JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A CRITICAL REVIEW OF LARGE LANGUAGE MODELS (LLM) AND RETRIEVAL-AUGMENTED GENERATION (RAG)

[1]Rashmi P, [2]Dr.Janapati Venkata Krishna
[1]PG Student, [2]Associate Professor,
[1]Computer science and Engineering,
Srinivas University, Mukka, Mangalore, India

Abstract: Retrieval-Augmented Generation (RAG) combines two essential technologies: information retrieval and generative AI. Training large language models (LLMs) requires large volumes of data, and keeping these models up-to-date or tailored to specific datasets for specialized applications is a significant challenge. This is where RAG comes in. It enables generative AI models or LLMs to retrieve information from outside sources, including enterprise databases or the internet, in real-time. By retrieving relevant information and augmenting the input prompt, RAG allows the model to generate responses that incorporate real-time, specific, and accurate data. This paper explores the core concepts, architecture, and applications of RAG. It also delves into the limitations of LLMs, the definition and significance of RAG, the architecture of RAG and its potential to enhance AI systems across diverse use cases.

Index Terms **– Machine Learning, RAG, LLM, NLP, Open AI.**

I. INTRODUCTION

Retrieval-Augmented Generation (RAG) has become a prominent approach to overcome key limitations of Large Language Models (LLMs), which include reliance on broad, static training data and the inability to access up-to-date, private, or domain-specific information. [2, 3, 4]. Consider the scenario where a business wants to create a chatbot tailored to its specific needs. An LLM might not provide accurate answers if it hasn't been trained on the company's data or if the information required is recent and not included in the model's training [7, 19]. These limitations make it difficult for LLMs to deliver precise, up-to-date, and context-specific responses.

This is where Retrieval-Augmented Generation (RAG) proves vital. RAG enables LLMs to utilize external sources, bridging the gap between the model's training limitations and the need for accurate, customized responses [1, 7, 16, 24]. By leveraging RAG, LLMs can access private or latest data, ensuring more relevant and precise outputs. Private data refers to information specific to an organization or domain that the LLM might not have encountered during its training. Similarly, the latest data includes updates or developments that occurred after the LLM was trained [6, 11, 22].

RAG is a hybrid approach that combines the strengths of two types of models: retrieval models and generative models [8, 9, 14, 17]. Retrieval models locate and retrieve pertinent material from outside sources, like text collections or databases. Generative models then synthesize and present the retrieved information as coherent, contextually appropriate responses. This synergy allows RAG to enhance the accuracy and relevance of responses, particularly in scenarios where up-to-date knowledge or domain-specific expertise is critical [16, 18, 27].

Retrieval-Augmented Generation is a revolutionary development in natural language processing, to sum up. RAG overcomes the drawbacks of conventional LLMs by allowing LLMs to use private and current data, making it an effective tool for providing accurate and context-aware replies. [7, 16, 23].

Scope of the Paper:

This paper reviews several published works to gain a deeper understanding of Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs). Additionally, it serves as a foundational resource for developing a model that leverages RAG to deliver specific, accurate and up-to-date educational content. By harnessing the vast reservoir of educational information, this model aims to revolutionize the creation of tailored and reliable educational materials.

II. Objectives of the Review:

The overall purpose of this critical review is to systematically review the strength, weakness, and evolving nature of Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG). This study aims to:

- Examine the architectural building blocks and functioning of LLMs and RAG systems.
- Review the strengths and weaknesses of LLMs, particularly with respect to factual factuality, domain adaptation, and real-time information access.
- Discuss how RAG enhances LLMs by applying external knowledge retrieval mechanisms to generate contextually responsive and accurate content.
- Provide insights into today's trends in research, applications in the real-world, and where the area is headed with respect to the integration of retrieval and generation models.
- Lastly, this review seeks to provide a comprehensive and balanced contribution towards the knowledge of how RAG improves LLMs in overcoming traditional challenges and advancing the area of intelligent information processing.

III. Literature Review

3.1 Review on Large Language Models (LLMs):

A subset of artificial intelligence models called Large Language Models (LLMs) are made to process and produce text that is similar to that of a human (Vaswani et al., 2017, Devlin et al., 2018, Brown et al., 2020). These models, which are constructed using deep learning techniques and trained on large datasets, are able to comprehend and generate language that is coherent and pertinent to the situation. OpenAI's GPT series, Google's BERT (Devlin et al., 2018), and Meta's LLaMA are well-known instances of LLMs.

LLMs have been transformative in natural language processing (NLP), powering applications like chatbots, language translation, sentiment analysis and content generation (Brown et al., 2020, Goyal et al, 2018). They are capable of understanding context, answering questions, summarizing text and even generating creative outputs like poetry or stories.

Over the past three years, NLP has advanced rapidly with the creation of increasingly large language models like BERT, GPT-2/3, and Switch-C, which have improved performance on many English-language tasks (Bender et al., 2021). These improvements have come through both new architectures and sheer model size, often achieved by fine-tuning pretrained models for specific tasks.

But this raises concerns about whether larger is always better and draws attention to possible hazards including financial and environmental consequences. It suggests giving curated datasets top priority, coordinating development with stakeholder values and research objectives, and looking into options other than merely creating bigger models (Bender et al., 2021).

Limitations of LLMs:

Despite their capabilities, LLMs face several critical limitations:

- **Lack of Domain-Specific Knowledge:**
  LLMs are trained on general datasets and may not perform well when unique or specialized knowledge is required, such as industry-specific jargon or secret information (Zhou et al., 2022).
- **Outdated Information:**
  Since LLMs are trained on data available up to a certain point in time, they lack real-time awareness of recent events or updates. For example, an LLM trained in 2023 won't have information on developments from 2024 or later.
- **Private Data Unavailability:**
  LLMs cannot access confidential or private data unless explicitly provided, which limits their effectiveness in applications requiring proprietary insights (Zhou et al., 2022).
- **Hallucination of Facts:**
  LLMs may generate outputs that sound plausible but are factually incorrect, especially when confronted with unfamiliar or incomplete information (Lin et al., 2021).
- **Scalability Concerns:**
  Training LLMs requires vast computational resources and energy, making them expensive to scale and update.
- **Ethical and Bias Issues:**
  The training data for LLMs often reflects biases present in society, leading to outputs that may inadvertently perpetuate stereotypes or discriminatory views (Bender et al., 2021, Mengru Wang et al, 2024).

By combining retrieval and generation, RAG acts as a bridge between LLMs and the dynamic, context-specific data needed for real-world applications. This hybrid approach enables LLMs to overcome their static nature, making them more adaptable, accurate, and relevant for diverse use cases (Patrick Lewis et al. 2020, Tom B. Brown et al. 2020)

3.2 Review on Retrieval-Augmented Generation (RAG)

Key Components of RAG:

- **Retrieval:**
  Searches external knowledge sources for information relevant to the user query (Kelvin Guu et al, 2020, Gautier Izacard and Edouard Grave, 2021).
- **Augmentation:**
  Combines retrieved information with the user query to create an enriched input for the generative model.
- **Generation:**
  Produces a detailed and contextually accurate response using the augmented input.

- **Response Delivery:**
  Outputs the response to the user, ensuring the information is precise and personalized (Patrick Lewis et al., 2020, Parvez et al., 2021).

RAG is particularly effective for tasks requiring access to specific, private, or up-to-date data. It allows LLMs to generate responses that are not only contextually accurate but also tailored to domain-specific needs, making it an excellent solution for business chatbots, customer service tools and knowledge-based applications (Vladimir Karpukhin et al, 2020).

Implementing a RAG system in production is far more complex than creating a simple demo, like a local chat-to-PDF tool. While demos might be easy to set up, deploying a fully functional RAG system for a business involves multiple intricate steps, each with its own set of challenges and techniques. This complexity makes the process demanding and often uncomfortable.

Paper by Xiaohua Wang et al, 2024, explores optimal practices for implementing RAG and introduces some best practice recommendations. While these suggestions are valuable and based on the authors' experiments and experiences, they don't have to be followed rigidly. Instead, they serve as a helpful roadmap, offering approaches that can be adapted to suit specific needs and circumstances.

Various RAG approaches aim to enhance large language models (LLMs) by using query-dependent retrievals (Yunfan Gao et al, Rev 2024, Huayang Li et al, 2022, Deng Cai et al., 2022). A typical RAG workflow (Xiaohua Wang et al, 2024) involves several key steps:

- **Query Classification**: Decides if retrieval is needed for the given input query.
- **Retrieval**: Efficiently fetches relevant documents for the query.
- **Reranking**: Refines the order of retrieved documents based on their relevance.
- **Repacking**: Organizes retrieved documents into a structured format for better generation.
- **Summarization**: Extracts key information while removing redundancies to aid response generation.

In addition to these steps, implementing RAG also requires:

- **Document Chunking**: Properly splitting documents into manageable pieces.
- **Embeddings**: Choosing suitable methods for semantically representing chunks.
- **Vector Databases**: Selecting efficient storage for feature representations.
- **LLM Fine-Tuning**: Optimizing the model for improved performance.

RAG Workflow and Components

Implementing a RAG system involves multiple complex steps, each with its own techniques and challenges. The authors of the paper (Xiaohua Wang et al, 2024) experimented with different methods for each component and evaluated their impact on RAG performance. They tested up to three methods per step, keeping other components constant, and selected the best-performing method for the next step.

1. **Chunking**

Chunking splits long documents into smaller pieces to fit the LLM's context window. Common techniques include:

- **Token-Level Chunking:** Splitting at the token level (similar to words).
- **Semantic-Level Chunking**: Grouping semantically similar paragraphs or sentences.
- **Sentence-Level Chunking**: Splitting by sentences for a balance between context and simplicity.

The study done by Xiaohua Wang et al, 2024 uses sentence-level chunking which suited most.

Chunk Size:

Chunk size plays a crucial role in performance. Larger chunks offer more context, improving comprehension, but they increase processing time. Smaller chunks are faster and boost retrieval recall but might lack enough context.

Finding the right chunk size requires balancing key metrics:

- **Faithfulness**: Ensures the response aligns with the retrieved text and avoids hallucination.
- **Relevancy**: Checks if the retrieved text and response match the query effectively.

Chunking Techniques.

Advanced techniques like small-to-big and sliding window enhance retrieval quality by managing the relationships between chunk blocks. Small-sized chunks are used to match queries, while larger blocks that contain the small ones along with additional context are returned.

To demonstrate these techniques, **Xiaohua Wang et al, 2024** used the LLM-Embedder (Peitian Zhang et al., 2023) model for embedding. The chunk sizes were set at 175 tokens for smaller chunks and 512 tokens for larger ones, with a 20-token overlap. These methods improve retrieval by preserving context and ensuring relevant information is retrieved.

2. **Vector Databases**

Vector databases store embeddings and help retrieve relevant data. The authors (Xiaohua Wang et al, 2024) evaluated various options based on features like scalability, indexing types, and cloud capabilities. They chose **Milvus** for its balance of flexibility, scalability, and ease of use.

3. **Query Classification**

Queries are classified as:

- **Sufficient**: The LLM can answer without retrieval.
- **Insufficient**: Requires retrieval from external sources.

They used a BERT-based classifier for this step, which improved accuracy and recall.

4. **Retrieval Methods**

Retrieving the most relevant context involves techniques like:

- **Query Rewriting**: Adjusting queries for precision.
- **Query Expansion**: Adding synonyms for broader recall.
- **Query Decomposition**: Breaking queries into subqueries.

- **Pseudo-Document Generation**: Creating imaginary documents to find similar real documents in the database.

5. **Summarization**

Retrieved documents can be passed to the LLM as-is or summarized.

- **Extractive Summarization:** Selects key sentences from documents.
- **Abstractive Summarization**: Synthesizes summaries using new words and sentences.
  Techniques like ReComp and Selective Context were tested.

6. **Fine-Tuning the Generator**

Fine-tuning the generative model improves response accuracy. The study found fine-tuned models performed significantly better than baseline LLMs.

The authors (Xiaohua Wang et al, 2024) documented the impact of each method on performance using metrics like faithfulness and relevancy. Their findings highlight the importance of balancing precision, recall, and computational efficiency for optimal RAG implementation.

The workflow and the techniques discussed above serve as a guide for developing robust RAG systems in production.

3.3 Review on Natural Language Processing and its limitations

An important advancement in artificial intelligence and natural language processing (NLP) is represented by pre-trained neural language models. Within their constraints, these models—including GPT, BERT, and others—have the capacity to learn and retain vast amounts of information. They serve as parameterised implicit knowledge bases since they are trained on large datasets; that is, the knowledge they learn during training is stored in the model and is accessible during inference (Patrick Lewis et al, 2020).

One of the most remarkable aspects of these models is their ability to perform various tasks without requiring external memory. They excel in tasks such as language generation, summarization and answering questions by relying solely on their internalized knowledge. This ability eliminates the need for explicit, external databases or knowledge sources in many applications, making these models efficient and powerful tools for a wide range of NLP tasks.

Despite their capabilities, these models exhibit several limitations. A major drawback is their inability to easily revise or expand their internalized knowledge. Once trained, their memory is effectively "frozen," and updating or adding new information requires extensive retraining, which is resource-intensive (Patrick Lewis et al, 2020).

Additionally, these models often struggle with interpretability. When they produce outputs, they cannot easily explain the reasoning behind their predictions or decisions. This opaqueness can lead to challenges in critical applications where understanding the basis of a model's response is essential, such as healthcare or legal systems.

Perhaps the most concerning limitation is their tendency to generate hallucinations—outputs that are plausible-sounding but factually incorrect. These hallucinations occur because the models may confidently generate responses based on incomplete or misunderstood information, leading to misinformation.

To mitigate these issues, researchers have explored hybrid models that combine the strengths of pre-trained models with external, non-parametric memory systems. Non-parametric memory refers to knowledge storage outside the model's parameters, such as databases or retrieval systems, which can be dynamically accessed during inference. By integrating parametric and non-parametric memory, hybrid models offer several advantages.

Knowledge stored in external memory can be directly revised or expanded without retraining the entire model. Moreover, the retrieved information can be inspected, verified and interpreted, making the models more transparent and reliable.

Two notable examples of hybrid models are REALM (Retrieval-Augmented Language Model) and ORQA (Open-Retrieval Question Answering). These models combine masked language models with a differentiable retriever, a mechanism that retrieves relevant information from external memory during inference. REALM and ORQA have demonstrated promising results in tasks like open-domain extractive question answering, where answers to questions are pulled from a large corpus of text (Patrick et al., 2020, Kelvin Guu et al., 2020, Kenton Lee et al., 2019).

Retrieval-Augmented Generation (RAG) is indeed a crucial approach for enterprise companies leveraging AI, as it ties directly to generating measurable business value by enhancing information retrieval and decision-making capabilities. Improving RAG involves innovations in retrieval techniques, indexing, and context-awareness, all of which can directly enhance system efficiency, user satisfaction, and ultimately, revenue (Antropic, 2024).

Key Insights:

- **Direct Business Value**: Enterprises invest in RAG because of its clear ROI. Improvements in retrieval accuracy and relevance directly translate to better customer experiences, faster insights, and reduced operational costs.
- **Contextual Retrieval by Anthropic**: This newly introduced technique focuses on enhancing the efficiency of RAG by making retrieval more context-aware. While it may carry marketing intentions as an upsell by Anthropic, it highlights real innovation that could yield tangible performance improvements.
- **Impact on Internal Systems**: For internal search systems or chatbots deployed within companies:
  1. Enhanced retrieval techniques improve response accuracy, leading to better user engagement and satisfaction.
  2. Improvements in metrics like precision, recall, and latency create a competitive advantage by enabling faster decision-making and reducing friction in workflows.
- **Simplicity and Effectiveness**: While terms like "contextual retrieval" sound complex, Anthropic's (Anthropic, 2024) emphasis on simplicity and measurable improvement makes this innovation accessible. Companies can adopt these methods without heavy technical overhauls, making it a practical solution.
- **Future Potential**: Organizations adopting advanced RAG techniques like contextual retrieval stand to gain significantly in customer-facing and internal applications. This could include better customer support, advanced knowledge management, and even predictive analytics.

If someone is considering implementing or improving RAG systems, it would be beneficial to explore contextual retrieval methods and evaluate the measurable gains one can bring to their use case.

3.4 Review on LongRAG

The introduction of **LongRAG** presents a significant advancement in the domain of Retrieval-Augmented Generation (RAG), combining the strengths of long-context Large Language Models (LLMs) with enhanced retrieval techniques. This approach not only improves performance on key benchmarks but also addresses critical challenges associated with traditional RAG systems.

Instead of retrieving many small units (e.g., individual documents or paragraphs), LongRAG operates on larger, aggregated retrieval units. This reduces the number of retrievals while maintaining or improving the overall relevance of the retrieved content. By leveraging the advanced zero-shot capabilities of long-context LLMs like GPT-4, LongRAG enables accurate answer extraction without requiring extensive fine-tuning. This simplifies deployment and adaptability across different domains.

By reducing the number of retrieval units, LongRAG optimizes computational resources while enhancing recall. This dual benefit is particularly valuable for large-scale applications where efficiency and performance are critical. LongRAG exemplifies how seemingly competing concepts—retrieval optimization and long-context LLMs—can be harmonized to create a more effective system. It challenges the misconception that long-context LLMs and retrieval systems must operate independently.

LongRAG's capability to handle long-context reasoning and retrieve fewer, more relevant units can transform workflows across various domains. Here's how it could be applied to key areas such as Legal and Compliance Document Analysis (Zhao et al., 2024), Customer Support Automation (Zhao et al., 2024), Academic Research and Summarization (Jiang et al., 2024), Healthcare Knowledge Management (Zhao et al., 2024), Technical Documentation Support (Jiang et al., 2024), Historical and Trend Analysis (Zhao et al., 2024) etc.

LongRAG represents a practical and scalable enhancement for industries reliant on large, unstructured datasets. By merging retrieval efficiency with the contextual reasoning of LLMs, it has the potential to redefine workflows and elevate decision-making (Zhao et al., 2024, Jiang et al., 2024).

Large Language Models (LLMs) are increasingly used in tasks like **workflow automation**, where they generate structured outputs (e.g., JSON workflows) from natural language inputs. However, hallucinations—where models fabricate non-existent elements—pose significant challenges, especially in high-stakes scenarios. The paper by Bechard and Ayala, 2024 introduces a **RAG-based approach** to mitigate hallucination by embedding **retrievers** into the generation process.

3.5 Review on Generative AI Capabilities Through Retrieval-Augmented Generation Systems and LLMs

The paper by Bansal and Suddala (2024) likely discusses the integration of Retrieval-Augmented Generation (RAG) techniques with Large Language Models (LLMs) to address the limitations inherent in Generative AI. While traditional LLMs are highly capable, they often produce generic answers, generate false information and lack specificity. RAG mitigates these issues by combining LLMs with retrieval systems that fetch relevant information from external sources, ensuring more accurate and reliable responses while reducing computational overhead.

The applications of RAG are vast, including enhancing the performance of chatbots, search engines, and research tools by providing contextually relevant and precise information. Additionally, the approach is scalable, capable of handling large datasets and complex queries, and offers the potential for integrating real-time data and multimodal inputs. The paper by Bansal and Suddala (2024) positions RAG as a transformative technique for advancing the capabilities and reliability of generative AI systems.

Despite the rapid advancements in generative artificial intelligence (AI), critiques surrounding its limitations have grown. These include evidence of various biases—such as gender and racial biases—as well as studies highlighting the limited ability of generative AI systems to interpret statements, follow complex prompts, and distinguish accurate from inaccurate information. Many of these limitations originate from the inner workings of generative AI models, which primarily rely on patterns in the language they were trained on. Rather than querying for real-world information, these models generate responses based on statistical patterns, reflecting the latent design choices of practitioners in the field (Bansal & Suddala, 2024).

While generative systems excel at producing novel and coherent language from a statistical perspective, they often lack a direct connection to real-world information, which is essential for providing accurate, relevant, and accessible outputs. To address these shortcomings, Bansal and Suddala (2024) proposed a Retrieval-Augmented Generation (RAG) approach and integrate it with various off-the-shelf generative models. Their method begun by retrieving a set of relevant passages containing the necessary information through a retrieval model. These passages are then incorporated into the generation process, reducing the likelihood of hallucinated responses. By bridging generative AI with traditional information retrieval frameworks, RAG effectively mitigates the lack of relevance and inability of generative models to integrate real-world knowledge directly. The results of the experiments demonstrated that incorporating retrieved passages significantly improves response quality, even when the retrieval process is imperfect. Moreover, RAG consistently outperforms simple answer-aggregation methods. User studies further confirm that human evaluators find RAG-generated outputs to be more accurate and human-like, underscoring its potential to enhance the reliability and contextual relevance of generative AI systems.

IV. TABLE: SUMMARY OF ANALYSIS ON LLMS, RAG, NLP LIMITATIONS, AND LONGRAG

Topic	Key Insights	Strengths	Limitations	References
NLP and Its Limitations	NLP is the basis of current AI language systems but is plagued by problems such as vagueness, domain constraints, and insufficiency of depth in context.	Effective for simple operations such as sentiment analysis, translation, and summarization.	Fails at context preservation, sarcasm, domain specificity, and real-time flexibility.	Bender et al., 2021 [4]; Lin et al., 2021 [5]; Goyal et al., 2018 [13]
Large Language	LLMs like GPT and BERT employ deep learning	Extremely generalizable, with few-shot and zero-shot	Do not have access to real-time and	Devlin et al., 2018 [3]; Brown et al.,

Models	to comprehend and produce human-like language from enormous data sets.	learning capabilities.	domain knowledge; susceptible to hallucinations.	2020 [2]; Vaswani et al., 2017 [12]
Retrieval-Augmented Generation	RAG improves LLMs by retrieving relevant external documents to anchor generation in factual and up-to-date content.	Enhances factual accuracy and response customizability.	Is dependent on retrieval quality; computationally more expensive than pure LLMs.	Lewis et al., 2020 [7]; Guu et al., 2020 [8]; Karpukhin et al., 2020 [9]
LongRAG	LongRAG is a sophisticated variant of RAG designed for long-context retrieval and reasoning, for longer documents and dialogues.	Processes large documents and multi-turn queries better with higher relevance and depth.	Currently still in research; performance is mixed by domain and retrieval source.	Jiang et al., 2024 [22]; Zhao et al., 2024 [23]
Generative AI Capabilities Through RAG and LLM Integration	Merging generative models with retrieval systems results in stronger AI able to respond to sophisticated, context-dependent, and domain-sensitive questions.	Strikes a balance between creativity and precision; is highly scalable for enterprise use.	Complexity of integration, external data sources' privacy concerns, and response generation latency.	Bansal & Suddala, 2024 [26]; Izacard et al., 2022 [25]; Cai et al., 2022 [18]

IV. CONCLUSION

Retrieval-Augmented Generation (RAG) enhances language models, such as GPT-4, by integrating relevant knowledge into interactions. This approach is critical for reducing **hallucinations**—responses that are not grounded in real data or logic. By providing additional, relevant context, RAG ensures the model's outputs are more reliable and accurate. For example, if we are building an AI tutorial using RAG, by leveraging this approach, we can regularly update the database with the latest information and provide the model access to updated knowledge without requiring retraining. This ensures that the responses stay current and grounded in up-to-date, factual data.

This paper aims to identify optimal practices for implementing Retrieval-Augmented Generation (RAG) to enhance the quality and reliability of content produced by large language models (LLMs). The author systematically assessed various solutions for each module within the RAG framework and recommended the most effective approaches for each. It is also learnt from few papers that few authors introduced a comprehensive evaluation benchmark for RAG systems. Through extensive experiments, they identified best practices and compared different alternatives, providing valuable insights into improving RAG's effectiveness. Long RAG was also studied

The findings from this study contribute to a deeper understanding of RAG systems and lay a solid foundation for future research and advancements in this field.

This paper also studies the Long RAG framework, a novel approach designed to address imbalances in retrieval tasks. This paper also discusses about key benefits of Long RAG. The paper studied also addresses a common limitation of traditional LLMs: their tendency to generate false or incomplete information.

While traditional LLMs sometimes generate inaccurate answers or skip critical details, RAG resolves these issues by integrating LLMs with external knowledge sources. This combination ensures that responses are more accurate, comprehensive and reliable.

REFERENCES

[1] Wang, X., et al. 2024. Searching for Best Practices in Retrieval-Augmented Generation. *arXiv preprint*, arXiv:2407.01219.

[2] Brown, T. B., et al. 2020. Language Models are Few-Shot Learners. arXiv preprint, arXiv:2005.14165.

[3] Devlin, J., et al. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint*, arXiv:1810.04805.

[4] Bender, E. M., et al. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? *Communications of the ACM*, 64(10): 62–71.

[5] Lin, S., et al. 2021. TruthfulQA: Measuring How Models Mimic Human Falsehoods. *arXiv preprint*, arXiv:2109.07958.

[6] Zhou, X., et al. 2022. Knowledge Augmented Language Models: A Survey of Methods and Challenges. *arXiv preprint*, arXiv:2202.05026.

[7] Lewis, P., et al. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv preprint*, arXiv:2005.11401.

[8] Guu, K., et al. 2020. REALM: Retrieval-Augmented Language Model Pre-Training. *arXiv preprint*, arXiv:2002.08909.

[9] Karpukhin, V., et al. 2020. Dense Passage Retrieval for Open-Domain Question Answering. *arXiv preprint*, arXiv:2004.04906.

[10] Brown, T. B., et al. 2020. Language Models are Few-Shot Learners. *arXiv preprint*, arXiv:2005.14165.

[11] Wang, M., et al. 2024. Knowledge Mechanisms in Large Language Models: A Survey and Perspective. *arXiv preprint*, arXiv:2407.15017.
[12] Vaswani, A., et al. 2017. Attention Is All You Need. *arXiv preprint*, arXiv:1706.03762.
[13] Goyal, A., et al. 2018. Deep Learning for Natural Language Processing: Creating Neural Networks with Python. Apress Media LLC, ISBN: 978-1-4842-3684-0.
[14] Izacard, G. and Grave, E. 2021. Leveraging Passage Retrieval with Generative Models for Open-Domain Question Answering. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, 874–880.
[15] Parvez, M. S., et al. 2021. Retrieval Augmented Code Generation and Summarization. *arXiv preprint*, arXiv:2108.11601.
[16] Gao, Y., et al. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint*, arXiv:2312.10997.
[17] Li, H., et al. 2022. A Survey on Retrieval-Augmented Text Generation. *arXiv preprint*, arXiv:2202.01110.
[18] Cai, D., et al. 2022. Recent Advances in Retrieval-Augmented Text Generation. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 3417–3419.
[19] Zhang, P., et al. 2023. Retrieve Anything to Augment Large Language Models. *arXiv preprint*, arXiv:2310.07554.
[20] Lee, K., et al. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. *arXiv preprint*, arXiv:1906.00300.
[21] Anthropic. 2024. Contextual Retrieval. Retrieved on 18th January, 2025 from https://www.anthropic.com/news/contextual-retrieval.
[22] Jiang, M., et al. 2024. LongRAG: Enhancing Retrieval-Augmented Generation with Long-context LLMs. *arXiv preprint*, arXiv:2406.15319.
[23] Zhao, J., et al. 2024. LongRAG: A Dual-Perspective Retrieval-Augmented Generation Paradigm for Long-Context Question Answering. *Proceedings of EMNLP 2024*, Paper 1259.
[24] Bechard, S. and Ayala, M. 2024. Reducing Hallucination in Structured Outputs via Retrieval-Augmented Generation. *arXiv preprint*, arXiv:2404.08189.
[25] Izacard, G., et al. 2022. Atlas: Few-shot Learning with Retrieval Augmented Language Models. *arXiv preprint*, arXiv:2208.03299.
[26] Bansal, A. and Suddala, S. 2024. Enhancing Generative AI Capabilities Through Retrieval-Augmented Generation Systems and LLMs. *Library Progress International*, 44(3): 17765–17775.
[27] Borgeaud, S., et al. 2022. Improving Language Models by Retrieving from Trillions of Tokens. *arXiv preprint*, arXiv:2112.04426.
[28] Jozefowicz, R., et al. 2016. Exploring the Limits of Language Modeling. *arXiv preprint*, arXiv:1602.02410.
[29] Vaswani, A., et al. 2017. Attention is All You Need. *Advances in Neural Information Processing Systems*, https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
[30] Brown, T. B., et al. 2020. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.