INTERFACE TECNOLÓGICA

ARQUITETURA RAG E AGENTES

RAG ARCHITECTURE AND AGENTS

Antonio Henrique Nascimento Machado de Souza – antonio.souza58@fatec.sp.gov.br
Faculdade de Tecnologia de Praia Grande – Praia Grande– São Paulo – Brasil

Isadora Mota de Souza – isadora.souza4@fatec.sp.gov.br
Faculdade de Tecnologia de Praia Grande – Praia Grande– São Paulo – Brasil

Jônatas Cerqueira Dias – jonatas.dias2@fatec.sp.gov.br
Faculdade de Tecnologia de Praia Grande – Praia Grande– São Paulo – Brasil

DOI: 10.31510/infa.v21i2.1995
Data de submissão: 13/08/2024
Data do aceite: 23/11/2024
Data da publicação: 20/12/2024

RESUMO

Este artigo propõe uma pesquisa bibliográfica com o intuito de fornecer uma visão abrangente sobre a Arquitetura RAG (*Retrieval-Augmented Generation*) e a aplicação de agentes no contexto atual da Inteligência Artificial (IA). A arquitetura RAG combina técnicas de recuperação de informações com modelos de geração, proporcionando uma abordagem híbrida que melhora a eficiência e a precisão das respostas geradas por sistemas de IA. O planejamento adequado da arquitetura é crucial para o sucesso dos sistemas baseados em RAG, pois afeta diretamente a capacidade dos agentes de lidar com grandes volumes de dados e fornecer respostas relevantes. Neste artigo, os princípios fundamentais da Arquitetura RAG são explorados. Além disso, foi realizada uma análise crítica das contribuições e tendências emergentes na literatura sobre o tema. Os resultados da pesquisa indicam que, embora a Arquitetura RAG ofereça avanços significativos na interação entre agentes e dados, ainda existem desafios importantes a serem superados para sua adoção em larga escala. Este estudo busca contribuir para o entendimento desses desafios e fornecer insights para futuras pesquisas e implementações.

Palavras-chave: Arquitetura RAG. Agentes. ReAct. Inteligência Artificial. LLM.

ABSTRACT

This article proposes a bibliographical survey to provide a comprehensive overview of the RAG (Retrieval-Augmented Generation) Architecture and the application of agents in the current context of artificial intelligence (AI). The RAG architecture combines information retrieval techniques with generation models, providing a hybrid approach that improves the efficiency and accuracy of responses generated by AI systems. Proper architecture planning is crucial to the success of RAG-based systems, as it directly affects the agents' ability to handle large volumes of data and provide relevant answers. In this article, the fundamental principles of the RAG Architecture are explored, as well as its practical applications, advantages,

disadvantages, and implementation challenges. In addition, a critical analysis of the contributions and emerging trends in the literature on the subject was carried out. The results of the research indicate that, although the RAG Architecture offers significant advances in the interaction between agents and data, there are still important challenges to be overcome for its large-scale adoption. This study seeks to contribute to understanding these challenges and provide insights for future research and implementation.

Keywords: RAG architecture. Agents. ReAct. Artificial intelligence. LLM.

1 INTRODUÇÃO

As técnicas empregadas na evolução dos sistemas de IA têm avançado de forma notável, impulsionadas pela crescente demanda por soluções mais eficientes e precisas. A Arquitetura RAG emergiu como uma abordagem promissora para superar as limitações dos modelos tradicionais de IA, ao combinar a recuperação de informações com a geração de linguagem. Essa arquitetura híbrida oferece um meio eficaz de melhorar a relevância e a precisão das respostas geradas, especialmente em contextos onde o volume de dados é grande e as respostas precisam ser contextualmente precisas (FINARDI et al., 2024). Embora a Arquitetura RAG traga avanços significativos, ainda enfrenta desafios importantes, como a necessidade de uma infraestrutura robusta e o manejo eficiente dos custos de processamento.

Com o aumento da complexidade dos sistemas de IA, a integração de técnicas de recuperação e geração tornou-se essencial para atender às exigências de aplicações que demandam respostas rápidas e precisas (GAO et al., 2023; LEWIS et al., 2020). O objetivo deste artigo é investigar em profundidade os princípios fundamentais da Arquitetura RAG e sua aplicação no desenvolvimento de agentes inteligentes, bem como os desafios e oportunidades que surgem na sua implementação em larga escala.

2 REFERENCIAL TEÓRICO

Nesta seção, é realizada uma revisão da literatura existente, com o objetivo de fundamentar teoricamente o presente estudo sobre Arquitetura RAG. As obras citadas fornecem o embasamento necessário para a compreensão dos conceitos-chave e das práticas envolvidas na Arquitetura RAG e demais tópicos pertinentes.

2.1 Inteligência Artificial

A IA é um campo vasto e dinâmico que tem se expandido rapidamente, abordando uma ampla gama de técnicas e aplicações que visam emular processos cognitivos humanos,

como reconhecimento de padrões, tomada de decisões e aprendizado. Andrew Ng, um dos principais pesquisadores na área de IA e cofundador do Google Brain, enfatiza que a IA tem o potencial de transformar indústrias inteiras, comparando seu impacto ao da eletricidade durante a Revolução Industrial. A revolução gerada pela IA, pode ser comparada com o surgimento da eletricidade, pois, assim como a eletricidade revolucionou múltiplos setores, a IA tem o potencial de ser uma força motriz para a inovação em diversas áreas, desde a saúde até a agricultura (NG, 2016).

2.2 Modelos de Linguagem de Grande Escala

Os Modelos de Linguagem de Grande Escala (LLMs) têm se tornado uma parte central no campo da Inteligência Artificial, transformando a maneira como interagimos com máquinas e processamos informações textuais. LLMs são redes neurais profundas treinadas em enormes quantidades de dados textuais para prever a próxima palavra em uma sequência, o que lhes permite gerar textos coerentes e responder a perguntas de maneira contextualizada. Esses modelos são conhecidos por sua capacidade de realizar uma variedade de tarefas de processamento de linguagem natural (NLP), desde tradução automática até redação de textos complexos (BROWN et al., 2020; GAO et al., 2023; TOUVRON et al., 2023).

Um dos LLMs mais conhecidos é o ChatGPT, desenvolvido pela OpenAI. O ChatGPT é baseado na arquitetura GPT (Generative Pre-trained Transformer) e foi projetado para gerar texto em linguagem natural de alta qualidade (BROWN et al., 2020).

Os modelos possuem cobranças pela quantidade de entradas de tokens e de tokens gerados, dessa forma, os tokens podem ser classificados como unidades básicas de texto que um modelo de linguagem usa para processar e gerar linguagem natural. Em termos simples, um token pode ser uma palavra, parte de uma palavra ou até mesmo um caractere, dependendo do modelo e da sua configuração (FINARDI et al., 2024).

Outro modelo notável é o LLaMA (Large Language Model Meta AI), desenvolvido pela Meta. O LLaMA é uma iniciativa para criar um modelo de linguagem eficiente em termos de recursos computacionais, oferecendo um desempenho competitivo em relação a outros LLMs maiores (TOUVRON et al., 2023). O LLaMA é escalável, sendo assim, uma característica distinta do LLaMA é que ele é open-source, permitindo que pesquisadores e desenvolvedores tenham acesso ao modelo treinado, o que promove a colaboração e a inovação na comunidade de IA.

Os LLMs têm implicações profundas para o futuro da IA e da sociedade em geral. Embora tenham mostrado capacidades impressionantes, questões éticas e técnicas ainda precisam ser abordadas, como a mitigação de vieses, a transparência dos modelos, e o controle sobre o uso dessas tecnologias(BOSTROM; YUDKOWSKY, 2014). À medida que esses modelos continuam a evoluir, o campo de LLMs promete transformar ainda mais a interface entre humanos e máquinas, criando oportunidades e novos desafios.

2.2.1 Parâmetros em LLMs

Os parâmetros de um modelo de linguagem são os valores ajustáveis que o modelo usa para aprender e fazer previsões, como pesos e vieses em redes neurais, que definem a força das conexões entre neurônios. Durante o treinamento, o modelo ajusta esses parâmetros para minimizar erros e melhorar a precisão das previsões ou geração de texto, com base nos dados recebidos. Modelos com mais parâmetros tendem a ser mais complexos e precisos, mas também requerem maior custo computacional e mais dados. Assim, a escolha do número de parâmetros envolve equilibrar a capacidade do modelo com a eficiência computacional (BROWN et al., 2020; GOODFELLOW; BENGIO; COURVILLE, 2016; TOUVRON et al., 2023).

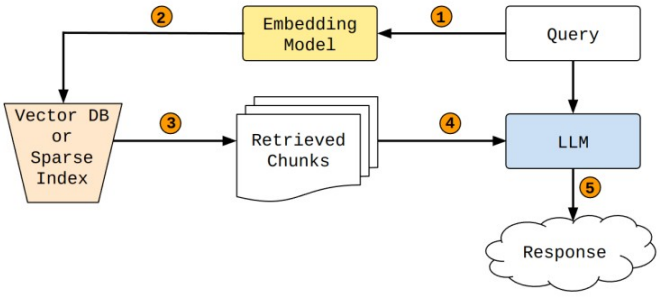
2.3 Arquitetura RAG

A arquitetura RAG representa uma abordagem inovadora no campo de modelos de linguagem, combinando técnicas de recuperação de informações com a geração de texto para criar sistemas mais precisos e informados. RAG integra um modelo de recuperação (Retriever) que busca informações relevantes em uma base de dados externa e um modelo gerador (Generator) que utiliza essas informações para produzir respostas ou conteúdos (FINARDI et al., 2024; GAO et al., 2023; LEWIS et al., 2020). Esse paradigma é particularmente eficaz em tarefas que requerem conhecimentos específicos e atualizado, como sistemas de perguntas e respostas, onde a precisão e a relevância da informação são cruciais.

A arquitetura RAG é composta por três componentes principais: o Retriever, o Chunk e o Generator. O Retriever é responsável por localizar e extrair passagens relevantes de uma base de conhecimento pré-existente, que podem incluir documentos, artigos ou qualquer outro tipo de texto estruturado (KARPUKHIN et al., 2020). O Chunk refere-se ao processo de segmentação do conteúdo recuperado em partes menores e mais gerenciáveis, que são, então,

fornecidas ao Generator. O Generator, utilizando algum LLM, sintetiza essas informações em respostas coerentes e contextualmente apropriadas.

Figura 1 – Estrutura simples RAG



Fonte: FINARDI et al. (2024)

A principal vantagem da arquitetura RAG é sua capacidade de combinar a flexibilidade dos modelos de geração de linguagem com a precisão dos métodos de recuperação de informações. A arquitetura faz uso de chunks, ou segmentos menores de texto, permitindo uma integração mais eficaz entre os dados recuperados e a geração de respostas, resultando em uma melhoria significativa na qualidade e relevância do conteúdo gerado (FINARDI et al., 2024; GAO et al., 2023; YAO et al., 2022).

Apesar de seus benefícios, a arquitetura RAG também enfrenta desafios, como a necessidade de volumes de dados para gerar a base de conhecimento no banco de dados vetorial. Além disso, a integração entre os componentes pode ser complexa, exigindo ajustes finos para garantir que o conteúdo recuperado seja adequadamente segmentado e utilizado pelo modelo gerador.

2.3.1 Evolução do RAG

A evolução da Arquitetura RAG foi impulsionada por inovações como o Dense Passage Retrieval (DPR), que aprimorou a recuperação de dados com embeddings densos (KARPUKHIN et al., 2020). A integração com modelos generativos de grande escala, como destacado por Gao et al. (2023), ampliou sua aplicação em tarefas de linguagem natural, melhorando a geração de respostas baseadas em grandes volumes de dados (LEWIS et al., 2020).

2.3.2 Banco de dados vetorial

Bancos de dados vetoriais são projetados para gerenciar dados que são representados por vetores multidimensionais, como embeddings gerados por modelos de aprendizado profundo (GAO et al., 2023; JOHNSON; DOUZE; JÉGOU, 2017; WANG et al., 2021).

Embeddings são representações densas de dados, como palavras, frases, imagens ou outros tipos de informações, em um espaço vetorial. Esses vetores capturam semântica e contexto de forma que facilita a análise e a busca por similaridades, otimizando a consulta e a busca por vetores semelhantes, utilizando técnicas avançadas de indexação e pesquisa (JOHNSON; DOUZE; JÉGOU, 2017; KARPUKHIN et al., 2020).

2.4 ReAct e Agentes

No campo da IA, um agente é um sistema projetado para observar seu ambiente, tomar decisões e executar ações com base nessas decisões. Os agentes são fundamentais para criar sistemas inteligentes que podem operar de maneira independente e adaptar seu comportamento a situações variáveis (RUSSEL; NORVIG, 2016).

Uma abordagem inovadora para aprimorar o desempenho dos agentes é a ReAct (*Reasoning and Acting*), que integra raciocínio e ação em um ciclo contínuo. Tradicionalmente, os modelos de agentes tratam o raciocínio e a ação como processos separados, o que pode limitar a eficácia em cenários dinâmicos. A ReAct supera essa limitação ao combinar raciocínio lógico com a execução de ações, permitindo que os agentes ajustem suas ações com base em raciocínios atualizados em tempo real (YAO et al., 2022).

Esta abordagem inovadora permite que os agentes intercalem raciocínio simbólico com técnicas de aprendizado de máquina, oferecendo uma capacidade aprimorada para planejar e agir de acordo com as mudanças no ambiente e nas metas (YAO et al., 2022).

2.5 Frameworks

No campo da IA, frameworks especializados têm desempenhado um papel crucial na construção e implementação de sistemas baseados em linguagem natural. Dois desses frameworks notáveis são LangChain e LlamaIndex, ambos oferecendo ferramentas poderosas para lidar com tarefas complexas de processamento e geração de linguagem.

2.5.1 LangChain

LangChain é um framework de código aberto desenvolvido para facilitar a criação de aplicações baseadas em LLMs. Seu foco principal é fornecer uma infraestrutura robusta para a integração e uso de LLMs em sistemas reais. O LangChain permite que desenvolvedores criem, implementem e gerenciem fluxos de trabalho complexos envolvendo processamento de linguagem natural, desde a simples consulta a um modelo de linguagem até a construção de pipelines de processamento de dados sofisticados (LANGCHAIN, 2024).

2.5.2 LlamaIndex

O LlamaIndex é um framework especializado em estruturar e recuperar informações de grandes volumes de texto, sendo ideal para documentos extensos e dados não estruturados. Desenvolvido por pesquisadores da comunidade de IA, ele facilita a extração de informações relevantes e a organização de conteúdos complexos, permitindo a criação de índices a partir de diversas fontes de dados, como textos e artigos acadêmicos, e consultas avançadas sobre esses índices (LLAMAINDEX, 2024).

3 PROCEDIMENTOS METODOLÓGICOS

A metodologia empregada neste trabalho baseou-se em uma pesquisa e revisão bibliográfica, com o intuito de fornecer uma visão abrangente do estado atual da pesquisa e das práticas no campo da Arquitetura RAG e dos agentes. A primeira etapa consistiu na identificação e seleção de materiais relevantes, incluindo livros, artigos acadêmicos, documentações e páginas web que abordam a Arquitetura RAG e a aplicação de agentes. Foram coletados trabalhos nacionais e internacionais que tratam das teorias e práticas associadas a essa arquitetura e ao uso de agentes em diferentes contextos. Posteriormente, foram consultados livros especializados e recursos digitais que exploram em profundidade os conceitos, implementações e desafios associados à Arquitetura RAG. A análise desses materiais permitiu a construção de uma visão consolidada e atualizada sobre o tema, identificando as principais contribuições, tendências emergentes e possíveis lacunas na literatura existente. Com base nessa revisão, foi elaborado o conteúdo do artigo, visando oferecer uma contribuição significativa para o entendimento e a evolução do campo.

4 RESULTADOS E DISCUSSÃO

Considerando os temas abordados em nosso referencial teórico, fica evidente que a integração entre Inteligência Artificial, LLMs, Arquitetura RAG, ReAct e frameworks como

LangChain e LlamaIndex oferece um panorama inovador e multifacetado para o desenvolvimento de sistemas avançados de IA. Primeiramente, a utilização de LLMs como o ChatGPT e LLaMA demonstra avanços significativos na capacidade de gerar e entender texto natural. Esses modelos, têm mostrado um desempenho notável na criação de respostas contextuais e na compreensão semântica, melhorando a interação humano-computador e ampliando as aplicações práticas da IA em diversas áreas (BROWN et al., 2020).

A Arquitetura RAG oferece um método eficiente para combinar recuperação de informações com geração de texto, esse modelo promove uma resposta mais informada e precisa ao incorporar dados externos diretamente no processo de geração, melhorando a relevância das respostas geradas. Por outro lado, o ReAct e o conceito de Agentes, trazem um avanço significativo ao integrar raciocínio e ação, possibilitando a criação de agentes mais adaptativos e responsivos a contextos dinâmicos.

Os frameworks para RAG, por sua vez, demonstram-se complementares em termos de funcionalidade. Desta forma, a adoção e integração dessas ferramentas e modelos confirmam a teoria existente e ampliam as possibilidades de aplicação da IA apresentando soluções práticas para desafios complexos na área.

4.1 Análise comparativa com o Fine-Tuning

Fine-Tuning (FT) é o processo de ajustar um modelo de linguagem pré-treinado com dados específicos de um domínio para melhorar seu desempenho em tarefas específicas, refinando seus pesos internos para alinhar-se às necessidades do caso de uso (GAO et al., 2023; OVADIA, O. et al, 2024).

RAG e FT são abordagens distintas para aprimorar modelos de linguagem em tarefas baseadas em conhecimento. Enquanto o FT ajusta um modelo pré-treinado para tarefas específicas, exigindo grandes volumes de dados rotulados e alto consumo computacional, o RAG usa bases de conhecimento externas para complementar a geração de respostas, sendo mais eficiente para tarefas que necessitam de informações atualizadas. O RAG combina embeddings densos para recuperar documentos relevantes, sem alterar o modelo base, sendo ideal para cenários dinâmicos onde o retreinamento constante é inviável (GAO et al., 2023; KARPUKHIN et al., 2020; OVADIA, 2024).

Em análise quantitativa, o FT melhora o desempenho, mas não alcança os níveis de precisão do RAG (SOUDANI, H. et al, 2024). A acurácia em tarefas de perguntas e repostas, a melhor configuração foi a combinação de FT e RAG, demonstrando precisão

significativamente superior. No estudo analisado, para o modelo FlanT5-base do Google, a configuração com FT e com RAG alcançou uma precisão de 63,29%, enquanto com FT e sem RAG ficou em 9,92%, demostrando assim a relevância do RAG na geração de respostas (NGUYEN, Z. et al, 2024; SOUDANI, H. et al, 2024).

4.2 Desafios para a utilização de RAG em larga escala

Os desafios para a adoção da Arquitetura RAG em larga escala incluem a complexidade na integração e no gerenciamento de grandes volumes de dados, exigindo uma infraestrutura robusta e escalável. Além disso, problemas de latência e sobrecarga computacional podem surgir durante o processamento e a recuperação em tempo real. O alto custo relacionado ao armazenamento, processamento de dados e manutenção de índices também é um fator crítico. Esses obstáculos evidenciam a necessidade de soluções técnicas eficientes para garantir a viabilidade da Arquitetura RAG em ambientes de alta demanda, equilibrando desempenho e custo. A complexidade do raciocínio em tempo real e a integração de fontes de dados diversas são desafios que continuam a ser abordados por novas pesquisas e algoritmos, visando melhorar a eficiência dos sistemas (KARPUKHIN et al., 2020).

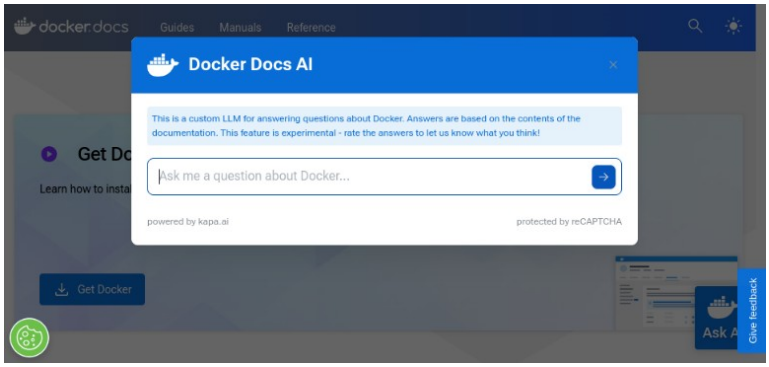
4.3 Estudo de caso da Kapa.ai e documentações técnicas

Embora o conceito de RAG esteja se consolidando como uma solução poderosa para aumentar a eficiência de sistemas de geração de texto, sua aplicação em documentações técnicas ainda é relativamente recente. Um exemplo notável dessa integração é o portfólio desenvolvido pela Kapa, empresa especializada na criação de chatbots para documentações técnicas. A Kapa implementa soluções baseadas em RAG em serviços como Docker, NextJS e CircleCI, utilizando essa arquitetura para otimizar a recuperação e apresentação de informações, facilitando o acesso dos usuários a conteúdos relevantes e precisos (KAPA, 2024).

A **Figura 2** ilustra como a Kapa implementou o projeto RAG na interface de busca da documentação do Docker, aprimorando a precisão das respostas para mais de 13 milhões de visitantes mensais (KAPA, 2024).

Podendo através da caixa de pergunta, “conversar” com a documentação fornecida.

Figura 2 – Exemplo Docker Docs



Fonte: Docker (2024)

A Kapa demonstrou uma redução significativa no volume mensal de tickets de suporte, com quedas de 20% e 28% nas ferramentas CircleCI e Mapbox, respectivamente (KAPA, 2024). Isso evidencia a eficiência da Arquitetura RAG em automatizar tarefas que tradicionalmente dependiam de equipes de suporte, otimizando processos e recursos.

4.4 Ética e Implicações sociais no uso de RAG

A integração de sistemas de inteligência artificial com considerações éticas representa um desafio crucial, questões relacionadas à privacidade, viés nos dados e impactos sociais emergem como barreiras importantes no desenvolvimento e uso responsável dessas tecnologias. As soluções baseadas em Arquitetura RAG levantam questões éticas importantes, como o uso de dados sensíveis, podendo comprometer a privacidade dos usuários caso os sistemas não sejam projetados com salvaguardas adequadas, e a redução de demanda de setores de suporte ao usuário (BOSTROM; YUDKOWSKY, 2014; ZENG, S. et al, 2024). Assim, é essencial adotar práticas rigorosas de auditoria e transparência para mitigar riscos éticos e garantir a justiça e a acessibilidade dos sistemas desenvolvidos.

5 CONSIDERAÇÕES FINAIS

Através da análise realizada, fica evidente que a Arquitetura RAG representa um avanço significativo na integração de modelos de recuperação e geração de informações. O objetivo principal deste artigo foi explorar a eficácia e as implicações da Arquitetura RAG em sistemas de inteligência artificial, com foco em suas aplicações práticas e desafios associados. Os principais resultados obtidos mostram que a combinação de técnicas de recuperação com modelos geradores proporciona melhorias notáveis na precisão e relevância das respostas, tornando os sistemas mais eficientes e adaptáveis.

No entanto, a pesquisa revela desafios importantes, como a complexidade na integração de grandes volumes de dados, custos elevados de armazenamento e processamento, e a latência que pode impactar o desempenho. Esses obstáculos destacam a necessidade de soluções técnicas avançadas para garantir a viabilidade da arquitetura em larga escala.

Para futuras pesquisas, recomenda-se uma investigação mais aprofundada sobre técnicas de otimização para melhorar a eficiência do processamento e reduzir os custos operacionais. Além disso, a exploração de métodos inovadores para integrar a Arquitetura RAG com outras abordagens emergentes, podem oferecer novos insights e possibilidades. Em conclusão, embora a Arquitetura RAG represente um avanço significativo no campo da IA, uma avaliação contínua e o desenvolvimento de soluções para suas limitações são essenciais para maximizar seu potencial e eficácia. A combinação de recuperação de informações com geração de linguagem abre novas possibilidades para a criação de sistemas mais robustos e capazes de lidar com tarefas complexas e exigentes em diversas áreas do conhecimento.

REFERÊNCIAS

BOSTROM, N.; YUDKOWSKY, E. The Ethics of Artificial Intelligence. 2014.

BROWN, T. B. et al. Language Models are Few-Shot Learners. 28 maio 2020.

FINARDI, P. et al. The Chronicles of RAG: The Retriever, the Chunk and the Generator. 15 jan. 2024.

GAO, Y. et al. Retrieval-Augmented Generation for Large Language Models: A Survey. 18 dez. 2023.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. MIT Press, 2016.

JOHNSON, J.; DOUZE, M.; JÉGOU, H. Billion-scale similarity search with GPUs. 28 fev. 2017.

KAPA. Kapa AI Documentation. Disponível em: <https://docs.kapa.ai>. Acesso em: 27 nov. 2024.

KARPUKHIN, V. et al. Dense Passage Retrieval for Open-Domain Question Answering. 10 abr. 2020.

LANGCHAIN. LangChain Documentation. Disponível em: <https://www.langchain.com/>. Acesso em: 11 ago. 2024.

LEWIS, P. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. 22 maio 2020.

LLAMAINDEX. LlamaIndex Documentation. Disponível em: <https://www.llamaindex.ai/>. Acesso em: 11 ago. 2024.

NG, A. What Artificial Intelligence Can and Can't Do Right Now. Disponível em: <https://hbr.org/2016/11/what-artificial-intelligence-can-and-cant-do-right-now>. Acesso em: 11 ago. 2024.

NGUYEN, Z. et al. Enhancing Q&A with Domain-Specific Fine-Tuning and Iterative Reasoning: A Comparative Study. 10 abr. 2024.

OVADIA, O. et al. Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLMs. 30 jan. 2024.

RUSSEL, S.; NORVIG, P. Artificial Intelligence: A Modern Approach. 2016.

SOUDANI, H. et al. Fine Tuning vs. Retrieval Augmented Generation for Less Popular Knowledge. 27 set. 2024.

TOUVRON, H. et al. LLaMA: Open and Efficient Foundation Language Models. 27 fev. 2023.

WANG, J. et al. Milvus: A Purpose-Built Vector Data Management System. Proceedings of the 2021 International Conference on Management of Data. New York, NY, USA: ACM, 9 jun. 2021.

YAO, S. et al. ReAct: Synergizing Reasoning and Acting in Language Models. 5 out. 2022.

ZENG, S. et al. The Good and The Bad: Exploring Privacy Issues in Retrieval-Augmented Generation (RAG). 23 fev. 2024.