Large Language Models (LLMs) demonstrate significant capabilities but face challenges such as hallucination, outdated knowledge, and nontransparent, untraceable reasoning processes. Retrieval-Augmented Generation (RAG) has emerged as a promising solution by incorporating knowledge from external databases. This enhances the accuracy and credibility of the models, particularly for knowledge-intensive tasks, and allows for continuous knowledge updates and integration of domain-specific information. RAG synergistically merges LLMs' intrinsic knowledge with the vast, dynamic repositories of external databases. This comprehensive review paper offers a detailed examination of the progression of RAG paradigms, encompassing the Naive RAG, the Advanced RAG, and the Modular RAG. It meticulously scrutinizes the tripartite foundation of RAG frameworks, which includes the retrieval , the generation and the augmentation techniques. The paper highlights the state-of-the-art technologies embedded in each of these critical components, providing a profound understanding of the advancements in RAG systems. Furthermore, this paper introduces the metrics and benchmarks for assessing RAG models, along with the most up-to-date evaluation framework. In conclusion, the paper delineates prospective avenues for research, including the identification of challenges, the expansion of multi-modalities, and the progression of the RAG infrastructure and its ecosystem.

5.1 Post-retrieval with Frozen LLM

In the realm of untunable LLMs , many studies rely on wellestablished models like GPT-4 [OpenAI, 2023] to harness their comprehensive internal knowledge for systematically synthesizing retrieved information from various documents. However, challenges persist with these large models, including limitations on context length and susceptibility to redundant information. To tackle these issues, certain research endeavors have turned their focus to post-retrieval processing. Post-retrieval processing involves treating, filtering, or optimizing the relevant information retrieved by the retriever from a large document database. Its main goal is to enhance the quality of retrieval results, aligning them more closely with user needs or subsequent tasks. It can be viewed as a reprocessing of the documents obtained during the retrieval phase. Common operations in post-retrieval processing typically include information compression and result reranking.

Information Compression

The retriever excels at retrieving relevant information from a vast knowledge base, but managing the substantial amount of information within retrieval documents is a challenge. Ongoing research aims to extend the context length of large language models to tackle this issue. However, current large models still struggle with context limitations. Therefore, there are scenarios where condensing information becomes necessary. Information condensation is significant for reducing noise, addressing context length restrictions, and enhancing generation effects. PRCA tackled this issue by training an information extractor [Yang et al., 2023b]. In the context extraction phase, when provided with an input text Sinput, it is capable of producing an output sequence Cextracted that represents the condensed context from the input document. The training process is designed to minimize the difference between Cextracted and the actual context Ctruth. Similarly, RECOMP adopts a comparable approach by training an information

condenser using contrastive learning [Xu et al., 2023a]. Each training data point consists of one positive sample and five negative samples, and the encoder undergoes training using contrastive loss throughout this process [Karpukhin et al., 2020] . Another study has taken a different approach by aiming to reduce the number of documents in order to improve the accuracy of the model's answers. In the study by [Ma et al., 2023b], they propose the "Filter-Reranker" paradigm, which combines the strengths of LLMs and Small Language Models (SLMs). In this paradigm, SLMs serve as filters, while LLMs function as reordering agents. The research shows that instructing LLMs to rearrange challenging samples identified by SLMs leads to significant improvements in various Information Extraction (IE) tasks.

Reranking

The re-ranking model is pivotal in optimizing the document set retrieved from the retriever. Language models often face performance declines when additional context is introduced, and re-ranking effectively addresses this issue. The core concept involves rearranging document records to prioritize the most relevant items at the top, thereby limiting the total number of documents. This not only resolves the challenge of context window expansion during retrieval but also enhances retrieval efficiency and responsiveness. The re-ranking model assumes a dual role throughout the information retrieval process, functioning as both an optimizer and a refiner. It provides more effective and accurate input for subsequent language model processing [Zhuang et al., 2023]. Contextual compression is incorporated into the reordering process to offer more precise retrieval information. This method entails reducing the content of individual documents and filtering the entire document, with the ultimate goal of presenting the most relevant information in the search results for a more focused and accurate display of pertinent content.

5.2 Fine-tuning LLM for RAG

Optimizing the generator within the RAG model is a critical aspect of its architecture. The generator's role is to take the retrieved information and produce relevant text, forming the final output of the model. The optimization of the generator aims to ensure that the generated text is both natural and effectively leverages the retrieved documents to better meet the user's query needs. In standard LLMs generation tasks, the input typically consists of a query. RAG stands out by incorporating not only a query but also various retrieved documents (structured/unstructured) by the retriever into the input. This additional information can significantly influence the model's understanding, particularly for smaller models. In such cases, fine-tuning the model to adapt to the input of both query and retrieved documents becomes crucial. Before presenting the input to the fine-tuned model, post-retrieval processing usually occurs for the documents retrieved by the retriever. It is essential to note that the fine-tuning method for the generator in RAG aligns with the general fine-tuning approach for LLMs. In the following, we will briefly describe some representative works involving data (formatted/unformatted) and optimization functions.

General Optimization Process

As part of the general optimization process, the training data typically consists of input-output pairs, aiming to train the model to produce the output y given the input x. In the work of Self-Mem [Cheng et al., 2023b], a traditional training process is employed, where given the input x, relevant documents z are retrieved (selecting Top-1 in the paper), and after integrating (x, z), the model generates the output y. The paper utilizes two common paradigms for fine-tuning, namely Joint-Encoder and Dual-Encoder [Arora et al., 2023, Wang et al., 2022b, Lewis et al., 2020, Xia et al., 2019, Cai et al., 2021, Cheng et al., 2022]. In the Joint-Encoder paradigm, a standard model based on an encoder-decoder is used. Here, the encoder initially encodes the input, and the decoder, through attention mechanisms, combines the encoded results to generate tokens in an autoregressive manner. On the other hand, in the DualEncoder paradigm, the system sets up two independent encoders, with each encoder encoding the input (query, context) and the document, respectively. The resulting outputs undergo bidirectional cross-attention processing by the decoder in sequence. Both architectures utilize the Transformer [Vaswani et al., 2017] as the foundational block and optimize with Negative Log-Likelihood loss. Utilizing Contrastive Learning In the phase of preparing training data for language models, interaction pairs of input and output are usually created. This traditional method can lead to ”exposure bias,” where the model is only trained on individual, correct output examples, thus restricting its exposure to a range of possible outputs citesequence. This limitation can hinder the model’s real-world performance by causing it to overfit to the particular examples in the training set, thereby reducing its ability to generalize across various contexts. To mitigate exposure bias, SURGE [Kang et al., 2023] proposes the use of graph-text contrastive learning. This method includes a contrastive learning objective that prompts the model to produce a range of plausible and coherent responses, expanding beyond the instances encountered in the training data. This approach is crucial in reducing overfitting and strengthening the model’s ability to generalize. For retrieval tasks that engage with structured data, the SANTA framework [Li et al., 2023d] implements a tripartite training regimen to effectively encapsulate both structural and semantic nuances. The initial phase focuses on the retriever, where contrastive learning is harnessed to refine the query and document embeddings. Subsequently, the generator’s preliminary training stage employs contrastive learning to align the structured data with its unstructured document descriptions. In a further stage of generator training, the model acknowledges the critical role of entity semantics in the representation learning of textual data for retrieval, as highlighted by [Sciavolino et al., 2021, Zhang et al., 2019]. This process commences with the identification of entities within the structured data, followed by the application of masks over these entities within the generator’s input data, thus setting the stage for the model to anticipate and predict these masked elements. The training regimen progresses with the model learning to reconstruct the masked entities by leveraging contextual information. This exercise cultivates the model’s comprehension of the textual data’s structural semantics and facilitates the alignment of pertinent entities within the structured data. The overarching optimization goal is to train the language model to accurately restore the obscured spans, thereby enriching its understanding of entity semantics[Ye et al., 2020].

6 Augmentation in RAG

This section is structured around three key aspects: the augmentation stage, sources of augmentation data, and the augmentation process. These facets elucidate the critical technologies pivotal to RAG's development. A taxonomy of RAG's core components is presented in Figure 4. 6.1 RAG in Augmentation Stages RAG, a knowledge-intensive endeavor, incorporates a variety of technical methodologies across the pre-training, finetuning, and inference stages of language model training.

Pre-training Stage

During the pre-training stage, researchers have investigated methods to bolster PTMs for open-domain QA through Figure 4: Taxonomy of RAG's core components retrieval-based strategies. The REALM model adopts a structured, interpretable method for knowledge embedding, framing pre-training, and fine-tuning as a retrieve-then-predict workflow within the masked language model (MLM) framework [Arora et al., 2023] . RETRO [Borgeaud et al., 2022] leverages retrieval augmentation for large-scale pre-training from scratch, achieving a reduction in model parameters while surpassing standard GPT models in terms of perplexity. RETRO distinguishes itself with an additional encoder designed to process features of entities retrieved from an external knowledge base, building on the foundational structure of GPT models. Atlas[Izacard et al., 2022] also incorporates a retrieval mechanism into the T5 architecture [Raffel et al., 2020] in both the pre-training and fine-tuning stages. It uses a pretrained T5 to initialize the encoder-decoder language model and a pre-trained Contriever for the dense retriever, improving its efficiency for complex language modeling tasks. Furthermore, COG [Lan et al., 2022] introduces a novel text generation methodology that emulates copying text fragments from pre-existing collections. Utilizing efficient vector search tools, COG computes and indexes contextually meaningful representations of text fragments, demonstrating superior performance in domains such as question-answering and domain adaptation when compared to RETRO. The advent of scaling laws has catalyzed the growth of model parameters, propelling autoregressive models into the mainstream. Researchers are expanding the RAG approach to pretrained larger models, with RETRO++ exemplifying this trend by scaling up the model parameters while preserving or enhancing performance [Wang et al., 2023b]. Empirical evidence underscores marked improvements in text generation quality, factual accuracy, reduced toxicity, and downstream task proficiency, especially in knowledgeintensive applications like open-domain QA. These results imply that integrating retrieval mechanisms into the pre-training of autoregressive language models constitutes a promising avenue, marrying sophisticated retrieval techniques with expansive language models to yield more precise and efficient language generation. The benefits of augmented pre-training include a robust foundational model that outperforms standard GPT models in perplexity, text generation quality, and task-specific performance, all while utilizing fewer parameters. This method is particularly adept at handling knowledge-intensive tasks and facilitates the development of domain-specific models through training on specialized corpora. Nonetheless, this approach faces challenges such as the necessity for extensive pre-training datasets and resources, as well as diminished update frequencies with increasing model sizes. Despite these hurdles, the approach offers significant advantages in model resilience. Once trained, retrieval-enhanced models can

operate independently of external libraries, enhancing generation speed and operational efficiency. The potential gains identified render this methodology a compelling subject for ongoing investigation and innovation in artificial intelligence and machine learning.

Fine-tuning Stage

RAG and Fine-tuning are powerful tools for enhancing LLMs, and combining the two can meet the needs of more specific scenarios. On one hand, fine-tuning allows for the retrieval of documents with a unique style, achieving better semantic expression and aligning the differences between queries and documents. This ensures that the output of the retriever is more aptly suited to the scenario at hand. On the other hand, fine-tuning can fulfill the generation needs of making stylized and targeted adjustments. Furthermore, finetuning can also be used to align the retriever and generator for improved model synergy. The main goal of fine-tuning the retriever is to improve the quality of semantic representations, achieved by directly fine-tuning the Embedding model using a corpus [Liu, 2023]. By aligning the retriever's capabilities with the preferences of the LLMs through feedback signals, both can be better coordinated [Yu et al., 2023b, Izacard et al., 2022, Yang et al., 2023b, Shi et al., 2023]. Fine-tuning the retriever for specific downstream tasks can lead to improved adaptability [cite]. The introduction of task-agnostic fine-tuning aims to enhance the retriever's versatility in multi-task scenarios [Cheng et al., 2023a]. Fine-tuning generator can result in outputs that are more stylized and customized. On one hand, it allows for specialized adaptation to different input data formats. For example, fine-tuning LLMs to fit the structure of knowledge graphs [Kang et al., 2023], the structure of text pairs [Kang et al., 2023, Cheng et al., 2023b], and other specific structures [Li et al., 2023d]. On the other hand, by constructing directive datasets, one can demand LLMs to generate specific formats content. For instance, in adaptive or iterative retrieval scenarios, LLMs are fine-tuned to generate content that will help determine the timing for the next step of action [Jiang et al., 2023b, Asai et al., 2023]. By synergistically fine-tuning both the retriever and the generator, we can enhance the model's generalization capabilities and avoid overfitting that may arise from training them separately. However, joint fine-tuning also leads to increased resource consumption. RA-DIT [Lin et al., 2023] presents a lightweight, dual-instruction tuning framework that can effectively add retrieval capabilities to any LLMs. The retrieval-enhanced directive fine-tuning updates the LLM, guiding it to make more efficient use of the information retrieved and to disregard distracting content. Despite its advantages, fine-tuning has limitations, including the need for specialized datasets for RAG fine-tuning and the requirement for significant computational resources. However, this stage allows for customizing models to specific needs and data formats, potentially reducing resource usage compared to the pre-training phase while still being able to fine-tune the model's output style. In summary, the fine-tuning stage is essential for the adaptation of RAG models to specific tasks, enabling the refinement of both retrievers and generators. This stage enhances the model's versatility and adaptability to various tasks, despite the challenges presented by resource and dataset requirements. The strategic fine-tuning of RAG models is therefore a critical component in the development of efficient and effective retrieval-augmented systems.

Inference Stage

The inference stage in RAG models is crucial, as it involves extensive integration with LLMs. Traditional RAG approaches, also known as Naive RAG, involve incorporating retrieval content at this stage to guide the generation process. To overcome the limitations of Naive RAG, advanced techniques introduce more contextually rich information during inference. The DSP framework [Khattab et al., 2022] utilizes a sophisticated exchange of natural language text between fronzen LMs and retrieval models (RMs), enriching the context and thereby improving generation outcomes. The PKG [Luo et al., 2023] method equips LLMs with a knowledge-guided module that allows for the retrieval of pertinent information without modifying the LMs' parameters, enabling more complex task execution. CREAICL [Li et al., 2023b] employs a synchronous retrieval of cross-lingual knowledge to enhance context, while RECITE [Sun et al., 2022] generates context by sampling paragraphs directly from LLMs. Further refinement of the RAG process during inference is seen in approaches that cater to tasks necessitating multi-step reasoning. ITRG [Feng et al., 2023] iteratively retrieves information to identify the correct reasoning paths, thereby improving task adaptability. ITERRETGEN [Shao et al., 2023] follows an iterative strategy, merging retrieval and generation in a cyclical process that alternates between "retrieval-enhanced generation" and "generation-enhanced retrieval". For non-knowledgeintensive (NKI) tasks, PGRA [Guo et al., 2023] proposes a two-stage framework, starting with a task-agnostic retriever followed by a prompt-guided reranker to select and prioritize evidence. In contrast, IRCOT [Trivedi et al., 2022] combines RAG with Chain of Thought (CoT) methodologies, alternating CoT-guided retrievals with retrieval-informed CoT processes, significantly boosting GPT-3's performance across various question-answering tasks. In essence, these inference-stage enhancements provide lightweight, cost-effective alternatives that leverage the capabilities of pre-trained models without necessitating further training. The principal advantage is maintaining static LLM parameters while supplying contextually relevant information to meet specific task demands. Nevertheless, this approach is not without limitations, as it requires meticulous data processing and optimization, and is bound by the foundational model's intrinsic capabilities. To address diverse task requirements effectively, this method is often paired with procedural optimization techniques such as step-wise reasoning, iterative retrieval, and adaptive retrieval strategies.