VI SIMAC
SIMPÓSIO ACADÊMICO
DA FACULDADE ENGENHEIRO
SALVADOR ARENA

Mitigação de alucinações em modelos de linguagem de grande escala: implementação e validação de uma arquitetura baseada em *retrieval-augmented generation* (RAG)

Lucas Alves Silva, 081200031@faculdade.cefsa.edu, FESA;
Leonardo Leal Fernandes, 081200040@faculdade.cefsa.edu.br, FESA;
Cayo Vinicius Neves Magalhães, 081200050@faculdade.cefsa.edu.br, FESA;
Lucas Vieira Queiroz, 081200020@faculdade.cefsa.edu.br, FESA;
Gabriel Lara Baptista, pro15969@cefsa.edu.br, FESA.

Resumo

Os *Large Language Models* (LLMs) são modelos probabilísticos de inteligência artificial generativa de texto que, devido a problemas de alucinações, podem gerar informações incorretas ou irrelevantes. Este estudo investiga a aplicação da técnica de *Retrieval-Augmented Generation* (RAG), que combina a geração de texto com a recuperação de informações contextuais de um banco de dados vetorizado, permitindo que as respostas sejam complementadas com dados mais precisos e atualizados. A metodologia incluiu o desenvolvimento de APIs para a extração e processamento de textos e sua validação foi realizada através de testes funcionais, de desempenhos e qualitativos, utilizando a ferramenta Postman. Os resultados preliminares indicam a eficiência da aplicação em reduzir alucinações devido a limitações de modelos treinados com dados desatualizados, além de uma alta escalabilidade e flexibilidade, mesmo sob condições de alta demanda. Esses resultados sugerem que o RAG é uma abordagem promissora para aprimorar a qualidade das respostas em LLMs, especialmente em aplicações onde a precisão e a atualização das informações são cruciais.

Palavras-chave: *Retrieval-Augmented Generation*, Mitigação de Alucinações, *Large Language Models*, Escalabilidade, Inteligência Artificial Generativa.

Introdução

Os modelos de linguagem de larga escala, conhecidos como *Large Language Models* (LLMs) permitem que sistemas de inteligência artificial produzam respostas complexas e contextualmente ricas. No entanto, a natureza probabilística dessas gerações traz um desafio: as chamadas "alucinações", onde o modelo gera informações que, embora coerentes, são incorretas ou irrelevantes (Bender *et al.*, 2021). Esse problema é particularmente preocupante em aplicações onde a precisão das informações é crítica, como na medicina, direito e outras áreas de alto impacto.

Para enfrentar esse desafio, este estudo investiga a aplicação de *Retrieval-Augmented Generation* (RAG) como uma abordagem eficaz para mitigar alucinações em LLMs. O RAG é uma técnica que combina a geração de texto com a recuperação de informações contextuais armazenadas em bancos de dados vetorizados, permitindo que o modelo complemente sua geração com dados precisos e atualizados, melhorando a confiabilidade das respostas (Monigatti, 2023). Essa integração busca minimizar as limitações inerentes às LLMs, ao fornecer um contexto mais sólido e específico, resultando em uma redução significativa na ocorrência de alucinações.

Este trabalho utilizou uma abordagem exploratória para identificar as ferramentas e metodologias mais adequadas para implementar essa arquitetura. A pesquisa incluiu o desenvolvimento de APIs

específicas para a extração e processamento de textos de diferentes formatos, como PDF, TXT e DOCX, segmentando-os em blocos (*chunks*) de tamanho adequado e com sobreposição (*overlap*) para garantir a continuidade do contexto (Cândido; Júnior, 2022). Além disso, foi utilizado o Redis como sistema de indexação, permitindo buscas rápidas e eficientes que enriquecem o processo de geração de texto pela IAG com dados contextuais relevantes em tempo real (Wahlin *et al.*, 2024).

Com a implementação do RAG, este estudo contribui para o desenvolvimento de sistemas de inteligência artificial mais precisos e confiáveis, especialmente em contextos em que a exatidão da informação é essencial.

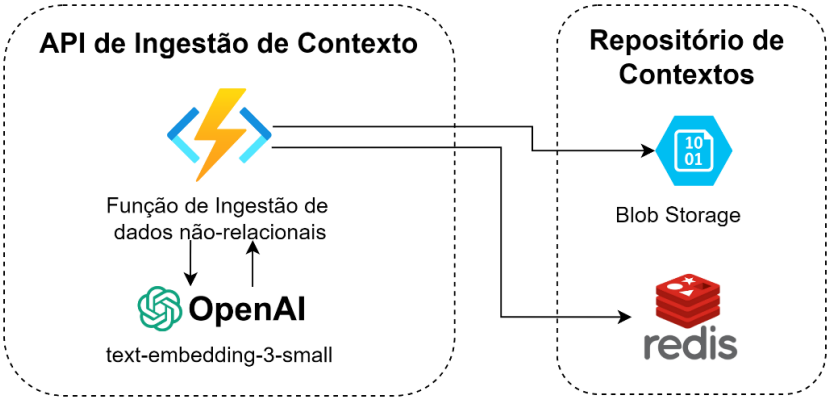
Metodologia

Este trabalho seguiu uma abordagem exploratória para identificar as ferramentas mais adequadas para a implementação de uma arquitetura baseada em *Retrieval-Augmented Generation* (RAG), com o objetivo de mitigar as alucinações em *Large Language Models* (LLMs). Segundo Cândido e Júnior (2022), a escolha criteriosa de tecnologias é fundamental para garantir a eficácia e a eficiência das soluções propostas, especialmente em cenários de alta demanda e complexidade.

A primeira etapa da metodologia envolveu o desenvolvimento de uma API em Function Servless para extração e processamento de textos provenientes de documentos nos formatos PDF, TXT e DOCX. A API segmenta o texto em blocos, ou *chunks*, com tamanho de x caracteres e sobreposição (*overlap*) de y caracteres. A escolha de se utilizar o *overlap* tem o intuito garantir a continuidade do contexto entre os blocos de texto, uma vez que a sobreposição permite que partes relevantes da informação sejam incluídas no bloco subsequente, evitando a perda de contexto importante (Cândido; Júnior, 2022).

Cada *chunk* passa por um processo de vetorização através do modelo “text-embedding-3-small” da Open AI, gerando um *embedding* (vetores que representam o bloco de texto). Em seguida, cada *chunk* e seu *embedding* respectivo será armazenado no Redis, criando um índice (*index*), e o arquivo recebido será armazenado no Repositório de armazenamento (Blob Storage), ferramenta do ambiente de computação em nuvem Azure. A figura 1 demonstra a arquitetura por trás do funcionamento.

Figura 1 – Arquitetura da API de Ingestão de Contexto.



Fonte: Autoria própria (2024)

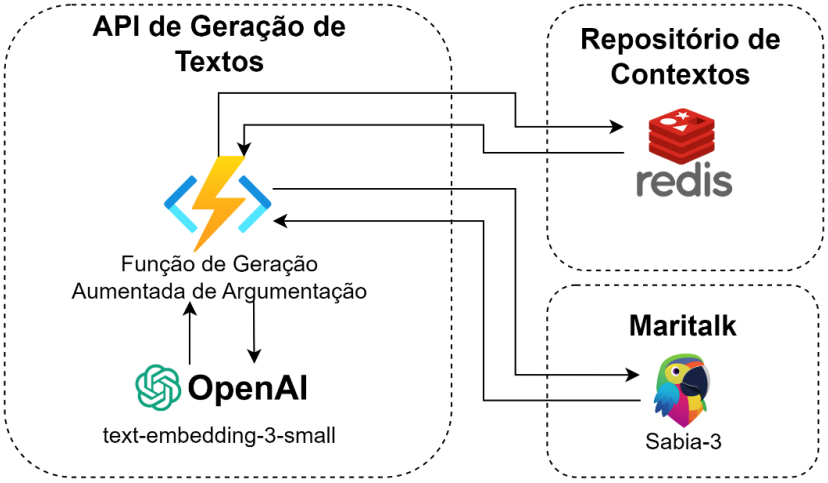
O Redis é uma solução de armazenamento em memória de alto desempenho, utilizado para gerenciar os índices dos dados vetorizados armazenados (Redis, 2024). Esses índices permitem que o sistema RAG

realize buscas rápidas e eficientes, garantindo que a IAG recupere dados contextuais relevantes em tempo real com alta disponibilidade (Wahlin *et al*., 2024).

A segunda API desenvolvida em Function Servless se concentra na geração de texto, aplicando a metodologia RAG. Esse processo integra a query do usuário com o contexto recuperado e uma regra específica que orienta o LLM na criação da resposta final. A pesquisa por similaridade é realizada com a pergunta do usuário vetorizada pelo modelo "text-embedding-3-small" e utilizando os índices armazenados no Redis, onde o sistema identifica os vetores dos blocos de texto mais semelhantes à query do usuário. Em seguida, a API seleciona os cinco resultados mais relevantes e os utiliza como contexto para a LLM (modelo Sabia-3 da Maritalk) gerar uma resposta informada e precisa (Monigatti, 2023; Rangasamy, 2024).

Essa abordagem, como ilustrado na Figura 2, garante que a resposta seja coerente e fundamentada em informações contextuais específicas, minimizando significativamente as chances de alucinações. Conforme Monigatti (2023), a combinação de recuperação de informação com geração de texto é uma técnica poderosa para aumentar a precisão e a relevância das respostas em LLMs.

Figura 2 – Arquitetura da API de Geração de Texto



Fonte: Autoria própria (2024)

A validação da arquitetura desenvolvida está conduzida em três tipos principais de testes:

- Testes Funcionais: Para avaliar o funcionamento básico do sistema e garantir que todas as funcionalidades operem conforme o esperado, serão realizados testes funcionais utilizando a ferramenta Postman. Essa ferramenta permitirá a execução de testes repetidos, como descrito na seção de resultados preliminares, assegurando a consistência e a robustez do sistema sob diferentes condições operacionais. Conforme destacado por Gordon (2022), os testes funcionais são essenciais para garantir que o sistema atenda aos requisitos especificados e funcione corretamente em diversos cenários de uso. O Postman, devido à sua capacidade de automatização e validação de requisições API, desempenha um papel crucial nesse processo.
- Testes de Desempenho: A mensuração do desempenho do sistema sob diferentes cargas de trabalho será realizada por meio de testes de desempenho, também utilizando o Postman. Esses testes permitirão simular cenários variados de carga, avaliando a resposta do sistema frente a variações no número de usuários e na quantidade de requisições. Testes de carga, como os discutidos

anteriormente, fornecerão métricas essenciais, tais como tempo de resposta, taxa de transferência (*throughput*) e estabilidade sob condições de carga. Wahlin e Steen (2024) destacam a importância dos testes de desempenho para garantir a escalabilidade e a capacidade de resposta de sistemas baseados em LLMs. A aplicação do Postman nesses testes oferece uma análise detalhada do comportamento do sistema em condições realistas, fornecendo subsídios valiosos para futuras otimizações.

- Testes Qualitativos: Além dos testes técnicos, será conduzida uma análise qualitativa comparativa entre as respostas geradas com e sem a utilização do RAG. Essa análise incluirá a avaliação da precisão, relevância e clareza das respostas, proporcionando uma visão abrangente sobre a eficácia do sistema. Adicionalmente, será coletado feedback de usuários finais para avaliar a satisfação geral com o sistema, oferecendo insights importantes sobre a experiência do usuário e identificando áreas para melhorias contínuas. O estudo de Bender *et al*. (2021) ressalta a relevância da qualidade das respostas em sistemas que utilizam LLMs, especialmente em contextos em que a precisão das informações é de suma importância.

Essas etapas de validação são cruciais para assegurar que a solução proposta não só atenda aos requisitos técnicos, mas também ofereça uma experiência de usuário satisfatória e informativa. As próximas fases do estudo incluirão a aplicação desses testes em ambientes controlados e a análise dos resultados para orientar ajustes e otimizações futuras.

Resultados preliminares

Os resultados preliminares deste estudo indicam que a aplicação da técnica *Retrieval-Augmented Generation* (RAG) em *Large Language Models* (LLMs) continua a mostrar-se eficaz na mitigação de alucinações, um desafio significativo nesses modelos. A implementação do RAG, que combina a geração de texto com a recuperação de informações contextuais armazenadas em um banco de dados vetorizado, resultou em respostas mais precisas, coerentes e baseadas em dados atualizados, superando limitações impostas pelo treinamento dos modelos, como o Sabiá-3, cujos dados se estendem até 2023.

O teste de performance realizado com um perfil de carga crescente e decrescente demonstrou que o sistema é capaz de manter sua eficiência mesmo sob alta demanda. O teste foi conduzido simulando uma carga inicial de 8 usuários durante 5 minutos, que aumentou gradualmente para 40 usuários ao longo de 10 minutos. Essa carga foi mantida por 30 minutos, seguida por uma redução gradual de 40 para 8 usuários nos 10 minutos subsequentes, finalizando com 8 usuários por mais 5 minutos. Os resultados obtidos estão representados na Tabela 3.

Tabela 3 – Resultados do teste de performance realizado via Postman da API de Geração de Textos

Métrica	**Resultado**
Total de Requisições	4.364 unidades
Throughput	1,21 requisições por segundo
Tempo médio de resposta	39,261 ms
Percentil 95 de tempo de resposta	52.952 ms
Percentil 90 de tempo de resposta	54.065 ms
Erro	0,00%

Fonte: Autoria própria (2024)

O Redis desempenhou um papel crucial, permitindo buscas rápidas em grandes volumes de dados vetorizados. O tempo médio de resposta foi de 39 s, com a maioria das requisições (95%) sendo processadas em menos de 54 s. Esses tempos de resposta são essenciais para garantir que as interações dos usuários sejam rápidas e informativas, mesmo sob alta carga.

A arquitetura demonstrou ser altamente flexível e escalável, sendo capaz de sustentar um aumento significativo na carga de usuários sem comprometer o desempenho. A carga máxima de 40 usuários foi mantida por 30 minutos, e o sistema respondeu consistentemente com tempos de resposta que variaram entre 14 s e 55 s, confirmando a robustez da solução.

Uma vantagem significativa do RAG é a capacidade de permitir que a LLM trabalhe com dados atualizados, superando a limitação do Sabiá-3, cujo treinamento abrange dados até 2023. Ao integrar dados contextuais mais recentes, recuperados dinamicamente, o sistema melhora a relevância e a precisão das respostas, adaptando-se melhor às necessidades dos usuários em tempo real.

Referências

CÂNDIDO, Ana Clara; ARAÚJO JÚNIOR, Rogério Henrique de. Potencialidades do desenvolvimento de cloud computing no âmbito da gestão da informação. Perspectivas em Ciência da Informação, 2022. Disponível em: https://www.scielo.br/j/pci/a/rXjTqsQByRGZp6NQxSr8Wyw/?lang=pt. Acesso em: 5 mai. 2024.

BENDER, Emily M.; GEBRU, Timnit; McMILLAN-MAJOR, Angelina; SHMITCHELL, Shmargaret. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021. Disponível em: https://dl.acm.org/doi/10.1145/3442188.3445922. Acesso em: 17 mar. 2024.

MONIGATTI, Leonie. Retrieval-Augmented Generation (RAG): From Theory to LangChain Implementation. Towards Data Science, 2023. Disponível em: https://towardsdatascience.com/retrieval-augmented-generation-rag-from-theory-to-langchain-implementation-4e9bd5f6a4f2. Acesso em: 17 mar. 2024.

RANGASAMY, Venkat. Mastering AI in Business: Building an Enterprise-Ready AI Platform with RAG and CRAG. Cubed Blog, 2024. Disponível em: https://blog.cubed.run/mastering-ai-in-business-building-an-enterprise-ready-ai-platform-with-rag-and-crag-b38baac8ad8b. Acesso em: 5 mar. 2024.

REDIS. Redis Documentation. 2024. Disponível em: https://redis.io/docs/. Acesso em: 20 ago. 2024.

WAHLIN, Dan; STEEN, Heidi. RAG (Geração Aumentada de Recuperação) na IA do Azure Search. Microsoft Azure, 2024. Disponível em: https://learn.microsoft.com/pt-br/azure/search/retrieval-augmented-generation-overview. Acesso em: 17 mar. 2024.