Review

Integrating Knowledge Retrieval with Generation: A Comprehensive Survey of RAG Models in NLP

Jeanie Genesis * and Frazier Keane

Posted Date: 4 April 2025

doi: 10.20944/preprints202504.0351.v1

Keywords: retrieval-augmented generation; natural language processing; information retrieval; text generation; knowledge integration; open-domain question answering; dialogue systems; generative models; ethics in AI; bias mitigation; multi-document reasoning; scalability

 doi:10.20944/preprints202504.0351.v1

Review

Integrating Knowledge Retrieval with Generation: A Comprehensive Survey of RAG Models in NLP

Jeanie Genesis * and Frazier Keane

Department of Computer Science, Lancaster University
* Correspondence: jeanie.genesis@lancaster.ac.uk

Abstract: Retrieval-Augmented Generation (RAG) models have emerged as a powerful paradigm in natural language processing (NLP), combining the strengths of information retrieval and text generation to enhance the quality and accuracy of generated responses. Recent advances in natural language processing have led to the development of Retrieval-Augmented Generation (RAG) models, a hybrid approach that combines the benefits of retrieval-based and generative models. Unlike traditional generative models that rely solely on pre-existing knowledge encoded within the model's parameters, RAG models leverage external knowledge sources, such as large-scale text corpora, to retrieve contextually relevant information to support the generation process. This ability to incorporate external information enhances the quality, relevance, and factual accuracy of the generated outputs, making RAG models particularly useful for tasks such as open-domain question answering, document summarization, dialogue generation, and specialized domains like legal and medical applications. In this survey, we provide a detailed exploration of RAG models, beginning with a comprehensive review of their underlying architecture. We describe the integration of retrieval mechanisms, such as sparse and dense retrieval, with large-scale pre-trained generative models, highlighting the process through which retrieved knowledge is utilized to guide and enrich the generation process. We also examine the different techniques employed to fuse retrieved information with the generative model, such as attention mechanisms, concatenation methods, and hybrid approaches. The survey further explores the diverse applications of RAG models, demonstrating their effectiveness across various NLP tasks and domains. Despite the success of RAG models, we identify and discuss several critical challenges that must be addressed for further advancement. These challenges include improving the quality and relevance of the retrieved documents, resolving issues of conflicting or ambiguous information between retrieval and generation components, enhancing model scalability to handle large corpora in real-time, and mitigating ethical concerns related to bias, fairness, and the potential generation of misinformation. We also explore the impact of these challenges on the real-world deployment of RAG systems, particularly in sensitive applications such as healthcare, law, and customer service. We provide an in-depth discussion of the current state-of-the-art techniques employed to address these challenges, including hybrid retrieval methods, novel strategies for knowledge integration, and advancements in model efficiency and scalability. Furthermore, we explore ethical considerations, such as the risk of bias in retrieved information and generated content, and propose methods for ensuring fairness and transparency in RAG systems. Additionally, we examine the need for new evaluation metrics that better capture the performance of RAG models in practical settings, where both retrieval quality and generative coherence are critical. Finally, the survey concludes by outlining promising future research directions aimed at advancing RAG models. These directions include the development of more sophisticated retrieval mechanisms, such as context-aware retrieval, the integration of structured knowledge sources like knowledge graphs, and the design of more robust and interpretable generative architectures. We also highlight the importance of addressing ethical concerns, such as improving bias mitigation techniques and enhancing the transparency of generative processes, to ensure that RAG systems can be deployed responsibly in a wide range of applications. This survey aims to serve as a comprehensive guide for researchers and practitioners seeking to understand the current landscape of Retrieval-Augmented Generation models and provides insights into the future evolution of this exciting area in NLP.

 doi:10.20944/preprints202504.0351.v1

Keywords: retrieval-augmented generation; natural language processing; information retrieval; text generation; knowledge integration; open-domain question answering; dialogue systems; generative models; ethics in AI; bias mitigation; multi-document reasoning; scalability

1. Introduction

Recent advances in natural language processing (NLP) have been largely driven by the development and application of large language models (LLMs), such as GPT-3, BERT , and T5 . These models have demonstrated remarkable capabilities across a wide range of tasks, including text generation, machine translation, question answering, summarization, and more. However, despite their impressive performance, LLMs often struggle with certain limitations, particularly when it comes to handling tasks that require extensive domain knowledge, reasoning, or the ability to recall specific information that may not have been encoded during training. This issue arises because the knowledge contained in these models is frozen at the point of training and cannot be updated without retraining the entire model, which is computationally expensive and time-consuming [1]. To address these challenges, a promising research direction has emerged known as Retrieval-Augmented Generation (RAG) [2]. RAG combines the power of LLMs with external knowledge sources through retrieval mechanisms, enabling models to access real-time information from a large external corpus to complement their internal knowledge. This hybrid approach significantly enhances the performance of language models by providing them with dynamic access to information beyond their fixed parameters. The basic idea behind RAG is to retrieve relevant documents from an external knowledge base or corpus, which are then used to augment the input to the language model, allowing the model to generate more informed, accurate, and contextually appropriate responses. Retrieval-augmented approaches can be broadly classified into two categories: those that rely on an explicit retrieval process to fetch external documents before generating the output and those that integrate retrieval into the generation process itself, often using attention mechanisms to incorporate retrieved information in real time. These approaches typically leverage powerful search engines or databases, such as dense retrievers based on models like DPR, to find documents or passages relevant to the input query [3]. These retrieved documents are then fed into the LLM, either as additional context or through more sophisticated methods like fusion-in-decoder (FiD) , to guide the model's generation process. Incorporating retrieval into LLMs allows them to overcome several inherent limitations of pre-trained models. First, it reduces the need for models to memorize vast amounts of knowledge, which can lead to inefficiencies and performance degradation over time. Instead, the model can rely on an external knowledge source that is continually updated, providing more accurate and timely information [4]. This is particularly useful in domains such as scientific research, healthcare, and technology, where the corpus of knowledge is constantly evolving. Second, RAG enhances the efficiency of LLMs by focusing the model's attention on the most relevant portions of knowledge during the generation process [5]. This selective access to relevant documents not only reduces the computational burden on the model but also leads to better performance in terms of both accuracy and relevance [6]. For instance, retrieval-augmented models can perform well in open-domain question answering tasks by dynamically selecting the most pertinent information from a wide array of possible documents, thus generating more precise answers compared to a traditional LLM relying solely on its fixed training data. Moreover, the integration of retrieval can help mitigate some of the biases and errors inherent in large pre-trained models. Since the external knowledge sources are typically curated and can be continuously updated, it is possible to introduce mechanisms that check for factual correctness, update out-of-date information, and even counteract harmful biases by drawing from more diverse and representative datasets [7]. This opens up new avenues for the deployment of LLMs in sensitive domains like healthcare, law, and education, where accuracy and fairness are critical [8]. This survey provides a comprehensive review of the literature surrounding Retrieval-Augmented Generation (RAG) for large language models. We begin

 doi:10.20944/preprints202504.0351.v1

by discussing the motivation behind combining retrieval and generation, exploring how RAG builds upon prior research in neural retrieval and language modeling [9]. We then provide an overview of the key techniques used in RAG, including both traditional methods and modern advancements in neural retrieval architectures, as well as the various approaches for integrating retrieved information into the generation process [10]. Next, we examine the diverse range of applications where RAG has shown promising results, such as knowledge-intensive tasks, question answering, summarization, and dialogue systems. We also explore the various challenges associated with implementing RAG, including issues related to the efficiency of the retrieval process, the complexity of integration strategies, and the impact of retrieval on the quality and coherence of the generated text. Finally, we outline the future directions of RAG research, highlighting areas for further exploration and potential improvements [11]. These include advancements in retrieval techniques, the development of more sophisticated models that combine retrieval and generation in novel ways, and the exploration of real-world use cases where RAG could offer significant benefits over traditional LLMs. By providing this survey, we aim to give both researchers and practitioners a comprehensive understanding of the current state of Retrieval-Augmented Generation and its potential to drive the next wave of innovations in natural language processing [12]. **Contributions of this Survey:** The primary contributions of this survey are as follows:

- A detailed review of the principles and motivations behind retrieval-augmented generation for large language models.
- An in-depth discussion of key techniques for retrieval and generation, including both traditional and cutting-edge methods.
- A thorough examination of the state-of-the-art applications of RAG in various domains, such as question answering, summarization, and dialogue generation.
- A critical analysis of the challenges and limitations associated with RAG, including retrieval efficiency, document quality, and coherence of generated responses.
- An exploration of future research directions and potential applications of RAG, with an emphasis on areas where further improvements can be made to enhance model performance [13].

The remainder of this survey is organized as follows: In Section 2, we provide an overview of the foundational concepts and related work in neural retrieval and language modeling. In Section 3, we delve into the core techniques used in retrieval-augmented generation. Section 4 discusses the diverse applications of RAG in NLP tasks [14]. In Section5 , we address the various challenges and open problems in the field [15]. Finally, in Section 6, we explore the future directions of RAG research, followed by concluding remarks in Section 6.

2. Background and Related Work

In this section, we provide a detailed overview of the foundational concepts underlying Retrieval-Augmented Generation (RAG) models, including the core techniques and methods from both retrieval-based information retrieval systems and generative language models. We also survey related work in these areas, highlighting the key advancements that have led to the development of RAG approaches. The discussion covers traditional retrieval systems, the advent of neural retrieval methods, and recent developments in generative models, with a focus on their integration.

2.1. Retrieval Techniques for Information Retrieval

Information retrieval (IR) is the task of finding relevant documents from a large corpus based on a user's query. The goal of traditional IR systems is to rank documents by their relevance to the query, enabling users to retrieve the most pertinent information efficiently. Traditional retrieval models, such as the vector space model (VSM) and the probabilistic retrieval model , rely on keyword matching and document scoring techniques based on term frequency and inverse document frequency (TF-IDF). However, with the rise of deep learning, these classical approaches were soon surpassed by more powerful neural methods that better capture semantic relationships between queries and documents

 doi:10.20944/preprints202504.0351.v1

[16]. Neural retrieval models use embeddings to represent both queries and documents in a continuous vector space, enabling the retrieval of documents based on semantic similarity rather than simple keyword matching.

2.1.1. Dense Retrieval Models

A significant breakthrough in retrieval systems was the introduction of dense retrieval models, which use pre-trained neural networks to map both queries and documents into fixed-length embeddings [17]. These models significantly improve upon traditional methods by capturing the semantic meaning of queries and documents, rather than relying solely on surface-level keyword matching. One of the most popular dense retrieval techniques is based on the use of bi-encoders, where separate neural networks are used to encode both the query and the document into a shared vector space. Once the embeddings are computed, the retrieval process involves computing the similarity between the query and all document embeddings in the corpus, typically using dot-product or cosine similarity. Formally, given a query q and a document d, the goal of a dense retrieval model is to learn a function f_q and f_d that map the query and document to an embedding space:

$$q' = f_q(q), \quad d' = f_d(d)$$

where $q' \in \mathbb{R}^n$ and $d' \in \mathbb{R}^n$ are the embeddings of the query and document, respectively, and n is the dimensionality of the embedding space. The relevance score between the query and document is computed as the similarity between their embeddings:

$$\text{Sim}(q', d') = \frac{q' \cdot d'}{\|q'\| \|d'\|}$$

This allows dense retrieval models to retrieve documents that are semantically similar to the query, even when they do not share exact keywords [18]. Recent innovations in dense retrieval include models such as the Dense Passage Retriever (DPR), which uses a pre-trained BERT-based encoder to generate dense representations for both queries and documents [19]. The retrieved documents can then be passed to a generative model for further processing.

2.1.2. Sparse Retrieval Models

In contrast to dense retrieval, sparse retrieval methods rely on traditional keyword-based retrieval, where the documents are represented as sparse vectors based on word counts or term frequencies. TF-IDF is one of the most commonly used sparse retrieval techniques, but more advanced methods such as BM25 have also been developed, which incorporate term frequency, document length, and inverse document frequency to improve retrieval performance. While sparse retrieval methods have proven effective for tasks like information retrieval and document ranking, they often struggle with understanding the deeper semantic meaning of queries and documents. This limits their applicability in tasks that require reasoning over complex queries or the retrieval of contextually relevant information.

2.2. Generative Models for Text Generation

Large language models (LLMs) are pre-trained on vast amounts of text and can generate coherent and contextually appropriate responses to a wide variety of inputs. These models are typically trained using unsupervised learning techniques, where the objective is to predict the next word in a sequence of text. The architecture of LLMs, especially those based on the Transformer , has revolutionized NLP due to its scalability and parallelizability, as well as its ability to capture long-range dependencies in text.

2.2.1. Autoregressive Models

Autoregressive language models, such as GPT-3 and GPT-2 , are trained to predict the next token in a sequence given the previous tokens. The model generates text by iteratively sampling the next

 doi:10.20944/preprints202504.0351.v1

token, conditioned on the previous tokens. This process allows for the generation of highly coherent text across a wide range of domains [20]. The objective function for autoregressive models can be expressed as:

$$\mathcal{L}_{\text{AR}} = -\sum_{t=1}^{T} \log P(x_t | x_1, \ldots, x_{t-1})$$

where x_t denotes the t-th token in the sequence, and T is the length of the sequence. Autoregressive models are highly effective for many generative tasks but have the limitation of being bound by their training data and the model's inherent knowledge, which can be outdated or incomplete.

2.2.2. Encoder-Decoder Models

Encoder-decoder models, such as BART and T5 , operate by first encoding the input sequence into a fixed-length representation and then decoding that representation into a target sequence [21]. These models are particularly effective for tasks like machine translation, summarization, and text generation. The objective function for encoder-decoder models can be written as:

$$\mathcal{L}_{\text{ED}} = -\sum_{t=1}^{T} \log P(y_t | y_1, \ldots, y_{t-1}, \mathbf{h})$$

where y_t is the t-th token in the target sequence, and $\mathbf{h}$ is the hidden representation of the input sequence generated by the encoder.

2.2.3. Challenges with Generative Models

Despite their impressive capabilities, LLMs suffer from several limitations. One of the most significant issues is the knowledge cutoff—LLMs only have access to information that was available up to the point of their training, meaning they are unable to incorporate more recent knowledge without retraining [22]. Furthermore, LLMs are not always able to generate text that is factually accurate or relevant, particularly when handling complex or domain-specific tasks.

2.3. Retrieval-Augmented Generation (RAG) Models

The limitations of traditional generative models, such as the inability to access external, up-to-date knowledge, have led to the development of retrieval-augmented generation models. RAG combines retrieval and generation by integrating a retrieval component with a generative language model, enabling the model to dynamically access external knowledge during the generation process [23]. The general framework for RAG is depicted in Figure **??**. Given an input query or prompt, the system first retrieves a set of relevant documents from an external knowledge base. These retrieved documents are then used as additional context for the generative model, guiding its output generation.

2.3.1. Retrieval-Augmented Generation Framework

In a typical RAG setup, the first step is the retrieval of relevant documents from an external corpus [24]. Let the input query be denoted by q. The retrieval component produces a set of candidate documents $\{d_1, d_2, \ldots, d_K\}$, where K is the number of retrieved documents. These documents are typically retrieved using a dense retrieval model, such as DPR, based on the similarity between the query q and the document embeddings. Once the documents are retrieved, they are passed as context to a generative model. In the context of RAG, the generative model takes the form of an encoder-decoder architecture, where the encoder processes both the input query q and the retrieved documents $\{d_1, d_2, \ldots, d_K\}$ [25]. The decoder then generates the output sequence $y = \{y_1, y_2, \ldots, y_T\}$ conditioned on both the query and the documents [26]. Formally, the generation process in RAG can be written as:

$$y = \text{Decoder}(\text{Encoder}(q, \{d_1, d_2, \ldots, d_K\}))$$

 doi:10.20944/preprints202504.0351.v1

The key benefit of this approach is that the generative model has access to external information, which can significantly improve the quality and relevance of the generated output.

2.4. Related Work in Retrieval-Augmented Generation

Several studies have explored the integration of retrieval mechanisms with generative models [27]. One of the earliest works in this direction is the RAG model proposed by Lewis et al.. This model uses a retriever to fetch relevant documents and a generator to produce a response conditioned on the retrieved documents [28]. The authors show that this approach outperforms traditional generation models on tasks like open-domain question answering [29]. Another significant development is the use of Fusion-in-Decoder (FiD) , which integrates multiple retrieved documents in the decoder to improve generation performance. FiD has been shown to outperform other RAG variants in several benchmark tasks, including question answering and summarization [30]. Other notable approaches, such as REALM, GTR, and T5+retriever , have also demonstrated the power of combining retrieval and generation to improve performance across various NLP tasks. In summary, RAG represents a promising fusion of retrieval and generation, which allows large language models to leverage external knowledge effectively. In the next section, we explore the key techniques used in RAG models in greater detail.

3. Retrieval-Augmented Generation: Techniques and Architectures

In this section, we discuss the core techniques and architectures used in Retrieval-Augmented Generation (RAG) models. We focus on the key components of RAG, including the retrieval mechanism, the generative model, and how these components are integrated to produce high-quality outputs. We also cover the different strategies for retrieval and document selection, as well as the various approaches to combining retrieved information with the generation process [31]. The section concludes with a discussion of recent advancements in the field.

3.1. Retrieval Mechanisms in RAG Models

The first essential component of any RAG system is the retrieval mechanism, which is responsible for selecting relevant documents from a large corpus based on a given query. Effective retrieval is crucial for the performance of RAG models because the quality of the retrieved documents directly impacts the quality of the generated output. In this section, we discuss two main types of retrieval methods: sparse retrieval and dense retrieval.

3.1.1. Sparse Retrieval Methods

Sparse retrieval techniques, such as TF-IDF and BM25, have long been the backbone of traditional information retrieval systems [32]. These models treat the documents as bags of words and rely on keyword matching to identify relevant documents [33]. Despite their simplicity, sparse retrieval methods are still widely used, particularly in scenarios where the corpus is relatively static, and exact keyword matches are sufficient for retrieving relevant information [34]. Formally, given a query q and a document d, the TF-IDF score is calculated as:

$$\text{TF-IDF}(q, d) = \text{TF}(q, d) \times \text{IDF}(q)$$

where $\text{TF}(q, d)$ is the term frequency of the query terms in document d, and $\text{IDF}(q)$ is the inverse document frequency of the query term across the entire corpus. The BM25 model extends TF-IDF by incorporating a saturation function for term frequency and document length normalization, yielding:

$$\text{BM25}(q, d) = \sum_{i=1}^{|q|} \text{IDF}(q_i) \times \frac{\text{TF}(q_i, d) \times (k_1 + 1)}{\text{TF}(q_i, d) + k_1 \times (1 - b + b \times \frac{|d|}{\text{avg\_len}})}$$

 doi:10.20944/preprints202504.0351.v1

where k_1 and b are tunable parameters, and avg_len is the average document length in the corpus. These models are effective for basic retrieval tasks but can struggle with more complex queries that require semantic understanding.

3.1.2. Dense Retrieval Methods

Dense retrieval methods, on the other hand, rely on neural networks to embed both queries and documents into a continuous vector space. These models aim to learn embeddings that capture the semantic meaning of the query and the document, rather than relying on exact keyword matches [35]. One of the most popular dense retrieval models is the Dense Passage Retriever (DPR), which uses a bi-encoder architecture to independently encode queries and documents into fixed-length vectors. The relevance score between a query q and a document d is then computed using cosine similarity or dot product between their embeddings:

$$\text{Sim}(q', d') = \frac{q' \cdot d'}{\|q'\| \|d'\|}$$

where q' and d' are the dense embeddings of the query and document, respectively [36]. Dense retrieval allows for semantic matching, enabling the retrieval of documents that are contextually relevant to the query, even if they do not share exact terms. Training a dense retrieval model typically involves supervised learning, where pairs of queries and relevant documents are used to fine-tune a neural network to generate embeddings that maximize the similarity of relevant document-query pairs while minimizing the similarity of irrelevant pairs [37]. Recent advancements in dense retrieval include models like ANTIQUE and ColBERT, which incorporate efficient techniques for fast retrieval and large-scale indexing [38].

3.2. Integrating Retrieval with Generation

Once relevant documents are retrieved, they must be effectively integrated into the generative process. This is where RAG models shine, as they merge the capabilities of retrieval and generation. In this section, we discuss several strategies for integrating retrieved documents into the generation process [39].

3.2.1. Simple Concatenation

The most straightforward approach for integrating retrieval and generation is to concatenate the retrieved documents with the input query and use this concatenated sequence as input to the generative model. This approach does not involve any sophisticated handling of the retrieved documents and treats them as additional context for the generator [40]. While this method is easy to implement, it does not take full advantage of the retrieved information, as the generative model may struggle to select the most relevant parts of the retrieved documents. Formally, let $d_1, d_2, \ldots, d_K$ be the K retrieved documents, and let q be the query [41]. The input to the generative model can be represented as:

$$\text{Input} = \text{Concat}(q, d_1, d_2, \ldots, d_K)$$

where Concat denotes the concatenation operation [42]. The generative model then produces the output sequence y based on this input.

3.2.2. Fusion-in-Decoder (FiD)

The Fusion-in-Decoder (FiD) model improves upon simple concatenation by combining the retrieved documents in the decoder, rather than at the input. This approach involves processing each retrieved document separately in the encoder, but then merging the embeddings of all documents within the decoder to allow the model to attend to multiple pieces of relevant information during generation. This method allows for more fine-grained control over which pieces of the retrieved information are used in the output generation. Formally, FiD can be expressed as follows. Given the

 doi:10.20944/preprints202504.0351.v1

query q and K retrieved documents $\{d_1, d_2, \ldots, d_K\}$, the input to the encoder is the query q and each document d_i:

$$\mathbf{h}_q = \text{Encoder}(q), \quad \mathbf{h}_{d_i} = \text{Encoder}(d_i), \quad i \in \{1, 2, \ldots, K\}$$

The decoder then generates the output y based on the query and all document embeddings:

$$y = \text{Decoder}(\mathbf{h}_q, \{\mathbf{h}_{d_1}, \mathbf{h}_{d_2}, \ldots, \mathbf{h}_{d_K}\})$$

This method has been shown to significantly improve generation performance, especially for complex tasks where multiple pieces of information from different documents need to be synthesized.

3.2.3. Retrieval-Augmented Generation with Attention Mechanisms

Another approach for integrating retrieval and generation is to incorporate attention mechanisms that allow the model to dynamically attend to different parts of the retrieved documents during the generation process. The attention mechanism enables the model to focus on the most relevant parts of the documents based on the query and the ongoing context in the generation process. In the context of RAG, attention-based methods allow the model to weigh the importance of different documents and passages during generation. The attention over the retrieved documents can be expressed as:

$$\alpha_i = \text{Attention}(q, d_i), \quad i \in \{1, 2, \ldots, K\}$$

where α_i represents the attention weight for the i-th document [43]. The weighted sum of the retrieved document embeddings is then used as input to the decoder to generate the final output:

$$y = \text{Decoder}(q, \sum_{i=1}^{K} \alpha_i \cdot d_i)$$

This approach ensures that the model dynamically selects the most relevant information from the retrieved documents based on both the query and the ongoing generation context [44].

3.3. Advancements in Retrieval-Augmented Generation

Recent advancements in RAG research have led to several promising directions for improving both retrieval and generation components [45]. One key area of progress is in the development of more efficient retrieval techniques that scale to large corpora and allow for real-time retrieval of relevant documents. Models like ColBERT and FAISS have introduced optimizations for large-scale dense retrieval that significantly speed up the retrieval process while maintaining high accuracy. Another promising direction is the use of multi-hop retrieval, where the system first retrieves a set of documents based on the query and then performs a second round of retrieval on the retrieved documents to find even more relevant information [46]. This approach has been shown to improve performance on complex tasks that require reasoning over multiple pieces of information [47]. Additionally, some recent RAG models have introduced techniques for improving the quality of the generated text, such as the use of fine-grained control over the attention mechanism, hierarchical generation, and multi-task learning to allow the model to handle multiple types of tasks simultaneously. In the following section, we explore the applications of RAG models in various natural language processing tasks, including question answering, summarization, and dialogue systems.

4. Applications of Retrieval-Augmented Generation

In this section, we explore the various applications of Retrieval-Augmented Generation (RAG) models in natural language processing (NLP). We focus on how RAG frameworks have been successfully applied to tasks such as open-domain question answering, text summarization, and dialogue systems. Additionally, we discuss the effectiveness of RAG models in specialized domains like medical

 doi:10.20944/preprints202504.0351.v1

or legal document processing, where access to up-to-date external knowledge plays a crucial role in improving task performance.

4.1. Open-Domain Question Answering

Open-domain question answering (QA) is one of the most well-known applications of RAG models. Traditional QA systems typically rely on a fixed set of documents or a predefined knowledge base, and the answers are extracted directly from these sources [48]. However, in open-domain QA, the system must be capable of answering questions based on an expansive and dynamic collection of external information, which may not be present in the training data. This is where RAG models shine, as they can retrieve the most relevant documents from a vast corpus and use that information to generate more accurate and contextually relevant answers [49].

4.1.1. RAG for Open-Domain QA

In the open-domain QA setup, the input is a question q, and the goal is to generate a natural language answer a based on external knowledge retrieved from a large corpus [50]. The process typically involves two steps:

1. **Retrieval:** Given a question q, a retrieval mechanism (usually a dense retrieval model like DPR) retrieves a set of relevant documents $\{d_1, d_2, \ldots, d_K\}$ from the corpus [51].
2. **Generation:** The retrieved documents are then passed to a generative model, which generates an answer a based on the question and the retrieved documents.

Formally, the process can be written as:

$$a = \text{Generator}(q, \{d_1, d_2, \ldots, d_K\})$$

Here, a is the generated answer, and the model generates it by conditioning on the question and the set of retrieved documents [52]. This approach allows the system to answer questions that go beyond the information contained in the model's parameters, effectively giving the model access to external, real-time knowledge. One key advantage of RAG in open-domain QA is that it allows models to handle questions about recent events or niche topics that may not be present in the model's training data. For example, in the case of medical or scientific questions, the retrieval component can access the latest research papers or medical guidelines, ensuring that the generated answers are both up-to-date and factually accurate [53].

4.1.2. Benchmarks and Performance

Several benchmarks have been used to evaluate RAG-based open-domain QA systems, including the Natural Questions (NQ) dataset and the TriviaQA dataset [54]. These datasets contain real-world, open-domain questions, which are challenging due to their complexity and variability [55]. RAG-based models have been shown to outperform traditional methods, including both sparse retrieval systems and standalone generative models, by providing more accurate and contextually grounded answers. The RAG framework allows for improved performance on the recall of relevant documents, which is critical for answering complex questions that require detailed information from multiple sources.

4.2. Text Summarization

Text summarization is another domain where RAG models have demonstrated significant potential [56]. Summarization can be classified into two categories: extractive summarization, where relevant portions of the text are selected and concatenated to form a summary, and abstractive summarization, where the model generates a new summary that may include novel phrasing or rewording of the original text.

 doi:10.20944/preprints202504.0351.v1

4.2.1. RAG for Abstractive Summarization

In the context of abstractive summarization, RAG models retrieve relevant documents or passages from a large corpus to improve the quality of generated summaries. The retrieved documents can provide additional context, helping the model generate summaries that are more comprehensive and factually accurate. For instance, when summarizing news articles or research papers, RAG systems can retrieve supporting information, such as background details or explanations, which can be woven into the generated summary. Formally, given an input document D, the system retrieves a set of relevant passages $\{p_1, p_2, \ldots, p_K\}$ and generates a summary S as:

$$S = \text{Generator}(D, \{p_1, p_2, \ldots, p_K\})$$

By conditioning on both the original document and the retrieved passages, the generative model is able to produce summaries that are more informative and coherent, especially for long or complex documents.

4.2.2. RAG for Extractive Summarization

While extractive summarization focuses on selecting the most relevant portions of text, RAG models can enhance extractive summarization systems by retrieving additional relevant content before performing the extraction [57]. This additional context can help the model identify more important or nuanced segments to include in the final summary. However, the primary task of extractive summarization—selecting relevant spans—remains the same, with retrieval serving as an augmentative mechanism to improve selection. In this setup, the retrieval component selects relevant passages, and the model uses these passages to identify key spans from the original text to form the summary:

$$S_{\text{extractive}} = \text{Extractor}(D, \{p_1, p_2, \ldots, p_K\})$$

This approach has been shown to produce more accurate and informative extractive summaries compared to non-retrieval-based methods.

4.2.3. Evaluating Summarization Systems

RAG-based summarization systems are typically evaluated using standard summarization metrics such as ROUGE , which measures the overlap between generated summaries and reference summaries. These models have demonstrated improvements over traditional summarization systems, particularly in terms of content coverage and coherence [58]. The ability to retrieve relevant external knowledge and incorporate it into the generated summary enables RAG models to provide more complete and contextually relevant summaries.

4.3. Dialogue Systems

Dialogue systems, which involve human-computer interactions in the form of text or speech, have also benefited from the integration of retrieval mechanisms in RAG frameworks. In task-oriented dialogue systems, RAG models can be used to retrieve relevant information from a knowledge base to support goal-directed conversations, such as booking tickets or providing product recommendations [59]. In open-domain dialogue systems, retrieval-augmented generation allows for more natural and informative conversations by providing access to external knowledge [60].

4.3.1. RAG for Dialogue Generation

In the context of open-domain dialogue systems, RAG models allow for the generation of more diverse and informative responses by retrieving contextually relevant information from external sources [61]. The dialogue system retrieves relevant documents or past conversations that inform the current dialogue, providing the generative model with additional context to produce a more relevant and personalized response [62]. This is especially useful in conversations involving questions about

 doi:10.20944/preprints202504.0351.v1

factual information, such as trivia, recent events, or specific domain knowledge. Formally, given a dialogue history H and a user query q, the system retrieves relevant information $\{d_1, d_2, \ldots, d_K\}$ and generates a response r as:

$$r = \text{Generator}(H, q, \{d_1, d_2, \ldots, d_K\})$$

By retrieving contextually relevant documents and using them as additional input for the generative model, RAG-based dialogue systems are able to produce more informative and contextually appropriate responses [63].

4.3.2. Challenges in Dialogue Systems

Despite the promising results of RAG models in dialogue systems, several challenges remain. One of the main challenges is ensuring the relevance and appropriateness of the retrieved information, especially when the retrieval system may pull in irrelevant or outdated content. Additionally, while RAG models can retrieve useful information, they still need to handle the complexities of conversation, including maintaining coherence across multiple turns and managing diverse conversational topics [64]. Research in multi-turn dialogue systems and long-term memory retrieval is ongoing to address these challenges.

4.4. Specialized Domains: Medical and Legal Applications

RAG models are particularly beneficial in specialized domains, such as medical and legal fields, where up-to-date, accurate, and domain-specific knowledge is essential for providing high-quality responses. In these domains, RAG frameworks enable models to retrieve relevant and authoritative documents—such as medical guidelines, research papers, or legal statutes—and generate responses based on the most current and reliable information [65].

4.4.1. RAG in Medical Applications

In medical applications, RAG models can be used to answer complex clinical questions, assist with diagnosis, or provide treatment recommendations based on the latest medical research. By retrieving relevant papers, clinical guidelines, and research articles, RAG systems can support healthcare professionals with evidence-based information, enhancing decision-making in real-time [66].

4.4.2. RAG in Legal Applications

Similarly, in the legal domain, RAG models can be used to retrieve relevant case law, statutes, and legal precedents to assist in legal research or decision-making. By incorporating this external knowledge, legal professionals can make more informed decisions based on the latest legal frameworks, rulings, and interpretations [67]. In both of these specialized domains, the integration of retrieval and generation provides a unique advantage by enabling RAG systems to access and synthesize vast amounts of domain-specific knowledge to generate accurate, timely, and contextually appropriate responses [68].

4.5. Conclusion

In this section, we have explored the wide-ranging applications of RAG models in various domains of natural language processing, including open-domain question answering, text summarization, dialogue systems, and specialized domains such as medical and legal applications. The ability to combine retrieval with generation allows RAG models to enhance the accuracy and relevance of the generated content by providing access to external, real-time knowledge. As the field continues to evolve, the potential applications of RAG are vast, and these systems hold great promise for improving performance across many NLP tasks.

 doi:10.20944/preprints202504.0351.v1

5. Challenges and Open Issues in Retrieval-Augmented Generation

Despite the significant advances made in Retrieval-Augmented Generation (RAG) models, there are still several challenges and open issues that need to be addressed in order to fully unlock the potential of these systems. In this section, we examine some of the key challenges faced by RAG models, including issues related to retrieval quality, the integration of retrieved information with the generation process, scalability, and ethical concerns. We also discuss some ongoing research directions that may help mitigate these challenges.

5.1. Challenges in Retrieval Quality

One of the primary challenges in RAG systems is ensuring the quality of the retrieved documents. Since RAG models rely on external retrieval mechanisms to gather relevant context for generating responses, the accuracy and relevance of the retrieved documents directly impact the overall performance of the system. Several issues can arise in this context:

5.1.1. Irrelevant or Noisy Documents

A common challenge in retrieval-based systems is the retrieval of irrelevant or noisy documents that do not contribute meaningfully to the query or the task at hand. These documents may introduce ambiguity, bias, or errors into the generative process, leading to less accurate or nonsensical output [69]. For example, in open-domain question answering, a model might retrieve outdated information or documents that are semantically similar but not directly relevant to the question. This issue is particularly pronounced in large-scale corpora where retrieving relevant documents from a vast pool of information can be difficult. One potential solution to this issue is the introduction of more sophisticated filtering mechanisms, which can prioritize high-quality documents based on factors such as credibility, recency, and relevance to the query. Additionally, hybrid retrieval methods that combine both sparse and dense retrieval techniques could help mitigate the retrieval of irrelevant documents by capturing both keyword-based and semantic relevance.

5.1.2. Out-of-Distribution and Unknown Information

Another challenge is dealing with out-of-distribution (OOD) or unknown information [70]. A RAG system may encounter questions or queries that reference information not present in the training data or the retrieval corpus [71]. In such cases, the system may either generate incorrect or incomplete answers or fail to generate any meaningful output [72]. This problem is particularly concerning for tasks that involve rare or domain-specific knowledge, where the corpus may not contain enough relevant examples [73]. To address this, RAG models could benefit from incorporating external knowledge sources or memory networks that can learn to generalize over unseen information. Techniques such as continual learning or few-shot learning could also help models adapt to new information without requiring a complete retraining process [74].

5.1.3. Evaluation of Retrieval Quality

Evaluating the quality of the retrieval process is another open issue. While existing benchmarks, such as the Natural Questions (NQ) and TriviaQA datasets, provide a standard for evaluating open-domain QA tasks, they often do not capture the nuances of retrieval quality [75]. Traditional metrics like precision and recall may not fully account for the relevance or importance of the retrieved documents, leading to suboptimal evaluation of RAG performance. Future work in retrieval evaluation could explore more granular metrics that assess the relevance and utility of retrieved documents in the context of the generation task. Additionally, developing benchmarks that consider the quality of retrieval in multi-turn or interactive tasks (e.g., dialogue systems) would be crucial for understanding how retrieval quality influences performance in real-world applications [76].

 doi:10.20944/preprints202504.0351.v1

5.2. Challenges in Information Integration

Once relevant documents are retrieved, integrating this external knowledge with the generative model presents another challenge. While the concatenation or fusion of documents and queries is a common strategy, it is not always sufficient to fully leverage the retrieved information. This section discusses some of the issues related to information integration in RAG models [77].

5.2.1. Handling Ambiguity and Conflicting Information

A major challenge when integrating external knowledge is handling ambiguity or conflicting information in the retrieved documents [78]. When the retrieved documents contain differing or contradictory information, the generative model may struggle to generate a coherent and accurate output [79]. For example, in question answering, if the retrieved documents provide conflicting answers to a question, the model might generate a response that is either vague or incorrect [80]. One approach to mitigating this issue is to incorporate methods that allow the model to reason over the conflicting information and synthesize a unified response. Multi-document summarization techniques or reasoning layers could be used to detect inconsistencies and resolve conflicts before generating the final output. Additionally, confidence scoring mechanisms could help the model decide which pieces of retrieved information are more trustworthy or likely to be correct.

5.2.2. Complexity of Combining Multiple Sources

Integrating multiple retrieved documents into a coherent output is another challenge [81]. If a system retrieves a large number of documents, the generative model must be able to efficiently process and fuse the information from all sources[82]. This can lead to challenges related to memory consumption, computational efficiency, and the ability to focus on the most relevant parts of the retrieved documents. Furthermore, the model must ensure that the final output does not become overly verbose or disjointed due to the inclusion of multiple sources. Recent approaches, such as Fusion-in-Decoder (FiD) , aim to address this issue by processing each document independently in the encoder and then combining the information in the decoder. However, there is still much work to be done in terms of improving the scalability of these approaches and ensuring that the model can handle large numbers of retrieved documents without a significant performance degradation [83].

5.3. Scalability and Efficiency

Another significant challenge for RAG models is scalability, especially when working with massive corpora or real-time retrieval systems [84]. As the size of the retrieval corpus grows, the time required to retrieve relevant documents increases, leading to higher latency and computational costs. Additionally, the process of encoding both the query and the retrieved documents can be computationally expensive, especially when using large pre-trained models.

5.3.1. Efficient Retrieval and Generation

To address these scalability issues, several techniques have been proposed to speed up both the retrieval and generation components of RAG systems. One approach is to use approximate nearest neighbor (ANN) search techniques, such as FAISS or HNSW , which enable faster retrieval by approximating the closest neighbors in high-dimensional spaces. These techniques reduce the time complexity of retrieval and make it feasible to scale up the system to handle large corpora. For the generation step, model pruning, quantization, or distillation techniques can be applied to reduce the size and complexity of the generative model, thereby improving inference speed. Furthermore, multi-threading or distributed processing frameworks can help parallelize the retrieval and generation processes, allowing RAG systems to scale more effectively in production environments.

5.3.2. Real-Time Retrieval in Open-Domain Systems

In many real-time applications, such as customer support chatbots or virtual assistants, the retrieval system must respond quickly to user queries. However, in large-scale open-domain systems,

 doi:10.20944/preprints202504.0351.v1

retrieving and generating responses in real-time remains a major challenge. To address this, RAG systems must employ highly efficient retrieval mechanisms that balance both speed and accuracy. Techniques like knowledge distillation or memory augmentation, where relevant information is preloaded into memory or embedded in the model's parameters, could be effective for improving real-time response times.

5.4. Ethical and Fairness Considerations

As with any advanced AI system, RAG models raise important ethical and fairness concerns, particularly with respect to the potential for bias and misinformation [85]. The reliance on external knowledge retrieval raises questions about the sources of information, their trustworthiness, and the possibility of reinforcing harmful stereotypes or misinformation.

5.4.1. Bias in Retrieval

Bias in the retrieved documents is a major concern, as the documents retrieved from an external corpus may contain biased, misleading, or discriminatory content [86]. This could lead to biased or unfair outcomes in the generated output. For example, in a healthcare setting, if a RAG model retrieves outdated or biased medical information, it could lead to incorrect or potentially harmful advice [87]. To mitigate this issue, it is crucial to ensure that the retrieval corpus is carefully curated and regularly updated to reflect accurate, unbiased, and inclusive information [88]. Additionally, fairness-aware models could be developed to detect and correct biases in both the retrieval and generation processes [89].

5.4.2. Misinformation and Hallucination

Another challenge is the potential for RAG models to generate misinformation, especially when the retrieved documents contain inaccurate or false information. In many cases, the generative model may "hallucinate" information, meaning it produces plausible-sounding but entirely fabricated content. This is particularly concerning in applications like medical advice, legal guidance, or financial consulting, where the consequences of generating incorrect information can be severe. To address this, future research could focus on methods for grounding the generation process in verifiable external sources and enhancing the model's ability to reason about the trustworthiness of the retrieved information. One potential solution is to incorporate fact-checking mechanisms within the RAG pipeline, where the model cross-checks the retrieved information against trusted databases or verifies it through external verification tools [90].

5.5. Ongoing Research Directions

There are several exciting research directions aimed at addressing the challenges discussed in this section:

1. **Improved Retrieval Techniques:** Continued advancements in retrieval models, such as more efficient dense retrieval methods or hybrid retrieval models, can help improve the quality and relevance of the retrieved documents [91].
2. **Enhanced Information Integration:** Research into better methods for handling conflicting information and fusing multiple retrieved sources could lead to more coherent and accurate outputs.
3. **Scalable Architectures:** More scalable architectures for RAG systems, including distributed retrieval and generation, will be crucial to deploy these models in real-time applications [92].
4. **Fairness and Bias Mitigation:** Developing techniques for bias detection and correction, as well as methods for ensuring fairness in retrieval and generation, is essential for building ethical RAG systems [93].

In the following section, we provide a summary of the key points discussed in this survey and outline potential future directions for research in Retrieval-Augmented Generation.

 doi:10.20944/preprints202504.0351.v1

6. Conclusion and Future Directions

In this survey, we have explored the fundamental aspects, applications, and challenges associated with Retrieval-Augmented Generation (RAG) models. RAG has demonstrated remarkable potential in enhancing a variety of natural language processing (NLP) tasks by integrating the strengths of retrieval-based and generative approaches. By incorporating external knowledge through a retrieval process, RAG models are able to generate more accurate, contextually relevant, and informative outputs across numerous domains, including open-domain question answering, text summarization, dialogue systems, and specialized areas like medical and legal applications.

However, despite the advances made, several challenges remain in the development and deployment of RAG systems. Key challenges include ensuring the quality of retrieved documents, effectively integrating retrieved knowledge into the generation process, addressing scalability concerns, and mitigating ethical issues such as bias, misinformation, and fairness. These challenges highlight the need for ongoing research and innovation to fully unlock the potential of RAG models.

6.1. Summary of Key Points

We summarize the major contributions and findings of this survey as follows:

- **The Concept of Retrieval-Augmented Generation:** RAG models combine a retrieval mechanism with a generative model to enhance the quality of responses by grounding them in external knowledge sources.
- **Applications of RAG:** We discussed the various domains in which RAG models have been successfully applied, such as open-domain question answering, text summarization, and dialogue systems. These models have shown significant improvements over traditional methods by utilizing dynamic, real-time information from vast corpora.
- **Challenges in RAG Models:** We identified several challenges, including the quality of retrieved documents, the integration of external knowledge into the generation process, issues with scalability, and ethical concerns such as bias and misinformation.
- **Evaluation and Benchmarks:** RAG models have been evaluated on standard NLP tasks, demonstrating improvements in accuracy and relevance. However, new evaluation metrics are needed to better assess the quality of retrieval, the integration of knowledge, and the ethical implications of the generated output.

6.2. Future Directions

As RAG models continue to evolve, several exciting directions for future research emerge. Below, we outline some key areas where advancements are needed:

6.2.1. Improved Retrieval Mechanisms

While retrieval-based methods have proven effective in improving the quality of generated text, there is still room for innovation in retrieval techniques. Future research could focus on:

- **Hybrid Retrieval Models:** Exploring hybrid models that combine sparse (e.g., TF-IDF, BM25) and dense (e.g., DPR, Sentence-BERT) retrieval techniques could further enhance the accuracy and efficiency of the retrieval process.
- **End-to-End Retrieval-Augmented Systems:** There is an opportunity to develop end-to-end architectures that seamlessly combine retrieval and generation processes, reducing the reliance on separate components and improving system integration.
- **Context-Aware Retrieval:** Current retrieval methods do not always consider the full conversational or document context. Future retrieval models should account for broader context to improve the relevance of retrieved documents, especially in multi-turn dialogue or long document summarization tasks.

 doi:10.20944/preprints202504.0351.v1

6.2.2. Enhanced Information Fusion and Reasoning

Integrating retrieved information into generative models remains a complex task. Future work could focus on improving how generative models reason over and synthesize multiple retrieved sources. Some possible directions include:

- **Multi-Document Reasoning:** Building models that can handle and reason over multiple documents simultaneously, allowing them to synthesize diverse pieces of information into a coherent and contextually grounded output.
- **Knowledge Graphs and Structured Data:** Incorporating structured knowledge from external sources, such as knowledge graphs or databases, could help RAG models reason more effectively and make better decisions based on factual relationships between entities.
- **Explainable Generation:** Research into explainable AI for generative models could allow RAG systems to provide explanations for their retrieved knowledge and reasoning, increasing trust and transparency in applications such as healthcare or law.

6.2.3. Scalability and Efficiency Improvements

As RAG models grow in complexity and scale, it will become increasingly important to address issues related to computational efficiency and scalability. Key directions include:

- **Optimized Retrieval Pipelines:** New techniques to speed up the retrieval process without compromising accuracy are necessary, especially in real-time applications. Methods such as quantization, pruning, and approximate nearest neighbor (ANN) search can be further optimized for RAG systems.
- **Model Distillation and Compression:** To reduce computational cost, distilling large RAG models into smaller, more efficient variants while retaining performance will be crucial for deploying these models in resource-constrained environments[94].
- **Efficient Inference:** Developing new inference techniques to streamline both the retrieval and generation phases of RAG systems, allowing them to scale up to larger corpora and deliver responses in real-time, is a key challenge for the next generation of systems.

6.2.4. Ethical Considerations and Fairness

As RAG models become more widely deployed, addressing ethical challenges such as bias, misinformation, and fairness will be essential. Future research should focus on:

- **Bias Mitigation in Retrieval and Generation:** Investigating methods to detect and mitigate biases both in the retrieval stage (e.g., biased corpora) and in the generative process (e.g., biased language models) will be key to ensuring fairness and equity in RAG systems.
- **Misinformation Detection and Prevention:** Developing strategies to detect and prevent the generation of false or misleading content is essential, particularly in high-stakes domains such as medical, legal, or financial advice.
- **Transparency and Accountability:** Research into mechanisms for improving transparency in RAG models, such as providing insights into the decision-making process and the sources of retrieved information, will be critical in building user trust and accountability in AI systems.

6.2.5. Personalized and Multi-Domain Systems

As RAG models continue to evolve, the ability to personalize responses based on user-specific information or preferences could open up new possibilities. In addition, extending the capabilities of RAG models to handle multiple domains simultaneously—such as general knowledge, specialized domains (e.g., medicine or law), and personal preferences—would greatly enhance their usability and flexibility. Research into transfer learning, few-shot learning, and domain adaptation will be crucial for enabling RAG systems to operate effectively in diverse and dynamic environments.

 doi:10.20944/preprints202504.0351.v1

6.3. Final Thoughts

The field of Retrieval-Augmented Generation has made significant strides in improving the performance of NLP systems by leveraging external knowledge sources. However, much work remains to be done in addressing challenges related to retrieval quality, information integration, scalability, and ethics. As RAG models continue to evolve and integrate advances in retrieval techniques, generation architectures, and ethical frameworks, they have the potential to revolutionize a wide range of applications, from conversational agents and question answering to specialized domain applications in medicine and law. By addressing these challenges and pursuing the research directions outlined in this survey, RAG models will continue to push the boundaries of what is possible in natural language processing and AI.

References

1. Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. Self-Refine: Iterative Refinement with Self-Feedback, 2023, [arXiv:cs.CL/2303.17651].
2. Thakur, N.; Reimers, N.; Rücklé, A.; Srivastava, A.; Gurevych, I. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663* **2021**.
3. Narayan, S.; Cohen, S.B.; Lapata, M. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization, 2018, [arXiv:cs.CL/1808.08745].
4. Wang, H.; Hu, M.; Deng, Y.; Wang, R.; Mi, F.; Wang, W.; Wang, Y.; Kwan, W.C.; King, I.; Wong, K.F. Large Language Models as Source Planner for Personalized Knowledge-grounded Dialogue. *arXiv preprint arXiv:2310.08840* **2023**.
5. Fan, A.; Jernite, Y.; Perez, E.; Grangier, D.; Weston, J.; Auli, M. ELI5: Long form question answering. *arXiv preprint arXiv:1907.09190* **2019**.
6. Nebel, B. On the compilability and expressive power of propositional planning formalisms. *Journal of Artificial Intelligence Research* **2000**, *12*, 271–315.
7. Trivedi, H.; Balasubramanian, N.; Khot, T.; Sabharwal, A. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509* **2022**.
8. Baek, J.; Aji, A.F.; Saffari, A. Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering. *arXiv preprint arXiv:2306.04136* **2023**.
9. Welbl, J.; Stenetorp, P.; et al. 2WikiMultiHopQA: Multihop Question Answering over Wikipedia Articles. *EMNLP* **2018**.
10. Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics* **2019**, *7*, 453–466.
11. Saad-Falcon, J.; Khattab, O.; Potts, C.; Zaharia, M. ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems. *arXiv preprint arXiv:2311.09476* **2023**.
12. Saha, A.; Pahuja, V.; Khapra, M.M.; Sankaranarayanan, K.; Chandar, S. Complex Sequential Question Answering: Towards Learning to Converse Over Linked Question Answer Pairs with a Knowledge Graph **2018**. [arXiv:1801.10314].
13. LangChain. LangSmith: The Ultimate Toolkit for Debugging and Monitoring LLM Applications. https://www.langchain.com/langsmith, 2025. Accessed: 2025-01-28.
14. Geva, M.; Khashabi, D.; Segal, E.; Khot, T.; Roth, D.; Berant, J. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies, 2021, [arXiv:cs.CL/2101.02235].
15. Qin, Y.; Cai, Z.; Jin, D.; Yan, L.; Liang, S.; Zhu, K.; Lin, Y.; Han, X.; Ding, N.; Wang, H.; et al. WebCPM: Interactive Web Search for Chinese Long-form Question Answering. *arXiv preprint arXiv:2305.06849* **2023**.
16. Li, S.; Ji, H.; Han, J. Document-Level Event Argument Extraction by Conditional Generation, 2021, [arXiv:cs.CL/2104.05919].
17. Pan, F.; Canim, M.; Glass, M.; Gliozzo, A.; Hendler, J. End-to-End Table Question Answering via Retrieval-Augmented Generation. *arXiv preprint arXiv:2203.16714* **2022**.
18. Li, X.; Zhao, R.; Chia, Y.K.; Ding, B.; Bing, L.; Joty, S.; Poria, S. Chain of Knowledge: A Framework for Grounding Large Language Models with Structured Knowledge Bases. *arXiv preprint arXiv:2305.13269* **2023**.
19. Lan, T.; Cai, D.; Wang, Y.; Huang, H.; Mao, X.L. Copy is All You Need. In Proceedings of the The Eleventh International Conference on Learning Representations, 2022.

 doi:10.20944/preprints202504.0351.v1

20. Khattab, O.; Zaharia, M. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In Proceedings of the Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 39–48.
21. Ranzato, M.; Chopra, S.; Auli, M.; Zaremba, W. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732* **2015**.
22. Lyu, Y.; Li, Z.; Niu, S.; Xiong, F.; Tang, B.; Wang, W.; Wu, H.; Liu, H.; Xu, T.; Chen, E. CRUD-RAG: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *arXiv preprint arXiv:2401.17043* **2024**.
23. Chen, D.; Yih, W.t. Open-domain question answering. In Proceedings of the Proceedings of the 58th annual meeting of the association for computational linguistics: tutorial abstracts, 2020, pp. 34–37.
24. Cheng, X.; Gao, S.; Liu, L.; Zhao, D.; Yan, R. Neural machine translation with contrastive translation memories. *arXiv preprint arXiv:2212.03140* **2022**.
25. Zheng, H.S.; Mishra, S.; Chen, X.; Cheng, H.T.; Chi, E.H.; Le, Q.V.; Zhou, D. Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models. *arXiv preprint arXiv:2310.06117* **2023**.
26. VoyageAI. Voyage's embedding models. https://docs.voyageai.com/embeddings/, 2023.
27. Xia, M.; Huang, G.; Liu, L.; Shi, S. Graph based translation memory for neural machine translation. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2019, Vol. 33, pp. 7297–7304.
28. Zhao, W.X.; Liu, J.; Ren, R.; Wen, J.R. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems* **2024**, *42*, 1–60.
29. Zhang, Y.; Li, Y.; Cui, L.; Cai, D.; Liu, L.; Fu, T.; Huang, X.; Zhao, E.; Zhang, Y.; Chen, Y.; et al. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *arXiv preprint arXiv:2309.01219* **2023**.
30. Elsahar, H.; Vougiouklis, P.; Remaci, A.; Gravier, C.; Hare, J.; Laforest, F.; Simperl, E. T-rex: A large scale alignment of natural language with knowledge base triples. In Proceedings of the Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), 2018.
31. Yoran, O.; Wolfson, T.; Ram, O.; Berant, J. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558* **2023**.
32. DeepLearning.AI. How Agents Can Improve LLM Performance. https://www.deeplearning.ai/the-batch/how-agents-can-improve-llm-performance/?ref=dl-staging-website.ghost.io, 2024. Accessed: 2025-01-13.
33. Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.W.; Salakhutdinov, R.; Manning, C.D. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering, 2018, [arXiv:cs.CL/1809.09600].
34. Melz, E. Enhancing llm intelligence with arm-rag: Auxiliary rationale memory for retrieval augmented generation. *arXiv preprint arXiv:2311.04177* **2023**.
35. Zhang, P.; Xiao, S.; Liu, Z.; Dou, Z.; Nie, J.Y. Retrieve Anything To Augment Large Language Models. *arXiv preprint arXiv:2310.07554* **2023**.
36. Shi, T.; Li, L.; Lin, Z.; Yang, T.; Quan, X.; Wang, Q. Dual-Feedback Knowledge Retrieval for Task-Oriented Dialogue Systems. *arXiv preprint arXiv:2310.14528* **2023**.
37. Yan, S.Q.; Gu, J.C.; Zhu, Y.; Ling, Z.H. Corrective Retrieval Augmented Generation. *arXiv preprint arXiv:2401.15884* **2024**.
38. Kim, G.; Kim, S.; Jeon, B.; Park, J.; Kang, J. Tree of Clarifications: Answering Ambiguous Questions with Retrieval-Augmented Large Language Models. *arXiv preprint arXiv:2310.14696* **2023**.
39. Xu, P.; Ping, W.; Wu, X.; McAfee, L.; Zhu, C.; Liu, Z.; Subramanian, S.; Bakhturina, E.; Shoeybi, M.; Catanzaro, B. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025* **2023**.
40. Kotonya, N.; Toni, F. Explainable Automated Fact-Checking for Public Health Claims, 2020, [arXiv:cs.CL/2010.09926].
41. Repository, L.D. Research Paper Report Generation Workflow using LlamaCloud. https://github.com/run-llama/llamacloud-demo/blob/main/examples/report_generation/research_paper_report_generation.ipynb, 2025. Accessed: 2025-01-13.
42. Wang, Z.; Araki, J.; Jiang, Z.; Parvez, M.R.; Neubig, G. Learning to filter context for retrieval-augmented generation. *arXiv preprint arXiv:2311.08377* **2023**.
43. Robertson, S.; Zaragoza, H.; et al. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* **2009**, *3*, 333–389.
44. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A Survey of Large Language Models, 2024, [arXiv:cs.CL/2303.18223].
45. Shao, Z.; Gong, Y.; Shen, Y.; Huang, M.; Duan, N.; Chen, W. Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy. *arXiv preprint arXiv:2305.15294* **2023**.

 doi:10.20944/preprints202504.0351.v1

46. Yoran, O.; Wolfson, T.; Ram, O.; Berant, J. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558* **2023**.
47. Ovadia, O.; Brief, M.; Mishaeli, M.; Elisha, O. Fine-tuning or retrieval? comparing knowledge injection in llms. *arXiv preprint arXiv:2312.05934* **2023**.
48. Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; Deng, L. Ms marco: A human-generated machine reading comprehension dataset **2016**.
49. Purwar, A.; Sundar, R. Keyword Augmented Retrieval: Novel framework for Information Retrieval integrated with speech interface. *arXiv preprint arXiv:2310.04205* **2023**.
50. Sun, Z.; Wang, X.; Tay, Y.; Yang, Y.; Zhou, D. Recitation-augmented language models. *arXiv preprint arXiv:2210.01296* **2022**.
51. Xiao, G.; Tian, Y.; Chen, B.; Han, S.; Lewis, M. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453* **2023**.
52. Gou, Z.; Shao, Z.; Gong, Y.; Shen, Y.; Yang, Y.; Duan, N.; Chen, W. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing, 2024, [arXiv:cs.CL/2305.11738].
53. Cheng, X.; Luo, D.; Chen, X.; Liu, L.; Zhao, D.; Yan, R. Lift Yourself Up: Retrieval-augmented Text Generation with Self Memory. *arXiv preprint arXiv:2305.02437* **2023**.
54. Raudaschl, A.H. Forget RAG, the Future is RAG-Fusion. https://towardsdatascience.com/forget-rag-the-future-is-rag-fusion-1147298d8ad1, 2023.
55. Cerny, T.; Abdelfattah, A.S.; Bushong, V.; Maruf, A.A.; Taibi, D. Microservice Architecture Reconstruction and Visualization Techniques: A Review, 2022, [arXiv:cs.SE/2207.02988].
56. He, X.; Tian, Y.; Sun, Y.; Chawla, N.V.; Laurent, T.; LeCun, Y.; Bresson, X.; Hooi, B. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering, 2024, [arXiv:cs.LG/2402.07630].
57. Kapoor, S.; Stroebl, B.; Siegel, Z.S.; Nadgir, N.; Narayanan, A. AI Agents That Matter, 2024, [arXiv:cs.LG/2407.01502].
58. Pang, R.Y.; Parrish, A.; Joshi, N.; Nangia, N.; Phang, J.; Chen, A.; Padmakumar, V.; Ma, J.; Thompson, J.; He, H.; et al. QuALITY: Question answering with long input texts, yes! *arXiv preprint arXiv:2112.08608* **2021**.
59. Luo, Z.; Xu, C.; Zhao, P.; Geng, X.; Tao, C.; Ma, J.; Lin, Q.; Jiang, D. Augmented Large Language Models with Parametric Knowledge Guiding. *arXiv preprint arXiv:2305.04757* **2023**.
60. Mavromatis, C.; Karypis, G. GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning, 2024, [arXiv:cs.CL/2405.20139].
61. Singh, A. A Survey of AI Text-to-Image and AI Text-to-Video Generators. In Proceedings of the 2023 4th International Conference on Artificial Intelligence, Robotics and Control (AIRC), 2023, pp. 32–36. https://doi.org/10.1109/AIRC57904.2023.10303174.
62. Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. Training Verifiers to Solve Math Word Problems, 2021, [arXiv:cs.LG/2110.14168].
63. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the Proceedings of the 2013 conference on empirical methods in natural language processing, 2013, pp. 1631–1642.
64. Yang, H.; Yue, S.; He, Y. Auto-GPT for Online Decision Making: Benchmarks and Additional Opinions. *arXiv preprint arXiv:2306.02224* **2023**.
65. Hayashi, H.; Budania, P.; Wang, P.; Ackerson, C.; Neervannan, R.; Neubig, G. WikiAsp: A dataset for multi-domain aspect-based summarization. *Transactions of the Association for Computational Linguistics* **2021**, *9*, 211–225.
66. Wang, S.; Xu, Y.; Fang, Y.; Liu, Y.; Sun, S.; Xu, R.; Zhu, C.; Zeng, M. Training data is more valuable than you think: A simple and effective method by retrieving from training data. *arXiv preprint arXiv:2203.08773* **2022**.
67. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*.
68. Levesque, H.J. Foundations of a functional approach to knowledge representation. *Artificial Intelligence* **1984**, *23*, 155–212.
69. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* **2022**, *35*, 27730–27744.

 doi:10.20944/preprints202504.0351.v1

70. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Advances in neural information processing systems* **2020**, *33*, 1877–1901.
71. Wen, T.H.; Gasic, M.; Mrksic, N.; Rojas-Barahona, L.M.; Su, P.H.; Ultes, S.; Vandyke, D.; Young, S. Conditional generation and snapshot learning in neural dialogue systems. *arXiv preprint arXiv:1606.03352* **2016**.
72. Wang, X.; Yang, Q.; Qiu, Y.; Liang, J.; He, Q.; Gu, Z.; Xiao, Y.; Wang, W. KnowledGPT: Enhancing Large Language Models with Retrieval and Storage Access on Knowledge Bases. *arXiv preprint arXiv:2308.11761* **2023**.
73. Chan, D.M.; Ghosh, S.; Rastrow, A.; Hoffmeister, B. Using External Off-Policy Speech-To-Text Mappings in Contextual End-To-End Automated Speech Recognition. *arXiv preprint arXiv:2301.02736* **2023**.
74. Dam, S.K.; Hong, C.S.; Qiao, Y.; Zhang, C. A Complete Survey on LLM-based AI Chatbots, 2024, [arXiv:cs.CL/2406.16937].
75. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M.; Wang, H. Retrieval-Augmented Generation for Large Language Models: A Survey, 2024, [arXiv:cs.CL/2312.10997].
76. Kočiský, T.; Schwarz, J.; Blunsom, P.; Dyer, C.; Hermann, K.M.; Melis, G.; Grefenstette, E. The NarrativeQA Reading Comprehension Challenge **2017**. [arXiv:cs.CL/1712.07040].
77. Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework. 2023, [arXiv:cs.AI/2308.08155].
78. Husain, H.; Wu, H.H.; Gazit, T.; Allamanis, M.; Brockschmidt, M. Codesearchnet challenge: Evaluating the state of semantic code search. *arXiv preprint arXiv:1909.09436* **2019**.
79. Ren, Y.; Cao, Y.; Guo, P.; Fang, F.; Ma, W.; Lin, Z. Retrieve-and-sample: Document-level event argument extraction via hybrid retrieval augmentation. In Proceedings of the Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 293–306.
80. Li, J.; Li, D.; Savarese, S.; Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597* **2023**.
81. Ravuru, C.; Sakhinana, S.S.; Runkana, V. Agentic Retrieval-Augmented Generation for Time Series Analysis, 2024, [arXiv:cs.AI/2408.14484].
82. Zniyed, Y.; Nguyen, T.P.; et al. Efficient tensor decomposition-based filter pruning. *Neural Networks* **2024**, *178*, 106393.
83. Trivedi, H.; Balasubramanian, N.; Khot, T.; Sabharwal, A. MuSiQue: Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics* **2022**, *10*, 539–554.
84. Li, X.; Nie, E.; Liang, S. From Classification to Generation: Insights into Crosslingual Retrieval Augmented ICL. *arXiv preprint arXiv:2311.06595* **2023**.
85. Wang, L.; Yang, N.; Wei, F. Query2doc: Query Expansion with Large Language Models. *arXiv preprint arXiv:2303.07678* **2023**.
86. He, R.; McAuley, J. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In Proceedings of the Proceedings of the 25th International Conference on World Wide Web, Republic and Canton of Geneva, CHE, 2016; WWW '16, p. 507–517. https://doi.org/10.1145/2872427.2883037.
87. Sarthi, P.; Abdullah, S.; Tuli, A.; Khanna, S.; Goldie, A.; Manning, C.D. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval. *arXiv preprint arXiv:2401.18059* **2024**.
88. Wang, X.; Chen, G.H.; Song, D.; Zhang, Z.; Chen, Z.; Xiao, Q.; Jiang, F.; Li, J.; Wan, X.; Wang, B.; et al. CMB: A Comprehensive Medical Benchmark in Chinese, 2024, [arXiv:cs.CL/2308.08833].
89. Bisk, Y.; Zellers, R.; Gao, J.; Choi, Y.; et al. Piqa: Reasoning about physical commonsense in natural language. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2020, Vol. 34, pp. 7432–7439.
90. Li, X.; Li, J. AnglE-optimized Text Embeddings. *arXiv preprint arXiv:2309.12871* **2023**.
91. NVIDIA. Spectrum-X: End-to-End Networking for AI and High-Performance Computing. https://www.nvidia.com/en-us/networking/spectrumx/, 2025. Accessed: 2025-01-28.
92. Lebret, R.; Grangier, D.; Auli, M. Neural text generation from structured data with application to the biography domain. *arXiv preprint arXiv:1603.07771* **2016**.

 doi:10.20944/preprints202504.0351.v1

93. Baek, J.; Jeong, S.; Kang, M.; Park, J.C.; Hwang, S.J. Knowledge-Augmented Language Model Verification. *arXiv preprint arXiv:2310.12836* **2023**.
94. Zniyed, Y.; Nguyen, T.P.; et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.