatzer-Levy, Jacob E. Sunshine, Jiening Zhan, Ming-Zher Poh, Shun Liao, Paolo Di Achille, and Shwetak N. Patel. 2023. Large language models are few-shot health learners. arXiv:2305.15525. Retrieved from https://arxiv.org/abs/2305.15525

[175] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Wei Chen, and Xueqi Cheng. 2023. On the robustness of generative retrieval models: An out-of-distribution perspective. arXiv:2306.12756. Retrieved from https://arxiv.org/abs/2306.12756

[176] Zhengliang Liu, Zihao Wu, Mengxuan Hu, Bokai Zhao, Lin Zhao, Tianyi Zhang, Haixing Dai, Xianyan Chen, Ye Shen, Sheng Li, et al. 2023. PharmacyGPT: The AI pharmacist. arXiv:2307.10432. Retrieved from https://arxiv.org/abs/2307.10432

[177] Xinwei Long, Jiali Zeng, Fandong Meng, Zhiyuan Ma, Kaiyan Zhang, Bowen Zhou, and Jie Zhou. 2024. Generative multi-modal knowledge retrieval with large language models. arXiv:2401.08206. Retrieved from https://arxiv.org/abs/2401.08206

[178] Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin B. Clement, Dawn Drain, Daxin Jiang, Duyu Tang, et al. 2021. CodeXGLUE: A machine learning benchmark dataset for code understanding and generation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1 (NeurIPS Datasets and Benchmarks '21), Virtual*. Retrieved from https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/c16a5320fa475530d9583c34fd356ef5-Abstract-round1.html

[179] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics* 9 (2021), 329–345. DOI: https://doi.org/10.1162/TACL_A_00369

[180] Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *Proceedings of the International Conference on Learning Representations*.

[181] Jun-Yu Ma, Jia-Chen Gu, Zhen-Hua Ling, Quan Liu, and Cong Liu. 2023. Untying the reversal curse via bidirectional language model editing. arXiv:2310.10322. Retrieved from https://arxiv.org/abs/2310.10322

[182] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 5303–5315. DOI: https://doi.org/10.18653/v1/2023.emnlp-main.322

[183] Zhengyi Ma, Zhicheng Dou, Yutao Zhu, Hanxun Zhong, and Ji-Rong Wen. 2021. One chatbot per person: Creating personalized chatbots based on implicit user profiles. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), Virtual Event*. ACM, 555–564. DOI: https://doi.org/10.1145/3404835.3462828

[184] Jean Maillard, Vladimir Karpukhin, Fabio Petroni, Wen tau Yih, Barlas Oguz, Veselin Stoyanov, and Gargi Ghosh. 2021. Multi-task retrieval for knowledge-intensive tasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (ACL/IJCNLP '21), Virtual Event*. Association for Computational Linguistics, 1098–1111. DOI: https://doi.org/10.18653/V1/2021.ACL-LONG.89

[185] Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian J. McAuley. 2019. Generating personalized recipes from historical user preferences. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP '19)*. Association for Computational Linguistics, 5975–5981. DOI: https://doi.org/10.18653/V1/D19-1613

[186] Yury A. Malkov and Dmitry A. Yashunin. 2020. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 4 (2020), 824–836. DOI: https://doi.org/10.1109/TPAMI.2018.2889473

[187] Udi Manber and Gene Myers. 1990. Suffix arrays: A new method for on-line string searches. In *Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '90)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 319–327.

[188] Julieta Martinez, Holger H. Hoos, and James J. Little. 2014. Stacked quantizers for compositional vector compression. arXiv:1411.2173. Retrieved from http://arxiv.org/abs/1411.2173

[189] Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2775–2779. DOI: https://doi.org/10.18653/V1/D18-1298

[190] Vittorio Mazzia, Alessandro Pedrani, Andrea Caciolai, Kay Rottmann, and Davide Bernardi. 2023. A survey on knowledge editing of neural networks. arXiv:2310.19704. Retrieved from https://arxiv.org/abs/2310.19704

[191] Sanket Vaibhav Mehta, Jai Gupta, Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Jinfeng Rao, Marc Najork, Emma Strubell, and Donald Metzler. 2022. DSI++: Updating transformer memory with new documents. arXiv:2212.09744. Retrieved from https://arxiv.org/abs/2212.09744

[192] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Proceedings of the Advances in Neural Information Processing Systems 35: Annual Conference on Neural*

Information Processing Systems 2022 (NeurIPS '22), 17359–17372. Retrieved from http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html

[193] Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *Proceedings of the 11th International Conference on Learning Representations (ICLR '23)*. OpenReview.net. Retrieved from https://openreview.net/pdf?id=MkbcAHIYgyS

[194] Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. 2021. Rethinking search: Making domain experts out of dilettantes. *ACM SIGIR Forum* 55, 1–27.

[195] Microsoft. 2023. Bing. Retrieved from https://www.bing.com

[196] Microsoft. 2023. Bing Chat. Retrieved from https://www.bing.com/new

[197] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, 12076–12100. Retrieved from https://aclanthology.org/2023.emnlp-main.741

[198] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022. Fast model editing at scale. In *Proceedings of the 10th International Conference on Learning Representations (ICLR '22), Virtual Event*. OpenReview.net. Retrieved from https://openreview.net/forum?id=0DcZxeWfOPt

[199] Fengran Mo, Kelong Mao, Ziliang Zhao, Hongjin Qian, Haonan Chen, Yiruo Cheng, Xiaoxi Li, Yutao Zhu, Zhicheng Dou, and Jian-Yun Nie. 2024. A survey of conversational search. arXiv:2410.15576. Retrieved from https://arxiv.org/abs/2410.15576

[200] Jisoo Mok, Jaeyoung Do, Sungjin Lee, Tara Taghavi, Seunghak Yu, and Sungroh Yoon. 2023. Large-scale lifelong learning of in-context instructions and how to tackle it. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12573–12589.

[201] Sheshera Mysore, Zhuoran Lu, Mengting Wan, Longqi Yang, Steve Menezes, Tina Baghaee, Emmanuel Barajas Gonzalez, Jennifer Neville, and Tara Safavi. 2023. PEARL: Personalizing large language model writing assistants with generation-calibrated retrievers. arXiv:2311.09180. Retrieved from https://arxiv.org/abs/2311.09180

[202] Usama Nadeem, Noah Ziems, and Shaoen Wu. 2022. CodeDSI: Differentiable code search. arXiv:2210.00328. Retrieved from https://arxiv.org/abs/2210.00328

[203] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. WebGPT: Browser-assisted question-answering with human feedback. arXiv:2112.09332. Retrieved from https://arxiv.org/abs/2112.09332

[204] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated MAchine Reading COmprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS '16)*. CEUR Workshop Proceedings, Vol. 1773, CEUR-WS.org. Retrieved from https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf

[205] Thong Nguyen and Andrew Yates. 2023. Generative retrieval as dense retrieval. arXiv:2306.11397. Retrieved from https://arxiv.org/abs/2306.11397

[206] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, et al. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP '22)*. Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 9844–9855. DOI: https://doi.org/10.18653/V1/2022.EMNLP-MAIN.669

[207] Rodrigo Nogueira and Jimmy Lin. 2019. From doc2query to docTTTTTquery. *Online Preprint* 6 (2019), 2.

[208] OpenAI. 2022. Introducing ChatGPT. Retrieved from https://openai.com/blog/chatgpt

[209] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022 (NeurIPS '22)*. Retrieved from http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html

[210] Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2023. Fine-tuning or retrieval? Comparing knowledge injection in LLMs. arXiv:2312.05934. Retrieved from https://arxiv.org/abs/2312.05934

[211] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. ACL, 311–318. DOI: https://doi.org/10.3115/1073083.1073135

[212] Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2023. Gorilla: Large language model connected with massive APIs. arXiv:2305.15334. Retrieved from https://arxiv.org/abs/2305.15334

[213] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: Outperforming curated corpora with web data only. In *Proceedings of the Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023 (NeurIPS '23)*, 79155–79172. Retrieved from http://papers.nips.cc/paper_files/paper/2023/hash/fa3ed726cc5073b9c31e3e49a807789c-Abstract-Datasets_and_Benchmarks.html

[214] Bohao Peng, Zhuotao Tian, Shu Liu, Mingchang Yang, and Jiaya Jia. 2024. Scalable language model with generalized continual learning. arXiv:2404.07470. Retrieved from https://arxiv.org/abs/2404.07470

[215] Denis Peskoff and Brandon Stewart. 2023. Credible without credit: Domain experts assess generative language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (ACL '23)*. Association for Computational Linguistics, 427–438. DOI: https://doi.org/10.18653/V1/2023.ACL-SHORT.37

[216] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2021. KILT: A benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2523–2544. DOI: https://doi.org/10.18653/v1/2021.naacl-main.200

[217] Sebastian Porsdam Mann, Brian D. Earp, Nikolaj Møller, Suren Vynn, and Julian Savulescu. 2023. AUTOGEN: A personalized large language model for academic enhancement—Ethics and proof of principle. *The American Journal of Bioethics* 23, 10 (2023), 28–41.

[218] Ronak Pradeep, Kai Hui, Jai Gupta, Ádám D. Lelkes, Honglei Zhuang, Jimmy Lin, Donald Metzler, and Vinh Q. Tran. 2023. How does generative retrieval scale to millions of passages? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, 1305–1321. Retrieved from https://aclanthology.org/2023.emnlp-main.83

[219] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 5687–5711. DOI: https://doi.org/10.18653/v1/2023.findings-emnlp.378

[220] Cheng Qian, Chi Han, Yi Fung, Yujia Qin, Zhiyuan Liu, and Heng Ji. 2023. CREATOR: Tool creation for disentangling abstract and concrete reasoning of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 6922–6939. DOI: https://doi.org/10.18653/v1/2023.findings-emnlp.462

[221] Hongjing Qian, Yutao Zhu, Zhicheng Dou, Haoqi Gu, Xinyu Zhang, Zheng Liu, Ruofei Lai, Zhao Cao, Jian-Yun Nie, and Ji-Rong Wen. 2023. WebBrain: Learning to generate factually correct articles for queries by grounding on large web corpus. arXiv:2304.04358. Retrieved from https://arxiv.org/abs/2304.04358

[222] Shanbao Qiao, Xuebing Liu, and Seung-Hoon Na. 2023. DiffusionRet: Diffusion-enhanced generative retriever using constrained decoding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 9515–9529. Retrieved from https://aclanthology.org/2023.findings-emnlp.638

[223] Yujia Qin, Zihan Cai, Dian Jin, Lan Yan, Shihao Liang, Kunlun Zhu, Yankai Lin, Xu Han, Ning Ding, Huadong Wang, Ruobing Xie, Fanchao Qi, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. WebCPM: Interactive web search for Chinese long-form question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 8968–8988. DOI: https://doi.org/10.18653/v1/2023.acl-long.499

[224] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. arXiv:2307.16789. Retrieved from https://arxiv.org/abs/2307.16789

[225] Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. InFoBench: Evaluating instruction following ability in large language models. arXiv:2401.03601. Retrieved from https://arxiv.org/abs/2401.03601

[226] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

[227] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019), 9.

[228] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Proceedings of the 37th Conference on Neural Information Processing Systems*. Retrieved from https://openreview.net/forum?id=HPuSIXJaa9

[229] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21 (2020), 140:1–140:67. Retrieved from http://jmlr.org/papers/v21/20-074.html

[230] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, *Volume 2: Short Papers (ACL '18)*. Association for Computational Linguistics, 784–789. DOI: https://doi.org/10.18653/V1/P18-2124

[231] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, et al. 2023. Recommender systems with generative retrieval. In *Proceedings of the Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023 (NeurIPS '23)*, 10299–10315. Retrieved from http://papers.nips.cc/paper_files/paper/2023/hash/20dcab0f14046a5c6b02b61da9f13229-Abstract-Conference.html

[232] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. arXiv:2302.00083. Retrieved from https://arxiv.org/abs/2302.00083

[233] Juan Ramos. 2003. Using TF-IDF to determine word relevance in document queries. In *Proceedings of the 1st Instructional Conference on Machine Learning*, Vol. 242. Citeseer, 29–48.

[234] Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. arXiv:2301.12314. Retrieved from https://arxiv.org/abs/2301.12314

[235] Ruiyang Ren, Wayne Xin Zhao, Jing Liu, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. TOME: A two-stage approach for model-based retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL '23)*, 6102–6114. DOI: https://doi.org/10.18653/V1/2023.ACL-LONG.336

[236] Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389. DOI: https://doi.org/10.1561/1500000019

[237] Nafis Sadeq, Byungkyu Kang, Prarit Lamba, and Julian J. McAuley. 2023. Unsupervised improvement of factual knowledge in language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL '23)*. Association for Computational Linguistics, 2952–2961. DOI: https://doi.org/10.18653/V1/2023.EACL-MAIN.215

[238] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. LaMP: When large language models meet personalization. arXiv:2304.11406. Retrieved from https://arxiv.org/abs/2304.11406

[239] Malik Sallam, Nesreen A. Salim, Ala'a B. Al-Tammemi, Muna M. Barakat, Diaa Fayyad, Souheil Hallit, Harapan Harapan, Rabih Hallit, and Azmi Mahafzah. 2023. ChatGPT output regarding compulsory vaccination and COVID-19 vaccine conspiracy: A descriptive study at the outset of a paradigm shift in online search for information. *Cureus* 15 (2023). Retrieved from https://api.semanticscholar.org/CorpusID:256897987

[240] Gerard Salton, Edward A. Fox, and Harry Wu. 1983. Extended Boolean information retrieval. *Communication of the ACM* 26, 11 (1983), 1022–1036. DOI: https://doi.org/10.1145/182.358466

[241] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. BLOOM: A 176B-parameter open-access multilingual language model. arXiv:2211.05100. Retrieved from https://arxiv.org/abs/2211.05100

[242] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. Are emergent abilities of large language models a mirage? In *Proceedings of the Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023 (NeurIPS '23)*, 55565–55581. Retrieved from http://papers.nips.cc/paper_files/paper/2023/hash/adc98a266f45005c403b8311ca7e8bd7-Abstract-Conference.html

[243] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. arXiv:2302.04761. Retrieved from https://arxiv.org/abs/2302.04761

[244] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. arXiv:1707.06347. Retrieved from http://arxiv.org/abs/1707.06347

[245] Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 7881–7892. DOI: https://doi.org/10.18653/v1/2020.acl-main.704

[246] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. arXiv:2305.15294. Retrieved from https://arxiv.org/abs/2305.15294

[247] Xiaoteng Shen, Rui Zhang, Xiaoyan Zhao, Jieming Zhu, and Xi Xiao. 2024. PMG: Personalized multimodal generation with large language models. arXiv:2404.08677. Retrieved from https://arxiv.org/abs/2404.08677

[248] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. HuggingGPT: Solving AI tasks with ChatGPT and its friends in hugging face. In *Proceedings of the Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023 (NeurIPS '23)*, 38154–38180. Retrieved from http://papers.nips.cc/paper_files/paper/2023/hash/77c33e6a367922d003ff102ffb92b658-Abstract-Conference.html

[249] Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen tau Yih. 2023. Trusting your evidence: Hallucinate less with context-aware decoding. arXiv:2305.14739. Retrieved from https://arxiv.org/abs/2305.14739

[250] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2023. REPLUG: Retrieval-augmented black-box language models. arXiv:2301.12652. Retrieved from https://arxiv.org/abs/2301.12652

[251] Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. 2022. BlenderBot 3: A deployed conversational agent that continually learns to responsibly engage. arXiv:2208.03188. Retrieved from https://arxiv.org/abs/2208.03188

[252] Zihua Si, Zhongxiang Sun, Jiale Chen, Guozhang Chen, Xiaoxue Zang, Kai Zheng, Yang Song, Xiao Zhang, and Jun Xu. 2023. Generative retrieval with semantic tree-structured item identifiers via contrastive learning. arXiv:2309.13375. Retrieved from https://arxiv.org/abs/2309.13375

[253] Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Kumar Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2022. Large language models encode clinical knowledge. arXiv:2212.13138. Retrieved from https://arxiv.org/abs/2212.13138

[254] Aviv Slobodkin, Eran Hirsch, Arie Cattan, Tal Schuster, and Ido Dagan. 2024. Attribute first, then generate: Locally-attributable grounded text generation. arXiv:2403.17104. Retrieved from https://arxiv.org/abs/2403.17104

[255] EuiYul Song, Sangryul Kim, Haeju Lee, Joonkee Kim, and James Thorne. 2024. Re3val: Reinforced and reranked generative retrieval. arXiv:2401.16979. Retrieved from https://arxiv.org/abs/2401.16979

[256] Yifan Song, Weimin Xiong, Dawei Zhu, Wenhao Wu, Han Qian, Mingbo Song, Hailiang Huang, Cheng Li, Ke Wang, Rong Yao, et al. 2023. RestGPT: Connecting large language models with real-world RESTful APIs. arXiv:2306.06624. Retrieved from https://arxiv.org/abs/2306.06624

[257] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. arXiv:2206.04615. Retrieved from https://arxiv.org/abs/2206.04615

[258] Alane Suhr and Yoav Artzi. 2024. Continual learning for instruction following from realtime feedback. In *Proceedings of the Advances in Neural Information Processing Systems 36*, 32340–32359.

[259] Hao Sun, Hengyi Cai, Bo Wang, Yingyan Hou, Xiaochi Wei, Shuaiqiang Wang, Yan Zhang, and Dawei Yin. 2023. Towards verifiable text generation with evolving memory and self-reflection. arXiv:2312.09075. Retrieved from https://arxiv.org/abs/2312.09075

[260] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-Graph: Deep and responsible reasoning of large language model with knowledge graph. arXiv:2307.07697. Retrieved from https://arxiv.org/abs/2307.07697

[261] LichaoSun, YueHuang, HaoranWang, SiyuanWu, QihuiZhang, ChujieGao, YixinHuang, WenhanLyu, YixuanZhang, XinerLi, et al.. 2024. TrustLLM: Trustworthiness in large language models. arXiv:2401.05561. Retrieved from https://arxiv.org/abs/2401.05561

[262] Weiwei Sun, Zhengliang Shi, Shen Gao, Pengjie Ren, Maarten de Rijke, and Zhaochun Ren. 2022. Contrastive learning reduces hallucination in conversations. arXiv:2212.10400. Retrieved from https://arxiv.org/abs/2212.10400

[263] Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten de Rijke, and Zhaochun Ren. 2023. Learning to tokenize for generative retrieval. arXiv:2304.04171. Retrieved from https://arxiv.org/abs/2304.04171

[264] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive network: A successor to transformer for large language models. arXiv:2307.08621. Retrieved from https://arxiv.org/abs/2307.08621

[265] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, 8968–8975.

[266] Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2022. Recitation-augmented language models. arXiv:2210.01296. Retrieved from https://arxiv.org/abs/2210.01296

[267] Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. ViperGPT: Visual inference via Python execution for reasoning. arXiv:2303.08128. Retrieved from https://arxiv.org/abs/2303.08128

[268] Chenmien Tan, Ge Zhang, and Jie Fu. 2023. Massive editing for large language models via meta learning. arXiv:2311.04661. Retrieved from https://arxiv.org/abs/2311.04661

[269] Jiejun Tan, Zhicheng Dou, Wen Wang, Mang Wang, Weipeng Chen, and Ji-Rong Wen. 2024. HtmlRAG: HTML is better than plain text for modeling retrieved knowledge in RAG systems. arXiv:2411.02959. Retrieved from https://arxiv.org/abs/2411.02959

[270] Jiejun Tan, Zhicheng Dou, Yutao Zhu, Peidong Guo, Kun Fang, and Ji-Rong Wen. 2024. Small models, big insights: Leveraging slim proxy models to decide when and what to retrieve for LLMs. arXiv:2402.12052. Retrieved from https://arxiv.org/abs/2402.12052

[271] Juntao Tan, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Zelong Li, and Yongfeng Zhang. 2024. Towards LLM-RecSys alignment with textual ID learning. arXiv:2403.19021. Retrieved from https://arxiv.org/abs/2403.19021

[272] Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. 2024. Democratizing large language models via personalized parameter-efficient fine-tuning. arXiv:2402.04401. Retrieved from https://arxiv.org/abs/2402.04401

[273] Qiaoyu Tang, Jiawei Chen, Bowen Yu, Yaojie Lu, Cheng Fu, Haiyang Yu, Hongyu Lin, Fei Huang, Ben He, Xianpei Han, et al. 2024. Self-retrieval: Building an information retrieval system with one large language model. arXiv:2403.00801. Retrieved from https://arxiv.org/abs/2403.00801

[274] Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, and Le Sun. 2023. ToolAlpaca: Generalized tool learning for language models with 3000 simulated cases. arXiv:2306.05301. Retrieved from https://arxiv.org/abs/2306.05301

[275] Xiangru Tang, Anni Zou, Zhuosheng Zhang, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. 2023. MedAgents: Large language models as collaborators for zero-shot medical reasoning. arXiv:2311.10537. Retrieved from https://arxiv.org/abs/2311.10537

[276] Yubao Tang, Ruqing Zhang, Jiafeng Guo, Jiangui Chen, Zuowei Zhu, Shuaiqiang Wang, Dawei Yin, and Xueqi Cheng. 2023. Semantic-enhanced differentiable search index inspired by learning strategies. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*. ACM, 4904–4913. DOI: https://doi.org/10.1145/3580305.3599903

[277] Yubao Tang, Ruqing Zhang, Jiafeng Guo, and Maarten de Rijke. 2023. Recent advances in generative information retrieval. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP '23)*. ACM, 294–297. DOI: https://doi.org/10.1145/3624918.3629547

[278] Yubao Tang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, and Xueqi Cheng. 2024. Listwise generative retrieval models via a sequential learning process. *ACM Transactions on Information Systems* 42, 5 (2024), 1–31.

[279] Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Prakash Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. 2022. Transformer memory as a differentiable search index. In *Proceedings of the Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022 (NeurIPS '22)*, 21831–21843. Retrieved from http://papers.nips.cc/paper_files/paper/2022/hash/892840a6123b5ec99ebaab8be1530fba-Abstract-Conference.html

[280] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. LaMDA: Language models for dialog applications. arXiv:2201.08239. Retrieved from https://arxiv.org/abs/2201.08239

[281] James Thorne. 2022. Data-efficient autoregressive document retrieval for fact verification. arXiv:2211.09388. Retrieved from https://arxiv.org/abs/2211.09388

[282] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: A large-scale dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 809–819.

[283] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. arXiv:2302.13971. Retrieved from https://arxiv.org/abs/2302.13971

[284] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288. Retrieved from https://arxiv.org/abs/2307.09288

[285] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. arXiv:2212.10509. Retrieved from https://arxiv.org/abs/2212.10509

[286] Ravisri Valluri, Akash Kumar Mohankumar, Kushal Dave, Amit Singh, Jian Jiao, Manik Varma, and Gaurav Sinha. 2024. Scaling the vocabulary of non-autoregressive models for efficient generative retrieval. arXiv:2406.06739. Retrieved from https://arxiv.org/abs/2406.06739

[287] Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation (INLG '19)*. Association for Computational Linguistics, 355–368. DOI: https://doi.org/10.18653/V1/W19-8643

[288] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in Neural Information*

Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 5998–6008. Retrieved from https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[289] Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry W. Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc V. Le, et al. 2023. FreshLLMs: Refreshing large language models with search engine augmentation. arXiv:2310.03214. Retrieved from https://arxiv.org/abs/2310.03214

[290] David Wan, Mengwen Liu, Kathleen McKeown, Markus Dreyer, and Mohit Bansal. 2023. Faithfulness-aware decoding strategies for abstractive summarization. arXiv:2303.03278. Retrieved from https://arxiv.org/abs/2303.03278

[291] Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Jiayang Cheng, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. arXiv:2310.07521. Retrieved from https://arxiv.org/abs/2310.07521

[292] Danqing Wang, Kevin Yang, Hanlin Zhu, Xiaomeng Yang, Andrew Cohen, Lei Li, and Yuandong Tian. 2023. Learning personalized story evaluation. arXiv:2310.03304. Retrieved from https://arxiv.org/abs/2310.03304

[293] Hongru Wang, Minda Hu, Yang Deng, Rui Wang, Fei Mi, Weichao Wang, Yasheng Wang, Wai-Chung Kwan, Irwin King, and Kam-Fai Wong. 2023. Large language models as source planner for personalized knowledge-grounded dialogues. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 9556–9569. DOI: https://doi.org/10.18653/V1/2023.FINDINGS-EMNLP.641

[294] Hongru Wang, Boyang Xue, Baohang Zhou, Tianhua Zhang, Cunxiang Wang, Guanhua Chen, Huimin Wang, and Kam-fai Wong. 2024. Self-DC: When to retrieve and when to generate? Self divide-and-conquer for compositional unknown questions. arXiv:2402.13514. Retrieved from https://arxiv.org/abs/2402.13514

[295] Haoyu Wang, Tuo Zhao, and Jing Gao. 2024. BlendFilter: Advancing retrieval-augmented large language models via query generation blending and knowledge filtering. arXiv:2402.11129. Retrieved from https://arxiv.org/abs/2402.11129

[296] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. arXiv:2212.03533. Retrieved from https://arxiv.org/abs/2212.03533

[297] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2023. SimLM: Pre-training with representation bottleneck for dense passage retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL '23)*. Association for Computational Linguistics, 2244–2258. DOI: https://doi.org/10.18653/V1/2023.ACL-LONG.125

[298] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Large search model: Redefining search stack in the era of LLMs. *SIGIR Forum* 57, 2 (2023), 23:1–23:16. DOI: https://doi.org/10.1145/3642979.3643006

[299] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2023. A comprehensive survey of continual learning::2302 Theory, method and application. arXiv.00487. Retrieved from https://arxiv.org/abs/2302.00487

[300] Shuting Wang, Zhicheng Dou, Jing Yao, Yujia Zhou, and Ji-Rong Wen. 2023. Incorporating explicit subtopics in personalized search. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*. ACM, 3364–3374. DOI: https://doi.org/10.1145/3543507.3583488

[301] Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2023. Knowledge editing for large language models: A survey. arXiv:2310.16218. Retrieved from https://arxiv.org/abs/2310.16218

[302] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2023. Generative recommendation: Towards next-generation recommender paradigm. arXiv:2304.03516. Retrieved from https://arxiv.org/abs/2304.03516

[303] Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023. Orthogonal subspace learning for language model continual learning. arXiv:2310.14152. Retrieved from https://arxiv.org/abs/2310.14152

[304] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. arXiv:2203.11171. Retrieved from https://arxiv.org/abs/2203.11171

[305] Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Hao Sun, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, et al. 2022. A neural corpus indexer for document retrieval. arXiv: 2206.02743. Retrieved from https://arxiv.org/abs/2206.02743

[306] Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023. Self-knowledge guided retrieval augmentation for large language models. arXiv:2310.05002. Retrieved from https://arxiv.org/abs/2310.05002

[307] Yidan Wang, Zhaochun Ren, Weiwei Sun, Jiyuan Yang, Zhixiang Liang, Xin Chen, Ruobing Xie, Su Yan, Xu Zhang, Pengjie Ren, et al. 2024. Enhanced generative recommendation via content and collaboration integration. arXiv:2403.18480. Retrieved from https://arxiv.org/abs/2403.18480

[308] Zhiruo Wang, Zhoujun Cheng, Hao Zhu, Daniel Fried, and Graham Neubig. 2024. What are tools anyway? A survey from the language model perspective. arXiv:2403.15452. Retrieved from https://arxiv.org/abs/2403.15452

[309] Zihan Wang, Yujia Zhou, Yiteng Tu, and Zhicheng Dou. 2023. NOVO: Learnable and interpretable document identifiers for model-based IR. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*. ACM, 2656–2665. DOI: https://doi.org/10.1145/3583780.3614993

[310] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research* 2022 (2022). Retrieved from https://openreview.net/forum?id=yzkSU5zdwD

[311] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. arXiv:2201.11903. Retrieved from https://arxiv.org/abs/2201.11903

[312] Orion Weller, Marc Marone, Nathaniel Weir, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2023. "According to...": Prompting language models improves quoting from pre-training data. arXiv:2305.13252. Retrieved from https://arxiv.org/abs/2305.13252

[313] Ryen W. White. 2023. Tasks, copilots, and the future of search: A keynote at SIGIR 2023. *SIGIR Forum* 57, 2 (2023), 4:1–4:8. DOI: https://doi.org/10.1145/3642979.3642985

[314] Stanislaw Wozniak, Bartlomiej Koptyra, Arkadiusz Janz, Przemyslaw Kazienko, and Jan Kocon. 2024. Personalized large language models. arXiv:2402.09269. Retrieved from https://arxiv.org/abs/2402.09269

[315] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023. Visual ChatGPT: Talking, drawing and editing with visual foundation models. arXiv:2303.04671. Retrieved from https://arxiv.org/abs/2303.04671

[316] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas A. Funkhouser. 2023. TidyBot: Personalized robot assistance with large language models. *Autonomous Robots* 47, 8 (2023), 1087–1102. DOI: https://doi.org/10.1007/S10514-023-10139-Z

[317] Shiguang Wu, Wenda Wei, Mengqi Zhang, Zhumin Chen, Jun Ma, Zhaochun Ren, Maarten de Rijke, and Pengjie Ren. 2024. Generative retrieval as multi-vector dense retrieval. arXiv:2404.00684. Retrieved from https://arxiv.org/abs/2404.00684

[318] Tongtong Wu, Massimo Caccia, Zhuang Li, Yuan-Fang Li, Guilin Qi, and Gholamreza Haffari. 2022. Pretrained language model in continual learning: A comparative study. In *Proceedings of the 10th International Conference on Learning Representations (ICLR '22), Virtual Event*. OpenReview.net. Retrieved from https://openreview.net/forum?id=figzpGMrdD

[319] Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual learning for large language models: A survey. arXiv:2402.01364. Retrieved from https://arxiv.org/abs/2402.01364

[320] Yuwei Wu, Xuezhe Ma, and Diyi Yang. 2021. Personalized response generation via generative split memory network. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1956–1970.

[321] Yong Xie, Karan Aggarwal, and Aitzaz Ahmad. 2023. Efficient continual pre-training for building domain specific large language models. arXiv:2311.08545. Retrieved from https://arxiv.org/abs/2311.08545

[322] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[323] Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. RECOMP: Improving retrieval-augmented LMs with compression and selective augmentation. arXiv:2310.04408. Retrieved from https://arxiv.org/abs/2310.04408

[324] Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023. Search-in-the-Chain: Towards the accurate, credible and traceable content generation for complex knowledge-intensive tasks. arXiv:2304.14732. Retrieved from https://arxiv.org/abs/2304.14732

[325] Xuhai Xu, Bingshen Yao, Yuanzhe Dong, Hong Yu, James Hendler, Anind K. Dey, and Dakuo Wang. 2023. Leveraging large language models for mental health prediction via online text data. arXiv:2307.14385. Retrieved from https://arxiv.org/abs/2307.14385

[326] Zhichao Xu, Fengran Mo, Zhiqi Huang, Crystina Zhang, Puxuan Yu, Bei Wang, Jimmy Lin, and Vivek Srikumar. 2025. A survey of model architectures in information retrieval. arXiv:2502.14822. Retrieved from https://arxiv.org/abs/2502.14822

[327] Haoyan Yang, Zhitao Li, Yong Zhang, Jianzong Wang, Ning Cheng, Ming Li, and Jing Xiao. 2023. PRCA: Fitting black-box large language models for retrieval question answering via pluggable reward-driven contextual adapter. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, 5364–5375. Retrieved from https://aclanthology.org/2023.emnlp-main.326

[328] Tianchi Yang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, and Qi Zhang. 2023. Auto search indexer for end-to-end document retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics, 6955–6970. Retrieved from https://aclanthology.org/2023.findings-emnlp.464

[329] Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda R. Petzold, William Yang Wang, and Wei Cheng. 2023. A survey on detection of LLMs-generated content. arXiv:2310.15654. Retrieved from https://arxiv.org/abs/2310.15654

[330] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate problem solving with large language models. arXiv:2305.10601. Retrieved from https://arxiv.org/abs/2305.10601

[331] Shunyu Yao, Jeffrey Zhao, Dian Yu, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2022. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the NeurIPS 2022 Foundation Models for Decision Making Workshop*. Retrieved from https://openreview.net/forum?id=tvI4u1ylcqs

[332] Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, 10222–10240. DOI: https://doi.org/10.18653/V1/2023.EMNLP-MAIN.632

[333] Xi Ye, Ruoxi Sun, Sercan Ö. Arik, and Tomas Pfister. 2023. Effective large language model adaptation for improved grounding. arXiv:2311.09533. Retrieved from https://arxiv.org/abs/2311.09533

[334] Howard Yen, Tianyu Gao, and Danqi Chen. 2024. Long-context language modeling with parallel context encoding. arXiv:2402.16617. Retrieved from https://arxiv.org/abs/2402.16617

[335] Soyoung Yoon, Chaeeun Kim, Hyunji Lee, Joel Jang, and Minjoon Seo. 2023. Exploring the practicality of generative retrieval on dynamic corpora. Retrieved from https://api.semanticscholar.org/CorpusID:258967398

[336] Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. arXiv:2310.01558. Retrieved from https://arxiv.org/abs/2310.01558

[337] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. arXiv:2209.10063. Retrieved from https://arxiv.org/abs/2209.10063

[338] Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023. Augmentation-adapted retriever improves generalization of language models as generic plug-in. arXiv:2305.17331. Retrieved from https://arxiv.org/abs/2305.17331

[339] Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2022. SeqDiffuSeq: Text diffusion with encoder-decoder transformers. arXiv:2212.10325. Retrieved from https://arxiv.org/abs/2212.10325

[340] Peiwen Yuan, Xinglin Wang, Shaoxiong Feng, Boyuan Pan, Yiwei Li, Heda Wang, Xupeng Miao, and Kan Li. 2024. Generative Dense Retrieval: Memory can be a burden. arXiv:2401.10487. Retrieved from https://arxiv.org/abs/2401.10487

[341] Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation. In *Proceedings of the Advances in Neural Information Processing Systems 34*, 27263–27277.

[342] Hansi Zeng, Chen Luo, Bowen Jin, Sheikh Muhammad Sarwar, Tianxin Wei, and Hamed Zamani. 2023. Scalable and effective generative information retrieval. arXiv:2311.09134. Retrieved from https://arxiv.org/abs/2311.09134

[343] Hansi Zeng, Chen Luo, and Hamed Zamani. 2024. Planning ahead in generative retrieval: Guiding autoregressive generation through simultaneous decoding. arXiv:2404.14600. Retrieved from https://arxiv.org/abs/2404.14600

[344] Hailin Zhang, Yujing Wang, Qi Chen, Ruiheng Chang, Ting Zhang, Ziming Miao, Yingyan Hou, Yang Ding, Xupeng Miao, Haonan Wang, et al. 2023. Model-enhanced vector index. arXiv:2309.13335. Retrieved from https://arxiv.org/abs/2309.13335

[345] Jingqing Zhang, Kai Sun, Akshay Jagadeesh, Mahta Ghahfarokhi, Deepa Gupta, Ashok Gupta, Vibhor Gupta, and Yike Guo. 2023. The potential and pitfalls of using a large language model such as ChatGPT or GPT-4 as a clinical assistant. arXiv:2307.08152. Retrieved from https://arxiv.org/abs/2307.08152

[346] Kui Zhang, Guangquan Lu, Guixian Zhang, Zhi Lei, and Lijuan Wu. 2022. Personalized headline generation with enhanced user interest perception. In *Proceedings of the International Conference on Artificial Neural Networks*. Springer, 797–809.

[347] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3836–3847.

[348] Lingxi Zhang, Yue Yu, Kuan Wang, and Chao Zhang. 2024. ARL2: Aligning retrievers for black-box large language models via self-guided adaptive relevance labeling. arXiv:2402.13542. Retrieved from https://arxiv.org/abs/2402.13542

[349] Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. 2024. A comprehensive study of knowledge editing for large language models. arXiv:2401.01286. Retrieved from https://arxiv.org/abs/2401.01286

[350] Peitian Zhang, Zheng Liu, Yujia Zhou, Zhicheng Dou, Fangchao Liu, and Zhao Cao. 2024. Generative retrieval via term set generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*. Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (Eds.), ACM, 458–468.

[351] Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve anything to augment large language models. arXiv:2310.07554. Retrieved from https://arxiv.org/abs/2310.07554

[352] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? arXiv:1801.07243. Retrieved from https://arxiv.org/abs/1801.07243

[353] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating text generation with BERT. arXiv:1904.09675. Retrieved from https://arxiv.org/abs/1904.09675

[354] Wenhao Zhang, Mengqi Zhang, Shiguang Wu, Jiahuan Pei, Zhaochun Ren, Maarten de Rijke, Zhumin Chen, and Pengjie Ren. 2024. ExcluIR: Exclusionary neural information retrieval. arXiv:2404.17288. Retrieved from https://arxiv.org/abs/2404.17288

[355] Yidan Zhang, Ting Zhang, Dong Chen, Yujing Wang, Qi Chen, Xing Xie, Hao Sun, Weiwei Deng, Qi Zhang, Fan Yang, et al. 2023. IRGen: Generative modeling for image retrieval. arXiv:2303.10126. Retrieved from https://arxiv.org/abs/2303.10126

[356] Zhexin Zhang, Leqi Lei, Lindong Wu, Rui Sun, Yongkang Huang, Chong Long, Xiao Liu, Xuanyu Lei, Jie Tang, and Minlie Huang. 2023. SafetyBench: Evaluating the safety of large language models with multiple choice questions. arXiv:2309.07045. Retrieved from https://arxiv.org/abs/2309.07045

[357] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. arXiv:2303.18223. Retrieved from https://arxiv.org/abs/2303.18223

[358] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Adapting large language models by integrating collaborative semantics for recommendation. arXiv:2311.09049. Retrieved from https://arxiv.org/abs/2311.09049

[359] Aakas Zhiyuli, Yanfang Chen, Xuan Zhang, and Xun Liang. 2023. BookGPT: A general framework for book recommendation empowered by large language model. arXiv:2305.15673. Retrieved from https://arxiv.org/abs/2305.15673

[360] Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. arXiv:2210.07197. Retrieved from https://arxiv.org/abs/2210.07197

[361] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. LIMA: Less is more for alignment. In *Proceedings of the Advances in Neural Information Processing Systems 36*, 55006–55021.

[362] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Encoding history with context-aware representation learning for personalized search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), Virtual Event*. ACM, 1111–1120. DOI: https://doi.org/10.1145/3397271.3401175

[363] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2023. Enhancing generative retrieval with reinforcement learning from relevance feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP '23)*. Association for Computational Linguistics, 12481–12490. Retrieved from https://aclanthology.org/2023.emnlp-main.768

[364] Yujia Zhou, Yan Liu, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Zheng Liu, Chaozhuo Li, Zhicheng Dou, Tsung-Yi Ho, and Philip S. Yu. 2024. Trustworthiness in retrieval-augmented generation systems: A survey. arXiv:2409.10102. Retrieved from https://arxiv.org/abs/2409.10102

[365] Yujia Zhou, Zheng Liu, and Zhicheng Dou. 2024. Boosting the potential of large language models with an intelligent information assistant. In *Proceedings of the Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024 (NeurIPS '24)*. Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (Eds.). Retrieved from http://papers.nips.cc/paper_files/paper/2024/hash/28d38c036365420f61ce03300418e44a-Abstract-Conference.html

[366] Yujia Zhou, Zheng Liu, Jiajie Jin, Jian-Yun Nie, and Zhicheng Dou. 2024. Metacognitive retrieval-augmented large language models. arXiv:2402.11626. Retrieved from https://arxiv.org/abs/2402.11626

[367] Yujia Zhou, Jing Yao, Zhicheng Dou, Yiteng Tu, Ledell Wu, Tat-Seng Chua, and Ji-Rong Wen. 2024. ROGER: Ranking-oriented generative retrieval. *ACM Transactions on Information Systems* 42, 6 (2024), 155:1–155:25. DOI: https://doi.org/10.1145/3603167

[368] Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, and Ji-Rong Wen. 2023. DynamicRetriever: A pre-trained model-based IR system without an explicit index. *Machine Intelligence Research* 20, 2 (2023), 276–288. DOI: https://doi.org/10.1007/S11633-022-1373-9

[369] Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, Peitian Zhang, and Ji-Rong Wen. 2022. Ultron: An ultimate retriever on corpus with a model-based indexer. arXiv:2208.09257. Retrieved from https://arxiv.org/abs/2208.09257

[370] Yujia Zhou, Jing Yao, Ledell Wu, Zhicheng Dou, and Ji-Rong Wen. 2023. WebUltron: An ultimate retriever on webpages under the model-centric paradigm. *IEEE Transactions on Knowledge and Data Engineering* 36, 9 (2023), 4996–5006.

[371] Yujia Zhou, Qiannan Zhu, Jiajie Jin, and Zhicheng Dou. 2024. Cognitive personalized search integrating large language models with an efficient memory mechanism. arXiv:2402.10548. Retrieved from https://arxiv.org/abs/2402.10548
[372] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. arXiv:2308.07107. Retrieved from https://arxiv.org/abs/2308.07107
[373] Shengyao Zhuang, Houxing Ren, Linjun Shou, Jian Pei, Ming Gong, Guido Zuccon, and Daxin Jiang. 2022. Bridging the gap between indexing and retrieval for differentiable search index with query generation. arXiv:2206.10128. Retrieved from https://arxiv.org/abs/2206.10128
[374] Noah Ziems, Wenhao Yu, Zhihan Zhang, and Meng Jiang. 2023. Large language models are built-in autoregressive search engines. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics, 2666–2678. DOI: https://doi.org/10.18653/V1/2023.FINDINGS-ACL.167

Appendix

A Details for Evaluation

A.1 Evaluation Metrics for GR

Recall. Recall is a metric that measures the proportion of relevant documents retrieved by the search system. For a given cutoff point k, the recall Recall@k is defined as:

$$\text{Recall@}k = \frac{1}{|Q|}\sum_{q=1}^{|Q|}\frac{ret_{q,k}}{rel_q}, \tag{A1}$$

where $|Q|$ is the number of queries in the set, $ret_{q,k}$ is the number of relevant documents retrieved for the qth query within the top k results, and rel_q is the total number of relevant documents for the qth query.

R-Precision. R-Precision measures the precision at the rank position R, which corresponds to the number of relevant documents for a given query q. It is calculated as:

$$\text{R-Precision} = \frac{ret_{q,R}}{rel_q}, \tag{A2}$$

where $ret_{q,R}$ is the number of relevant documents retrieved within the top R positions, and R is equivalent to rel_q.

MRR. MRR reflects the average rank position of the first relevant document returned in the search results. It is computed as follows:

$$\text{MRR} = \frac{1}{|Q|}\sum_{q=1}^{|Q|}\frac{1}{\text{rank}_q}, \tag{A3}$$

where rank_q is the rank of the first relevant document returned for the qth query.

MAP. MAP calculates the average precision across multiple queries. It considers the exact position of all relevant documents and is calculated using the following formula:

$$\text{MAP} = \frac{1}{|Q|}\sum_{q=1}^{|Q|}\left(\frac{1}{rel_q}\sum_{k=1}^{n_q}\text{P@}k \times I(q,k)\right), \tag{A4}$$

where P@k is the precision at cutoff k, and $I(q,k)$ is an indicator function that is 1 if the document at position k is relevant to the q-th query and 0 otherwise.

nDCG. nDCG takes into account not only the relevance of the documents returned but also their positions in the result list. It is defined by:

$$\text{DCG@}k = \sum_{i=1}^{k}\frac{2^{\text{rel}_i}-1}{\log_2(i+1)}, \tag{A5}$$

$$\text{nDCG@}k = \frac{\text{DCG@}k}{\text{IDCG@}k}, \tag{A6}$$

where rel_i represents the graded relevance of the ith document, DCG@k is the discounted cumulative gain, and IDCG@k represents the maximum possible DCG@k.

A.2 Benchmarks for GR

MS MARCO. MS MARCO (Microsoft Machine Reading Comprehension) is a large-scale dataset developed by Microsoft for evaluating machine reading comprehension, retrieval, and QA capabilities within web search contexts. It comprises two primary benchmarks:

- *Document Ranking*: This benchmark includes approximately 3.2 million documents derived from real user queries extracted from Microsoft Bing's search logs. Each query is paired with annotated relevant documents, facilitating the evaluation of retrieval accuracy and scalability.
- *Passage Ranking*: Containing around 8.8 million passages, this benchmark focuses on more granular retrieval tasks, assessing the system's ability to identify relevant information at the passage level.

The diversity of question types and document genres in MS MARCO aims to mimic complex web search scenarios, making it a pivotal resource for testing the robustness and effectiveness of GR systems.

NQ. NQ is a QA dataset introduced by Google, utilizing Wikipedia as its foundational corpus. It encompasses approximately 3.2 million documents, each corresponding to a Wikipedia page. The dataset includes a wide array of natural user queries along with their respective answers extracted directly from web pages in Google search results. NQ is designed to evaluate the retrieval performance of GR systems in addressing real-world, information-seeking questions, emphasizing the ability to understand and retrieve precise answers from a vast knowledge base.

KILT. KILT is an extensive benchmark dataset that integrates five categories of knowledge-intensive tasks, including:

- *Fact Checking*: Utilizing datasets like FEVER, KILT assesses the system's ability to verify factual claims against a knowledge base.
- *Entity Linking*: Incorporates datasets such as AIDA CoNLL-YAGO, WNED-WIKI, and WNED-CWEB to evaluate the linking of entities mentioned in text to their corresponding entries in a knowledge base.
- *Slot Filling*: Includes T-REx and Zero Shot RE datasets to test the system's ability to populate predefined slots with relevant information extracted from the text.
- *Open-Domain QA*: Combines datasets like NQ, HotpotQA, TriviaQA, and ELI5 to evaluate the retrieval and comprehension capabilities of the system in answering open-ended questions.
- *Dialogue*: Utilizes the Wizard of Wikipedia dataset to assess the system's performance in maintaining informative and coherent dialogues based on retrieved knowledge.

KILT employs Wikipedia as its primary corpus, consisting of approximately 5.9 million wiki pages. The benchmark aims to evaluate the effectiveness of IR systems in handling complex language tasks that require extensive background knowledge and the ability to integrate information across multiple domains.

TREC Deep Learning Track 2019 and 2020. The TREC Deep Learning Tracks for 2019 and 2020 are specialized evaluation campaigns focusing on the application of deep learning techniques to enhance the efficiency and effectiveness of IR systems. The primary tasks in these tracks include:

— *Document Ranking*: Assessing the ability of retrieval systems to rank entire documents based on their relevance to a given query.
— *Passage Ranking*: Evaluating the system's capability to identify and rank specific passages within documents that are most relevant to the query.

Received 13 May 2024; revised 31 October 2024; accepted 28 February 2025