Comparação entre modelos de LLM para recuperação de dados proveniente de páginas estáticas com arquitetura RAG

Comparison of LLM Models for Data Retrieval from Static Pages Using RAG Architecture

Filipe Vasconcelos Moreno[1]

Orientador: Rodrigo F. A. Paiva de Oliveira

Resumo

Este estudo apresenta uma análise comparativa entre quatro modelos de linguagem de grande porte (LLMs) Llama 3.1, Dolphin 3, Mistral e Zephyr, aplicados à recuperação de dados de páginas estáticas com uso da arquitetura RAG (Retrieval-Augmented Generation). Utilizando cinco páginas da Wikipédia em diferentes idiomas e temáticas, o experimento avaliou o desempenho dos modelos com base em métricas como robustez a ruído, recusa negativa, integração de informações, resistência a contrafactuais, fidedignidade e relevância da resposta. Os testes foram conduzidos em ambiente local com hardware controlado, utilizando ferramentas como LangChain, ChromaDB e RAGAS. Os resultados indicam que o Llama 3.1 foi o mais robusto contra ruído e ausência de contexto, enquanto o Dolphin 3 demonstrou melhor relevância e menor tempo de resposta. O Mistral apresentou alta fidedignidade, porém com respostas menos pertinentes, e o Zephyr obteve desempenho inferior em quase todos os critérios. As conclusões reforçam que não há um modelo universalmente superior, sendo essencial alinhar a escolha do LLM às necessidades específicas da aplicação. A arquitetura RAG provou-se eficaz na mitigação de alucinações e na ampliação do contexto, embora ainda dependa fortemente da qualidade dos documentos recuperados. Este trabalho contribui para decisões mais fundamentadas na escolha de LLMs para aplicações em ambientes com dados estáticos e restrições de infraestrutura.

Palavras-chave: LLM; recuperação de dados; RAG; páginas estáticas; avaliação de modelos.

Abstract

This study presents a comparative analysis of four large language models (LLMs), Llama 3.1, Dolphin 3, Mistral, and Zephyr, applied to data retrieval from static web pages using the Retrieval-Augmented Generation (RAG) architecture. The experiment involved five Wikipedia pages in different languages and topics, evaluating model performance through metrics such as noise robustness, negative rejection, information integration, counterfactual robustness, faithfulness, and answer relevancy. All tests were conducted in a controlled local environment using tools like LangChain, ChromaDB, and RAGAS. Results show that Llama 3.1 was the most resilient to irrelevant or missing context, while Dolphin 3 stood out for its high answer relevancy and faster response time. Mistral achieved the highest faithfulness scores but struggled with directness and completeness of responses, whereas Zephyr performed the weakest across most dimensions. Findings suggest there is no universally superior model, and selecting the right LLM depends on the specific needs of each application. The RAG architecture proved effective in enhancing factual grounding and mitigating hallucinations, though it remains sensitive to the quality of retrieved content. This research supports more informed decisions in LLM selection for applications involving static data and limited computational infrastructure.

Keywords: LLM; data retrieval; RAG; static pages; model evaluation.

1 Introdução

Este trabalho tem como objetivo apresentar uma análise comparativa sobre a recuperação de dados de páginas estáticas utilizando modelos de linguagem de grande porte, Large Language Models (LLMs) e uma arquitetura de Retrieving Augmented Generation (RAG). O estudo busca compreender as nuances e desafios envolvidos na escolha de um modelo mais eficaz para essa tarefa, destacando a importância de métodos robustos e padronizados de avaliação.

1.1 Motivação e Contexto

A recuperação de dados de páginas estáticas, frequentemente referida como "Web-Retrieving" ou "Web-Scraping" em um contexto mais amplo, é um desafio contínuo e complexo no campo da ciência da computação. Esse processo envolve a extração de informações específicas de páginas web, onde a precisão, a exatidão e a relevância dos dados recuperados são fundamentais para que esses dados sejam úteis para os objetivos específicos da análise ou da aplicação (Dogucu; Çetinkaya-Rundel, 2020; Khder, 2021).

A demanda por métodos eficazes de extração de dados cresceu drasticamente, impulsionada pela necessidade de informações confiáveis para tomada de decisão em áreas como pesquisa acadêmica, inteligência de mercado e desenvolvimento tecnológico.

A complexidade desse desafio aumenta quando se busca não apenas a coleta de dados, mas também a garantia de sua qualidade e relevância. Informações imprecisas ou incorretas podem levar a decisões baseadas em dados falhos, resultando em despesas adicionais, riscos operacionais e, potencialmente, danos à reputação das organizações.

Além disso, informações que não sejam factuais podem contribuir para a disseminação de dados incorretos ou enganosos, especialmente em contextos críticos, como o desenvolvimento de políticas públicas ou estratégias empresariais (Vijayaragavan Pichiyan et al., 2023).

Com o avanço dos LLMs, surgiu uma oportunidade de explorar métodos mais sofisticados e adaptáveis, capazes de interpretar o conteúdo de páginas web com maior sensibilidade ao contexto, oferecendo assim respostas mais ajustadas às necessidades de precisão e contextualização dos dados (CHEN, J. et al., 2024).

Essa nova abordagem representa um avanço importante em relação às técnicas de scraping convencionais, pois permite que modelos entendam nuances de linguagem e contexto que antes eram difíceis de capturar, devido a necessidade de alocação manual de variáveis e ajustes a cada atualização da página ao qual modifique posicionamento ou conteúdo.

A motivação científica por trás deste estudo está na necessidade de entender o desempenho de LLMs na tarefa de recuperar informações de páginas estáticas de maneira consistente, robusta e factível de modo a oferecer maior confiança e utilidade na utilização de tais modelos para busca de informações confiáveis.

Além das motivações científicas, há um interesse crescente no mercado por soluções que otimizem a recuperação de informações, especialmente em áreas onde o volume e a complexidade dos dados aumentam continuamente, como pesquisa acadêmica, negócios e gestão de conhecimento. A capacidade de acessar dados relevantes e confiáveis de maneira rápida e eficiente é um diferencial competitivo, permitindo que organizações e pesquisadores tomem decisões informadas e adaptem-se rapidamente às mudanças BRAND, J.; ISRAELI, A.; NGWE, D., 2023).

Inovações recentes, como a Geração Aumentada por Recuperação (Retrieve Augmented Generation, RAG), têm demonstrado grande potencial na recuperação de informações, permitindo a

combinação de dados gerados por LLMs com informações recuperadas de fontes confiáveis através de uma busca vetorial.

No entanto, esses métodos também enfrentam desafios significativos, como a complexidade de integração de sistemas e a dependência da qualidade dos dados utilizados para alimentar esses modelos, o que pode impactar a precisão das respostas geradas (ZHAO, P. et al, 2024).

1.2 Justificativa

A relevância deste estudo reside na crescente dependência de sistemas automatizados para a recuperação de informações precisas e relevantes, uma demanda que é amplamente evidenciada pela literatura e pelo mercado atual.

Empresas e instituições de pesquisa têm explorado o potencial da automação para processos de indexação e recuperação de dados provenientes de múltiplas fontes, incluindo páginas web, documentos em PDF e outros repositórios de informações. Esse movimento busca otimizar a eficiência e a precisão na obtenção de dados, permitindo decisões mais rápidas e informadas em setores como tecnologia, negócios e pesquisa acadêmica.

Contudo, as LLMs apresentam limitações importantes que, se não forem superadas, podem comprometer a confiabilidade e a eficácia dessas soluções em um ambiente onde a precisão e a contextualização são essenciais. Entre os desafios mais críticos está a capacidade limitada dos LLMs em lidar com contextos longos, principalmente refletida em LLMs com quantidade de parâmetros reduzida, o que afeta diretamente a qualidade das informações recuperadas e pode levar a respostas fragmentadas ou imprecisas.

A evolução contínua dos LLMs para incorporar capacidades de contexto estendido (Li, Z. et al., 2024), demonstrando assim, um passo promissor, pois possibilita que esses modelos atuem de forma mais autônoma e assertiva, potencialmente impulsionados com arquiteturas de aumentação como o RAG, resultando em uma maior precisão e menor custo computacional.

Diante desse cenário, a escolha inadequada de um modelo de linguagem pode resultar não apenas em dados imprecisos ou irrelevantes, mas também em decisões equivocadas, cujas consequências incluem aumento de custos, riscos operacionais e até danos à reputação da organização.

O uso de modelos com limitações contextuais eleva a probabilidade de má interpretação dos dados, levando a ações baseadas em informações incorretas e, por extensão, comprometendo a segurança e a eficácia de aplicações críticas (Yang, Y. et al., 2024). Assim, este estudo, ao comparar e avaliar o resultado de diversos modelos open-source, permitirá uma escolha mais consciente e fundamentada de LLM, promovendo uma utilização otimizada que reduz falhas e aumenta a confiabilidade dos sistemas com arquitetura RAG ou suas variações.

1.3 Problema

A recuperação de dados provenientes de páginas web não é um desafio recente, no entanto, muito complexo e delicado, este desafio apresenta cenários significativamente complexos, assim tornando a coleta de dados não apenas uma uma etapa fundamental para o fornecimento de dados, mas sensível. É crucial para diversas aplicações, desde a pesquisa acadêmica até a análise de mercado, onde a precisão e a relevância das informações recuperadas são de extrema importância. No entanto, a eficácia dos LLMs nessa tarefa ainda é limitada por vários fatores, incluindo a capacidade de contextualização, a precisão dos dados recuperados e a eficiência computacional.

A recuperação de dados a partir de páginas estáticas da web com o uso de LLMs envolve várias camadas conceituais e técnicas. Primeiramente, entende-se por recuperação de dados o processo de extrair informações relevantes de fontes digitais, estruturadas ou não estruturadas, de maneira que os dados sejam acessíveis e utilizáveis para análises específicas. No caso das páginas estáticas, estas se caracterizam por conteúdo fixo, que não muda com a interação do usuário e, portanto, não são atualizadas dinamicamente como ocorre em plataformas interativas.

Tais páginas são comuns em ambientes acadêmicos e empresariais, onde informações precisam ser consultadas sem que o conteúdo dependa de scripts ou da interação com sistemas de banco de dados.

Os LLMs, como GPT-4 e derivantes da arquitetura Transformers (VASWANI et al., 2017), utilizam processamento de linguagem natural para compreender e gerar textos. Contudo, quando aplicados à recuperação de dados, esses modelos enfrentam desafios específicos, como a necessidade de precisão contextual e eficiência computacional para lidar com grandes volumes de dados de forma confiável. Além disso, surge a necessidade de métricas robustas para avaliar a eficácia desses modelos, ou seja, sua capacidade de recuperar dados de forma precisa, relevante e com um consumo computacional factível para a demanda esperada, principalmente, quando é considerado o escopo de modelos open-source e que possuem uma quantidade de parâmetros consideravelmente inferior aos quais são providos por empresas como a OpenAI e Anthropic.

Sem o entendimento de métricas condizentes e padronizadas, é difícil determinar qual modelo é mais adequado para uma aplicação específica, o que pode resultar em escolhas subótimas e, consequentemente, em dados de baixa qualidade (Chang et al., 2023).

Em resumo, os principais conceitos abordam: (i) a natureza da recuperação de dados; (ii) a especificidade das páginas estáticas; (iii) a complexidade dos LLMs; e (iv) a avaliação do desempenho desses modelos.

O problema de recuperar dados de páginas estáticas usando LLMs é particularmente relevante em ambientes de pesquisa, empresas de tecnologia e setores de inteligência de mercado.

Em empresas voltadas à análise de dados, como consultorias de mercado ou startups de análise preditiva, a extração de informações de páginas estáticas se faz essencial para manter a competitividade e a inovação.

No setor acadêmico, a recuperação eficiente de dados pode facilitar a revisão de literatura e a coleta de informações em grande escala, o que contribui para avanços em diversas áreas do conhecimento.

Além disso, sistemas como o RAG uma técnica que combina modelos de linguagem com módulos de recuperação de informações enfrentam esse problema em contextos como plataformas de atendimento ao cliente e assistentes virtuais.

Nesses casos, a habilidade dos LLMs em buscar dados com precisão, mantendo a relevância das respostas, é vital para a experiência do usuário e para a confiabilidade da tecnologia.

Nesse sentido, estudos recentes destacam que métricas de recuperação e resposta são essenciais para solucionar falhas em LLMs e em sistemas RAG (YU, H. et al., 2024), permitindo a identificação e correção de problemas de precisão e relevância nos dados recuperados. Instituições de padronização e pesquisa, como o National Institute of Standards and Technology (NIST), também desempenham um papel importante, pois promovem benchmarks que orientam o desenvolvimento e avaliação desses modelos em ambientes onde a qualidade da recuperação de dados é crucial (NIST, 2024).

Sem critérios de avaliação claros, torna-se difícil determinar qual modelo é o mais adequado para um uso específico, o que leva à utilização de ferramentas que, muitas vezes, não conseguem atender à demanda de precisão e relevância exigida. Esse problema resulta em escolhas subótimas, que comprometem a qualidade dos dados recuperados e, por consequência, a confiabilidade das decisões baseadas nesses dados (Chang et al., 2023).

Em contextos empresariais, a falha na recuperação precisa de dados pode resultar em estratégias baseadas em informações incompletas ou incorretas, levando a prejuízos financeiros e perda de competitividade.

No ambiente acadêmico, a falta de precisão e eficiência na recuperação pode atrasar pesquisas, além de dificultar a realização de revisões literárias abrangentes e confiáveis.

Na área de atendimento ao cliente, respostas imprecisas ou irrelevantes geradas por LLMs prejudicam a experiência do usuário e afetam a reputação da empresa.

Por esses motivos, existe uma necessidade latente no entendimento e na comparação das LLMs, utilizando métricas claras, para entender e medir a eficácia dos modelos em termos de precisão das respostas, relevância e coerência dos dados recuperados, além de considerar as limitações devidas ao tamanho dos respectivos modelos em cenários open-source e on-premise ao qual não obrigatoriamente desejam expor seus dados tal como ter hardware especializado ou potente suficiente para executar modelos mais robustos.

Portanto, a questão de pesquisa que orienta este trabalho é: “Qual modelo de linguagem de grande porte open-source, com arquitetura RAG, apresenta o melhor desempenho na recuperação de dados de páginas estáticas, considerando diferentes páginas e considerando métricas de precisão, relevância, robustez ao ruído e assertividade?”

1.4 Objetivos

O presente estudo tem como objetivo principal avaliar a qualidade das respostas geradas por LLMs utilizando a arquitetura RAG, e através de avaliação humana e computacional gerar insights de vantagens, desvantagens e boas práticas da utilização de grandes modelos de linguagem para o retrieving de páginas estáticas.

1.4.1 Objetivo geral

Realizar a avaliação dos LLMs de maneira padronizada para mensurar e comparar o desempenho em diferentes cenários, levando em consideração, idioma, informação contida e métricas tais "Noise Robustness”, “Negative Rejection”, “Information Integration” e “Counterfactual Robustness”.

1.4.2 Objetivos específicos

- OE1: Identificar e analisar técnicas existentes para avaliação da qualidade das respostas geradas por LLMs, especificamente no contexto da arquitetura RAG e retrieving de páginas estáticas.
- OE2: Comparar os resultados dos modelos em cada métrica identificada, observando diferenças de desempenho e apontando quais modelos se mostram mais robustos em cada critério (CHEN et al. 2024).
- OE3: Analisar o impacto do idioma das páginas (inglês vs. português) na precisão e relevância das respostas produzidas pelos modelos.

1.5 Estrutura do Artigo

Este trabalho está estruturado em seis capítulos. O Capítulo 1 apresenta a introdução, incluindo a motivação, justificativa, definição do problema, objetivos da pesquisa e a organização do artigo. O Capítulo 2 traz a fundamentação teórica, abordando conceitos essenciais como técnicas tradicionais de extração de dados, fundamentos dos LLMs, arquitetura RAG, embeddings, métricas de avaliação e o impacto da automação em diversos setores. O Capítulo 3 descreve a metodologia utilizada, caracterizando a natureza da pesquisa, os modelos avaliados, os critérios de escolha e a infraestrutura computacional, além de detalhar as métricas adotadas para mensuração da eficácia dos modelos. No Capítulo 4, são apresentados o desenvolvimento e a execução prática do experimento, contemplando a coleta, fragmentação e indexação dos dados, a configuração dos modelos, a geração de embeddings e a aplicação das métricas Faithfulness e Answer Relevancy por meio do framework RAGAS. O Capítulo 5 reúne os resultados obtidos, oferecendo uma análise comparativa entre os modelos Llama 3.1, Mistral, Dolphin 3 e Zephyr, além de discutir limitações, desempenho computacional e robustez das respostas, incluindo um teste ilustrativo com perguntas reais. Por fim, o Capítulo 6 apresenta as conclusões, destacando os principais achados da pesquisa, a relação com os objetivos propostos, as limitações do estudo e sugestões para trabalhos futuros.

2 Fundamentação Teórica

2.1 Técnicas Tradicionais de Recuperação de Dados

2.1.1 Web Scraping

Web scraping é uma técnica tradicional e amplamente usada para a extração de dados de páginas estáticas na web. Essa técnica consiste no acesso direto ao código HTML de sites, extraindo dados específicos, como textos, tabelas e imagens, de forma programática (Dogucu & Çetinkaya-Rundel, 2020). Ferramentas como BeautifulSoup e Selenium facilitam o processo, permitindo organizar e utilizar os dados para análises variadas. No entanto, essa abordagem enfrenta limitações significativas, como a dependência da estrutura da página, que, ao sofrer alterações, pode invalidar o script de extração. Além disso, surgem questões éticas e legais em relação à coleta de dados, pois muitos sites proíbem a extração automatizada de conteúdo (Khder, 2021).

2.1.2 Métodos de Extração de Dados Estruturados e Não Estruturados

A extração de dados varia conforme a estrutura dos dados presentes nas páginas web. Dados estruturados, como tabelas e listas, são mais fáceis de extrair devido ao formato organizado, facilitando a identificação de padrões (Dogucu & Çetinkaya-Rundel, 2020). Por outro lado, dados não estruturados, como textos de artigos, representam desafios maiores e demandam o uso de processamento de linguagem natural (PLN) para interpretação. A extração de dados não estruturados geralmente requer técnicas adicionais para transformar dados textuais em informações compreensíveis para máquinas (Khder, 2021).

2.2 Modelos de Linguagem de Grande Porte (LLMs)

2.2.1 Desenvolvimento e Estrutura dos LLMs

Os LLMs são sistemas de rede neural treinados para processar e gerar texto em linguagem natural. Avanços no processamento de linguagem natural com redes neurais profundas resultaram em modelos como GPT-3 e GPT-4, que possuem bilhões de parâmetros e são capazes de interpretar contextos complexos (ZHAO et al., 2023). Esses modelos são treinados em grandes volumes de dados textuais e ajustados para responder a uma ampla gama de perguntas e tarefas de geração de texto, sendo aplicáveis em áreas como atendimento ao cliente, criação de conteúdo, medicina (ALBERTS et al., 2023), indústria (AGGARWAL et al., 2023) e educação (KASNECI et al., 2023).

2.2.2 Aplicações dos LLMs em Recuperação de Dados

Os LLMs têm sido amplamente utilizados em tarefas relacionadas à recuperação de dados devido à sua extraordinária capacidade de interpretar e compreender linguagem natural. Essa habilidade permite que esses modelos executem uma extração de informações com maior precisão, mesmo em cenários marcados por alta complexidade contextual (XU et al., 2023).

Modelos como GPT-4 e sub variações provindas da arquitetura Transformers destacam-se por oferecer soluções inovadoras quando comparados aos métodos tradicionais de recuperação de dados, uma vez que são capazes de gerar respostas que não apenas atendem às consultas de forma mais relevante, mas também apresentam um alto nível de contextualização. Isso faz com que sejam ferramentas poderosas para lidar com informações em diversos domínios e aplicações.

Apesar de suas vantagens, esses modelos enfrentam desafios significativos relacionados à precisão das informações recuperadas. Tais dificuldades se tornam mais evidentes quando o contexto envolve dados complexos ou oriundos de múltiplas fontes com diferentes níveis de confiabilidade (ZHAO et al., 2023).

2.2.3 Desafios dos LLMs em Contextos Longos e Precisão Factual

Dois dos principais desafios dos LLMs é sua limitação em lidar com contextos longos e a necessidade de consulta em múltiplos arquivos (XU et al., 2023), o que pode resultar em perda de precisão e factualidade, levando ao fenômeno de “alucinações”, onde o modelo gera respostas que parecem factuais, mas são incorretas (HUANG et al., 2024). Embora melhorias contínuas visem mitigar esse problema, o uso em contextos extensos ainda requer técnicas complementares para garantir a precisão factual (XU et al., 2023).

2.3 Modelos de Embeddings

2.3.1 Fundamentos da estrutura de embeddings

Embeddings são representações vetoriais densas que mapeiam palavras, sentenças ou documentos para um espaço contínuo de alta dimensão, de modo que itens semanticamente semelhantes fiquem próximos uns dos outros nesse espaço (ALMEIDA, F.; XEXÉO, G., 2024).

2.3.2 Estrutura de embeddings na arquitetura RAG

Na arquitetura RAG, os embeddings são gerados por um encoder dual que projeta consultas e documentos em um mesmo espaço vetorial denso, permitindo a recuperação eficiente por similaridade. Primeiro, o document encoder, normalmente um modelo pré-treinado como BERT ou um transformer similar, segmenta cada fonte em "chunks" de tamanho fixo (por exemplo, 100 palavras) e computa um embedding para cada fragmento, armazenando-os em um índice de vetores acelerado por FAISS com HNSW para buscas aproximadas de vizinhança (LEWIS et al., 2021).

Em paralelo, o question encoder codifica a pergunta do usuário no mesmo espaço vetorial, de modo que a similaridade cosseno entre embeddings de consulta e de documentos reflita a relevância semântica (LEWIS et al., 2021)

Durante a inferência, o sistema recupera os k embeddings de documentos mais próximos da consulta e concatena ou funde esses fragmentos como contexto adicional para o modelo gerador, que condiciona sua geração de texto nesses contextos não-paramétricos, frequentemente, LLMs.

2.4 Geração Aumentada por Recuperação (RAG)

2.4.1 Fundamentos da Geração Aumentada por Recuperação

A arquitetura RAG combina a capacidade dos LLMs de gerar linguagem natural com a recuperação de dados de fontes confiáveis. Essa abordagem visa mitigar problemas de factualidade e aumentar a relevância das respostas, integrando informações verificadas diretamente nas respostas geradas (ZHANG, P. et al., 2021).

O funcionamento básico do RAG se apoia em embeddings vetores densos que capturam o sentido semântico dos documentos e das consultas dos usuários. Ao receber uma pergunta (query), o sistema converte tanto o texto da consulta quanto o corpus de documentos em embeddings, em seguida compara esses vetores para identificar os trechos mais relevantes. Esses trechos são posteriormente fornecidos ao LLM como contexto adicional, garantindo que a resposta resultante reflita informações precisas e atualizadas.

A arquitetura tem se mostrado especialmente eficaz em áreas onde a precisão é fundamental, tal qual a escolha específica de informações, que podem ser passadas sem a necessidade de retreinamento do modelo de linguagem. Áreas como medicina e pesquisa acadêmica, onde a confiabilidade dos dados é essencial, têm sido zonas promissoras (ALBERTS et al., 2023).

2.4.2 Benefícios e Limitações da RAG

Os principais benefícios da RAG incluem ganhos expressivos em precisão e contextualização das respostas, uma vez que ela incorpora informações recuperadas de fontes confiáveis muitas vezes não disponíveis no conjunto de treinamento dos LLMs. Isso permite acessar dados proprietários de empresas, detalhes específicos de conversas, textos e artigos recentes ainda não catalogados, entre outros insumos especializados. Ao enriquecer o processo

de geração com esse contexto externo, a RAG assegura respostas mais atualizadas e aderentes às necessidades do usuário (Tang et al., 2024).

Contudo, a técnica também apresenta desafios, como a complexidade de integração com sistemas e a necessidade de qualidade nos dados de origem, uma vez que dados imprecisos podem comprometer as respostas (XU et al., 2023). A arquitetura RAG deve ser rigorosamente configurada para evitar distorções nas respostas geradas (HUANG et al., 2024).

2.5 Avaliação de Modelos de Recuperação de Dados

2.5.1 Métricas de Avaliação de Modelos de LLM

Para avaliar o desempenho dos LLMs, são usadas métricas como precisão, assertividade e diversidade das respostas, que quantificam a qualidade dos dados recuperados (BANERJEE et al., 2023). Além disso, métricas adicionais, como consistência e relevância, além de degradação de informação, têm sido aplicadas para uma avaliação mais completa, oferecendo uma análise objetiva da eficácia dos modelos em recuperação de dados (SINGH; ZOU, 2024).

2.5.2 Benchmarks e Ferramentas de Avaliação

Frameworks de benchmarks como o Ragas fornecem uma avaliação padronizada do desempenho dos LLMs em tarefas de recuperação de dados, permitindo uma análise comparativa consistente (Yang et al., 2024). Essas ferramentas são essenciais para validar o uso dos LLMs em contextos práticos e garantir a confiabilidade das respostas geradas e consequentemente, assegurar que as informações passadas, não apenas são relevantes para o contexto, mas também seguras para uso prático.

2.5.3 Avaliação de Factualidade e Controle de Alucinações

Mecanismos automatizados de avaliação da factualidade ajudam a controlar a ocorrência de alucinações nos LLMs. Ferramentas como o Open LLM Leaderboard Archive (“Open LLM Leaderboard - a Hugging Face Space by open-llm-leaderboard”, 2025) ajudam a identificar possíveis discrepâncias referentes a cada LLM, aos quais possibilitam promover melhor entendimento e maior confiabilidade nas respostas geradas pelos modelos (Li et al., 2024). Esse controle é essencial em aplicações onde informações incorretas podem ter consequências graves, tais quais meios médicos, industriais, ou até mesmo contextos conversacionais. (HUANG et al., 2024)

2.6 Impacto da Recuperação de Dados Automatizada em Diversos Setores

2.6.1 Pesquisa Acadêmica

A recuperação automatizada de dados pode facilitar a revisão de literatura e análise de dados, já que a mesma consegue acelerar o processo de pesquisa acadêmica através de sintetização e coleta múltipla de fontes de dados. A rapidez e precisão desses dados têm mostrado o potencial de reduzir o tempo de pesquisa e melhorar a qualidade dos estudos , como por exemplo no caso de (Dogucu & Çetinkaya-Rundel, 2020), ao qual propunha uma busca performática utilizando Web-Scraping, para assistir alunos em tarefas de coleta, garantindo que os acadêmicos tivessem mais tempo para focar em tópicos mais relevantes de suas pesquisas.

2.6.2 Inteligência de Negócios e Gestão de Dados

No setor empresarial, a recuperação automatizada de dados permite insights estratégicos e decisões baseadas em dados confiáveis. A extração precisa e contextualizada de informações ajuda a reduzir custos e a melhorar a agilidade das respostas em um ambiente corporativo competitivo, além disso, as LLM's, tem se mostrado promissoras em agregar valor aos negócios de diferentes áreas, o que nesse sentido, foi

estudado por (BRAND; ISRAELI; NGWE, 2023), ao qual propuseram o uso de LLMs para pesquisas de mercado, o que somado a arquitetura RAG, pode permitir centenas de combinações, incluindo contextos com informações privadas.

2.6.3 Impacto em Ferramentas de Atendimento e Assistentes Virtuais

LLMs e a técnica de RAG estão impulsionando o avanço de sistemas de atendimento ao cliente e assistentes virtuais, proporcionando respostas mais precisas e contextualizadas que aprimoram a experiência do usuário através de janelas de contexto, e dados provindos de fontes aos quais não foram usadas no treinamento da respectiva LLM, assim, possibilitando que a LLM, ganhe mais informações para trabalhar, sem a necessidade de um fine-tuning ou re-treinamento. Esse desenvolvimento é essencial para tornar a interação com esses sistemas mais satisfatória e confiável (Zhao, P et al., 2024).

3 Metodologia

A pesquisa realizada neste trabalho é de Natureza Aplicada, pois o objetivo é comparar a eficácia e desempenho de LLMs para uma aplicação prática específica: a recuperação de dados em páginas estáticas.

Quanto à abordagem da pesquisa, esta classifica-se como quantitativa, uma vez que a comparação entre modelos LLM será baseada na análise dos dados originados em experimentos controlados. Serão avaliados quatro modelos open-source Llama 3.1, Dolphin 3, Zephyr e Mistral com capacidade entre 7 e 8 bilhões de parâmetros, disponibilizados gratuitamente via Ollama e Hugging Face. A extração de informações ocorreu em cinco páginas da Wikipédia, sendo quatro em inglês e uma em português brasileiro. Essa escolha justifica-se pelo fato de os modelos de embedding terem sido treinados predominantemente em inglês, de modo que o uso de texto em português pode comprometer a precisão das respostas. Para medir a eficácia na recuperação de dados, foi adotado as seguintes métricas específicas (CHEN et al., 2024), neste sentido, considerando que Noise Robustness, Negative Rejection, Information Integration e Counterfactual Robustness, foram avaliadas por humanos:

Noise Robustness: Capacidade do modelo de manter a performance diante de ruídos ou perturbações nos dados de entrada. Alterna entre 0 e 1, para elucidar a presença do tipo de erro, ao qual, devido a quantidade de perguntas (10), pode obter até 10 pontos no teste.

Negative Rejection: Habilidade do modelo em rejeitar informações irrelevantes ou incorretas; Pode obter até 10 pontos ao fim do teste. Individualmente 1 ponto por pergunta.

Information Integration: Nível de coerência e consistência ao integrar diferentes blocos de informação. Varia entre 0 e 1, podendo somar até 10 pontos ao fim do teste.

Counterfactual Robustness: Estabilidade das respostas frente a variações contrafactuais no conteúdo apresentado. Varia entre 0 e 1, podendo somar até 10 pontos ao fim do experimento;

Faithfulness: Mede o quão consistente uma resposta é, em termos factuais, com o contexto recuperado. Varia de 0 a 1, sendo que valores mais altos indicam maior consistência.

Answer Relevancy: Medida da pertinência da resposta em relação à pergunta, avaliando se o conteúdo gerado atende de forma direta, objetiva e completa ao que foi solicitado, pode variar de 0 a 1.

Em relação aos objetivos, trata-se de uma pesquisa descritiva e explicativa. O caráter descritivo vem da intenção de documentar as características e capacidades dos modelos de LLM analisados, enquanto o caráter explicativo se aplica ao esforço de compreender os fatores que influenciam o desempenho de cada modelo. Isso permitiu uma análise detalhada das variáveis que afetam a recuperação de dados e a eficácia de cada modelo no contexto específico de páginas estáticas.

Finalmente, quanto aos procedimentos técnicos, este trabalho se enquadra como um experimento, onde as LLMs serão testadas em condições controladas para avaliar a eficácia e o desempenho na tarefa de recuperação de dados. A estrutura em código se teve via scripts em Python, ao qual a definição da infraestrutura ocorreu pelo uso da biblioteca LangChain, a instanciação dos modelos via Ollama e a avaliação de métricas automatizadas através da biblioteca Ragas com auxílio de API da OpenAI utilizando o modelo 4.1 nano, para avaliação assistida.

4 Desenvolvimento

4.1 Seleção dos Dados e Modelos

- 4.1.1 Origem e Características das Fontes Utilizadas

  A seleção das páginas utilizadas neste estudo teve como critério principal a diversidade linguística e temática, com o intuito de avaliar o comportamento dos LLMs sob diferentes contextos e idiomas. Foram utilizadas cinco páginas da Wikipédia, sendo quatro em inglês e uma em português brasileiro, conforme descrito a seguir:

  - **pt.wikipedia.org/wiki/Processamento_de_linguagem_natural**

    Esta foi a única página em português utilizada intencionalmente, com o objetivo de avaliar a capacidade dos modelos treinados predominantemente em inglês de lidar com conteúdo em língua portuguesa. Essa decisão permitiu analisar as limitações dos LLMs em relação a textos fora do seu domínio linguístico majoritário.

  - **en.wikipedia.org/wiki/Natural_language_processing**

    Página central sobre o tema de Natural Language Processing (NLP), escolhida por seu conteúdo técnico e detalhado. Representa uma fonte esperada de bom desempenho por parte dos modelos, já que apresenta terminologias e estruturas comuns aos dados de treinamento típicos de LLMs.

  - **en.wikipedia.org/wiki/Brazil**

Embora trate de um tema generalista e geopolítico, esta página foi selecionada para avaliar como os modelos lidam com conteúdos amplamente documentados e acessíveis em bases de dados públicas. Esperava-se aqui uma performance sólida, dada a abrangência do tema.

- **en.wikipedia.org/wiki/GPT-4.5**

  Esta página refere-se a uma versão recente da arquitetura GPT, publicada após o treinamento dos modelos utilizados neste estudo. Por essa razão, presume-se que os modelos não tenham tido acesso a seu conteúdo durante o processo de aprendizagem, o que a torna uma excelente referência para avaliação de capacidade de generalização e inferência contextual.

- **en.wikipedia.org/wiki/DeepSeek_(chatbot)**

  Semelhante ao caso da página do GPT-4.5, o conteúdo desta página foi publicado após o treinamento dos modelos avaliados. Sua escolha visa analisar a robustez dos LLMs diante de informações inéditas, além de investigar o risco de alucinação factual quando confrontados com dados não presentes em seu treinamento original.

Essa composição de fontes oferece um panorama equilibrado entre temas técnicos, generalistas, linguísticos e temporais. Além disso, permite investigar como o idioma, a atualidade da página e o grau de tecnicidade influenciam a precisão, a relevância e a fidelidade das respostas geradas pelos modelos na arquitetura RAG. A presença de fontes que, presumivelmente, não fizeram parte do corpus de treinamento dos modelos é proposital, de modo a avaliar sua habilidade de lidar com situações fora do escopo visto anteriormente, forçando a buscar respostas provenientes de fato das páginas e não do corpus previamente treinado pelo modelo.

- 4.1.2 Critérios para Escolha das Páginas Avaliadas

A escolha das páginas da Wikipédia utilizadas neste estudo foi orientada por três critérios fundamentais: **(i) diversidade temática**, **(ii) variação linguística** e **(iii) temporalidade do conteúdo**. Essa abordagem visa avaliar não apenas o desempenho geral dos modelos, mas também sua sensibilidade a aspectos linguísticos e informacionais distintos.

(i) Diversidade Temática

Buscou-se selecionar páginas com escopos diferentes, desde tópicos técnicos, como *Processamento de Linguagem Natural* (tanto em português quanto em inglês), até temas mais amplos e culturais, como *Brazil*. Isso permitiu analisar como os modelos reagem a conteúdos com graus variados de estrutura, terminologia técnica e contextualização factual. Além disso, foram incluídas páginas voltadas a tecnologias emergentes como *GPT-4.5* e *DeepSeek*, para testar a capacidade dos modelos de responder sobre tópicos que envolvem linguagem especializada e inovação.

(ii) Variação Linguística

A introdução de uma página em português foi proposital, com o objetivo de verificar a capacidade dos LLMs treinados majoritariamente em inglês de lidar com conteúdos em outros idiomas. Essa escolha contribui para avaliar limitações linguísticas e medir o impacto direto do idioma na qualidade das respostas, sendo este um dos objetivos específicos deste trabalho.

(iii) Temporalidade do Conteúdo

As páginas relacionadas a *GPT-4.5* e *DeepSeek* foram selecionadas intencionalmente por terem sido criadas após o período de treinamento dos modelos avaliados. Com isso, foi possível testar o comportamento dos LLMs frente a informações inéditas, e mensurar a ocorrência de alucinações ou inferências não fundamentadas. Essa escolha também permitiu avaliar a efetividade da arquitetura RAG em fornecer contexto factual a partir de fontes externas, mesmo quando o modelo base não possui o conhecimento em seu treinamento.

Esses critérios combinados compõem uma base de avaliação ampla e robusta, permitindo não apenas uma análise comparativa entre os modelos, mas também uma leitura crítica sobre os limites e capacidades da arquitetura RAG frente a conteúdos de diferentes naturezas. A partir dessa curadoria, foi possível gerar perguntas que testassem atributos como relevância, precisão factual, robustez ao ruído e integração contextual, aspectos essenciais para a aplicação de LLMs em tarefas de recuperação de dados.

- 4.1.3 Origem e Características dos LLMs

Neste estudo, foram selecionados quatro modelos de linguagem de grande porte (LLMs), todos open-source e com aproximadamente 7 bilhões de parâmetros, a fim de garantir a viabilidade de execução local e comparabilidade técnica dentro de uma faixa semelhante de complexidade. Os modelos utilizados foram: **Llama 3.1**, **Mistral**, **Dolphin 3** e **Zephyr**.

A escolha por modelos open-source se justifica por sua acessibilidade, transparência e flexibilidade de experimentação, além de representar uma alternativa viável para aplicações locais que demandam controle sobre os dados e infraestrutura. Todos os modelos foram instanciados localmente via *Ollama*, garantindo padronização no ambiente computacional e na interface de acesso às suas capacidades.

A seguir, apresenta-se um breve resumo sobre as origens e principais características de cada modelo avaliado:

- **Llama 3.1**

  Desenvolvido pela Meta AI, o Llama 3.1 é uma versão refinada da série LLaMA (Large Language Model Meta AI). Reconhecido por sua arquitetura baseada em transformers com foco em eficiência contextual e compatibilidade com múltiplos idiomas, o modelo tem se destacado em tarefas de geração e recuperação de informações. A versão utilizada neste trabalho é otimizada para performance em dispositivos com até 24 GB de memória GPU.

- **Mistral**

  Lançado pela Mistral AI, esse modelo tem ganhado notoriedade por seu balanceamento entre performance e custo computacional. O Mistral se apoia em mecanismos avançados de atenção rotativa e janelas de contexto ampliadas, o que o torna promissor para tarefas de recuperação em contextos densos. Seu desempenho robusto em benchmarks recentes justificou sua inclusão no comparativo.

- **Dolphin 3**

  O Dolphin 3 é um modelo derivado de abordagens de fine-tuning sobre bases generalistas, com especial atenção para instruções e alinhamento conversacional. Sua arquitetura permite respostas coerentes mesmo com perguntas abertas ou vagas, o que o torna interessante para recuperação de dados em páginas com conteúdo pouco estruturado ou ambíguo.

- **Zephyr**

  Desenvolvido como uma alternativa de código aberto altamente otimizada, o Zephyr tem como foco a precisão factual e a clareza na geração de texto. Ele é conhecido por seu bom desempenho em tarefas de QA (Question Answering), com mecanismos de atenção refinados e priorização da consistência entre diferentes blocos de informação.

Seu uso aqui visa avaliar sua capacidade de integração semântica em contextos múltiplos.

- 4.1.4 Critérios para Escolha dos LLMs

A escolha dos modelos de linguagem utilizados neste estudo foi orientada por critérios técnicos, práticos e metodológicos, com o objetivo de garantir um comparativo justo, acessível e representativo dentro do contexto de recuperação de dados com arquitetura RAG. Os quatro modelos selecionados **Llama 3.1**, **Mistral**, **Dolphin 3** e **Zephyr** foram escolhidos com base nos seguintes critérios:

(i) Código aberto e disponibilidade local

Todos os modelos são open-source, o que permite sua execução local sem necessidade de acesso a APIs pagas ou infraestrutura proprietária. Essa característica foi essencial para garantir a independência experimental, maior controle sobre as execuções e conformidade com ambientes on-premise, onde o envio de dados sensíveis para servidores externos não é viável.

(ii) Tamanho compatível (7–8 bilhões de parâmetros)

Foi estabelecido um limite de parâmetros em torno de 7B–8B para manter a paridade entre os modelos e possibilitar a execução eficiente em hardware local, mais especificamente em uma GPU RTX 3090 com 24 GB de memória. Essa escolha também reflete uma realidade prática de aplicações empresariais e acadêmicas que não contam com acesso a clusters de alto desempenho.

(iii) Popularidade e desempenho recente em benchmarks

Todos os modelos escolhidos têm apresentado bom desempenho em benchmarks recentes de QA (Question Answering), geração condicional e RAG. Além disso, cada modelo representa diferentes abordagens de design e pré-treinamento, o que enriquece o comparativo. Por exemplo, o Dolphin 3 é conhecido por seu alinhamento conversacional, enquanto o Zephyr foca em precisão factual.

(iv) Diversidade arquitetural e filosófica

Os modelos foram selecionados para representar distintas abordagens de projeto: desde a robustez arquitetônica da Meta (Llama 3.1), passando pela eficiência de implementação da Mistral AI, até os ajustes finos voltados à conversação (Dolphin

3) e à factualidade (Zephyr). Essa diversidade contribui para uma análise mais abrangente sobre como diferentes estratégias impactam a recuperação de dados.

(v) Suporte à integração com LangChain e RAGAS

Todos os modelos apresentam compatibilidade com bibliotecas de orquestração como LangChain e sistemas de avaliação como RAGAS, facilitando sua integração à pipeline experimental adotada. Esse fator também garantiu uma uniformidade no processo de teste e coleta de métricas, evitando tratamentos diferenciados que comprometessem a validade dos resultados.

Esses critérios visaram não apenas a obtenção de resultados comparáveis, mas também a elaboração de recomendações realistas para projetos que desejam implementar RAG com modelos open-source. A escolha cuidadosa dos LLMs permitiu assegurar que as diferenças observadas nos resultados fossem oriundas das capacidades dos modelos, e não de variações no ambiente ou configuração experimental.

4.2 Preparação e Processamento dos Dados

- 4.2.1 Coleta e Extração do Conteúdo

A coleta do conteúdo textual das páginas selecionadas foi realizada de forma automatizada, utilizando a biblioteca **LangChain**, amplamente adotada em arquiteturas RAG. Para este fim, empregou-se o módulo UnstructuredURLLoader, que permite extrair o corpo textual de uma página web a partir de seu endereço (URL), convertendo-o diretamente em documentos utilizáveis na etapa de geração de embeddings.

O UnstructuredURLLoader possui integração com o pacote unstructured, que realiza o parsing e a segmentação inteligente de diferentes componentes do conteúdo web (títulos, parágrafos, listas, etc.), preservando a estrutura lógica do texto. Esse mecanismo foi crucial para garantir que a informação extraída mantivesse coesão e relevância, aspectos fundamentais para a etapa posterior de indexação vetorial e geração de respostas.

Durante o processo de extração, foram utilizadas as cinco páginas previamente descritas, sendo cada uma carregada individualmente e convertida em um objeto do tipo Document, conforme especificado pelo LangChain. Os documentos resultantes foram armazenados temporariamente em memória para posterior fragmentação e geração de embeddings. Assim garantindo que o embedder, teria apenas uma fonte para busca, mantendo os vetores específicos da página atual a ser testada.

A opção por realizar a extração diretamente da web, e não a partir de arquivos locais ou HTML pré-processados, teve como objetivo preservar a integridade e a atualidade do conteúdo, assegurando que os modelos fossem expostos exatamente ao mesmo material que seria acessado por um usuário comum. Esse fator foi particularmente relevante para as páginas mais recentes, como *GPT-4.5* e *DeepSeek*, cujos conteúdos refletem tecnologias emergentes e foram incluídos para avaliar a capacidade dos LLMs em lidar com dados inéditos.

Por fim, vale ressaltar que nenhuma modificação manual foi feita no conteúdo coletado, garantindo assim um cenário experimental realista e não enviesado. A extração automatizada e padronizada assegurou que todas as fontes fossem tratadas com equidade, permitindo uma comparação justa do desempenho dos modelos frente aos mesmos dados brutos.

- 4.2.2 Divisão e Organização em Chunks (Fragmentos)

  Após a extração das páginas, os documentos foram submetidos a um processo de segmentação em fragmentos menores, conhecidos como *chunks*. Essa etapa é essencial em pipelines com arquitetura RAG, pois os modelos de recuperação e geração obtêm melhor desempenho ao operar sobre blocos compactos e semanticamente coesos de informação.

  Para essa tarefa, foi utilizada a biblioteca **LangChain**, empregando-se a classe CharacterTextSplitter, configurada com um chunk_size de **512 caracteres** e um chunk_overlap de **50 caracteres**, com o separador definido como espaço " ". Essa configuração foi escolhida com base em testes preliminares, buscando um equilíbrio entre granularidade e contextualização, permitindo a criação de blocos suficientemente informativos sem comprometer a performance da recuperação vetorial.

  A sobreposição de 50 caracteres entre os chunks é especialmente importante para preservar a continuidade semântica entre fragmentos adjacentes. Isso reduz o risco de perda de contexto em pontos de corte, garantindo que informações relevantes não fiquem isoladas em bordas de blocos e possam ser corretamente associadas no momento da geração da resposta pelo modelo.

  Cada documento extraído foi, assim, dividido em múltiplos chunks de 512 caracteres, que passaram a compor o corpus vetorial utilizado na etapa de indexação e recuperação. Esses fragmentos funcionam como unidades de consulta semântica, representando os trechos mais relevantes do conteúdo original para serem comparados com a pergunta do usuário.

  A escolha de um tamanho de chunk menor, como o adotado neste experimento, busca mitigar os riscos associados a blocos excessivamente longos que tendem a diluir o foco semântico e a blocos muito curtos que podem conter informação insuficiente para gerar uma resposta

completa. Ao final dessa etapa, o corpus estava devidamente estruturado, com os chunks prontos para serem transformados em embeddings e utilizados pelo banco vetorial na recuperação orientada por similaridade.

- 4.2.3 Geração dos Embeddings

  Com os documentos devidamente divididos em chunks, a próxima etapa consistiu na geração dos **vetores de embeddings**, que representam semanticamente cada fragmento de texto em um espaço de alta dimensão. Esses vetores são essenciais para o funcionamento da arquitetura RAG, pois permitem calcular similaridades entre a pergunta do usuário e os conteúdos presentes no corpus, orientando a recuperação de contexto mais relevante para a resposta final.

  Para a geração dos embeddings, foi utilizado o modelo **sentence-transformers/all-MiniLM-L6-v2**, amplamente reconhecido por seu bom desempenho em tarefas de similaridade textual com baixo custo computacional. O modelo, hospedado na plataforma Hugging Face, foi integrado à pipeline por meio do conector da **LangChain**, utilizando a classe HuggingFaceEmbeddings.

  Esse modelo específico foi escolhido por reunir as seguintes vantagens:

  - **Desempenho competitivo em benchmarks de recuperação de informação**, mesmo sendo consideravelmente menor do que encoders mais robustos como BERT-base ou MPNet.
  - **Latência reduzida e baixa exigência de memória**, o que o torna adequado para aplicações locais e em larga escala.
  - **Compatibilidade nativa com a biblioteca LangChain**, permitindo integração fluida e rápida com os demais componentes da pipeline.

  Cada chunk gerado anteriormente foi transformado em um vetor denso de 384 dimensões, correspondente à representação embutida produzida pelo MiniLM. Esses vetores foram, então, armazenados em um banco vetorial baseado na biblioteca **chromadb**, preparado para realizar buscas por similaridade utilizando distância cosseno como métrica.

  O uso de embeddings precisos e semanticamente representativos é crucial para garantir que o processo de recuperação selecione os fragmentos mais coerentes com a pergunta formulada. Fragmentos com embeddings mal formados ou semanticamente imprecisos podem resultar em resgates irrelevantes, prejudicando a fidelidade e a relevância das respostas produzidas pelo LLM.

Ao final desta etapa, a base de dados vetorial estava completamente indexada, pronta para ser consultada pelos modelos de linguagem no momento da geração das respostas dentro da arquitetura RAG.

4.3 Ambiente Experimental

- 4.3.1 Configuração dos Modelos de Linguagem (LLMs)

  Para garantir a equidade na avaliação e eliminar interferências causadas por configurações diferenciadas, todos os modelos de linguagem utilizados neste estudo foram submetidos aos mesmos parâmetros de geração durante os testes. A uniformização desses parâmetros é essencial para assegurar que as comparações entre os LLMs reflitam suas competências intrínsecas, e não variações externas de temperatura, amostragem ou políticas de truncamento.

  As seguintes configurações foram adotadas para todos os modelos:

  - **Temperatura (temperature) = 0.1**

    Um valor de temperatura mais baixo foi escolhido para promover respostas mais determinísticas e coerentes, reduzindo a aleatoriedade do modelo. Isso foi especialmente importante para garantir que o foco da análise permanecesse na capacidade de recuperação e integração de informações, e não na criatividade textual do modelo.
  - **Top-K (top_k) = 30**

    O uso de top_k define o número de tokens mais prováveis considerados na amostragem. Com um valor moderado como 30, buscou-se manter um equilíbrio entre controle e flexibilidade, permitindo que o modelo considere variações relevantes sem comprometer a consistência das respostas.
  - **Top-P (top_p) = 0.8**

    Também conhecida como “nucleus sampling”, essa técnica complementa o top-k, garantindo que apenas os tokens com probabilidade cumulativa de até 80% sejam considerados na geração. Isso reforça o controle sobre a saída textual, filtrando respostas que fogem ao domínio semântico da consulta.

  A execução dos modelos foi realizada por meio do **servidor Ollama**, ferramenta que permite a instância local e simplificada de LLMs compatíveis, como Llama 3.1, Mistral, Zephyr e Dolphin 3. A interface do Ollama provê consistência na chamada dos modelos e facilita a integração com ferramentas de orquestração como LangChain.

Essas configurações foram mantidas fixas ao longo de todos os testes, tanto para garantir a reprodutibilidade dos resultados quanto para permitir uma análise precisa das diferenças de desempenho entre os modelos. Qualquer variação nos resultados, portanto, é atribuída às capacidades do próprio modelo e não a alterações na forma como eles geram ou interpretam o texto.

- 4.3.2 Configuração do Banco de Vetores (Chroma)

  Para a etapa de recuperação de informações, foi utilizado o **Chroma**, uma biblioteca de armazenamento vetorial leve e eficiente, que se destaca por sua integração direta com LangChain e facilidade de uso em ambientes locais. O Chroma atuou como banco de vetores responsável por armazenar os embeddings previamente gerados e por retornar os fragmentos mais relevantes a cada nova consulta.

  A configuração do Chroma foi padronizada em todos os testes, com destaque para o seguinte parâmetro:

  - **Top-K = 3**

    A recuperação foi configurada para retornar os **três chunks mais semanticamente próximos** da consulta (pergunta) feita ao sistema. Esse valor foi definido com base em práticas consolidadas na literatura sobre RAG e teve como objetivo manter o foco em contextos altamente relevantes, sem sobrecarregar o modelo com excesso de informação.

  A métrica de similaridade utilizada para determinar os chunks mais relevantes foi a **distância cosseno**, que mede a proximidade angular entre os vetores de embedding da pergunta e dos documentos. Essa métrica é amplamente aceita em tarefas de recuperação textual, por capturar similaridades semânticas mesmo quando a estrutura sintática diverge.

  O banco vetorial foi mantido em memória durante toda a execução dos testes, garantindo desempenho e agilidade nas requisições. Cada chunk armazenado foi indexado com um identificador único, juntamente com metadados como a origem da página e a posição relativa no documento original informações úteis durante o processo de auditoria e análise qualitativa das respostas.

  A escolha do Chroma, por sua leveza e eficiência, mostrou-se adequada ao escopo do trabalho, que exigia múltiplas consultas por modelo em diferentes contextos, sem comprometer a capacidade computacional da máquina local. Além disso, sua integração nativa com

LangChain permitiu a construção de uma pipeline fluida, desde a geração dos embeddings até a recuperação dos contextos usados pelos modelos LLMs.

- 4.3.3 Detalhes Técnicos do Ambiente Computacional Utilizado

  Os experimentos deste trabalho foram conduzidos em ambiente computacional local, configurado para suportar de forma eficiente modelos de linguagem de médio porte (7B–8B parâmetros) e operações intensivas de indexação vetorial. A máquina utilizada contava com **Windows 11 Pro**, o que possibilitou compatibilidade com ferramentas nativas do ecossistema Python e execução do Ollama.

  A infraestrutura incluía uma **GPU NVIDIA GeForce RTX 3090 com 24 GB de memória GDDR6X**, responsável por permitir o carregamento e inferência local dos modelos sem necessidade de processamento em nuvem, além de **32 GB de memória RAM DDR4**, suficientes para suportar as operações de todo o pipeline que foi desenvolvido em **Python 3.12.3**, com bibliotecas executadas em ambiente virtual isolado, incluindo LangChain, Chroma, Hugging Face Transformers e Sentence Transformers, garantindo estabilidade e reprodutibilidade nos testes.

4.4 Metodologia de Avaliação

- 4.4.1 Definição das Métricas (Faithfulness e Answer Relevancy)

  A avaliação da qualidade das respostas geradas pelos modelos de linguagem neste estudo foi conduzida com base em duas métricas centrais: **Faithfulness** e **Answer Relevancy**. Ambas foram selecionadas por sua relevância no contexto de tarefas de recuperação aumentada por geração (RAG), especialmente quando se busca garantir não apenas a coerência das respostas, mas também sua aderência factual ao conteúdo recuperado.

  - **Faithfulness** refere-se à fidelidade da resposta em relação ao contexto fornecido. Essa métrica avalia se as informações presentes na resposta são, de fato, fundamentadas nos chunks resgatados pelo sistema, sem adições, distorções ou inferências infundadas. Respostas que se mantém estritamente dentro do escopo do conteúdo recuperado são consideradas mais “fiéis”, enquanto aquelas que extrapolam ou introduzem elementos não sustentados são penalizadas. A pontuação varia entre 0 e 1, sendo que valores mais altos indicam maior consistência com a base factual fornecida.
  - **Answer Relevancy**, por sua vez, avalia a pertinência da resposta em relação à pergunta formulada. Essa métrica considera se a resposta aborda diretamente o que foi questionado, oferecendo conteúdo informativo, claro e relevante. Respostas vagas,

genéricas ou que desviam do foco da pergunta obtêm pontuações menores. Assim como a métrica anterior, sua escala varia de 0 a 1.

A escolha por essas métricas se deve à sua capacidade de representar aspectos complementares da performance de um sistema RAG. Enquanto Faithfulness avalia a integridade factual da resposta, Answer Relevancy foca na utilidade prática da informação fornecida. Juntas, oferecem uma visão abrangente da eficácia do modelo tanto do ponto de vista técnico quanto do ponto de vista da experiência do usuário final.

Essas métricas foram operacionalizadas por meio do framework **RAGAS**, que automatiza a avaliação com o suporte de um modelo auxiliar da OpenAI para julgamento das respostas neste caso, o modelo **GPT-4.1 nano**, utilizado como revisor imparcial durante a etapa de validação.

5 Resultados e Discussão

5.1 Comparação das Métricas de Desempenho entre os Modelos

A avaliação comparativa dos quatro modelos considerou métricas relacionadas à robustez e à qualidade das respostas, com foco em seis capacidades essenciais para sistemas RAG: robustez a ruído, recusa diante da ausência de informação (negative rejection), integração de informações, resistência a contrafactuais, fidedignidade (faithfulness) e relevância da resposta (answer relevancy).

Cada métrica analisou uma habilidade específica: a robustez a ruído mede a capacidade de ignorar conteúdos irrelevantes; a recusa negativa indica se o modelo evita responder quando não há suporte nos documentos; a integração de informações avalia se o modelo combina dados de diferentes trechos de forma eficaz; e a robustez a contrafactuais verifica a resistência a informações incorretas. A fidedignidade identifica se a resposta está fiel ao conteúdo fornecido, enquanto a relevância mede o quão diretamente a resposta atende à pergunta.

Nos critérios de robustez, o Llama 3.1 foi o mais consistente. Ele obteve média de 4,2 pontos em robustez a ruído e 5,4 em recusa negativa, superando Dolphin 3 (3,4 e 3,2), Mistral (3,2 e 3,0) e Zephyr (3,0 e 2,6). Esses resultados indicam que o Llama 3.1 foi o mais resistente a distrações e o mais capaz de evitar respostas sem suporte.

Na integração de informações, Dolphin 3 se destacou com média de 6,8 pontos, enquanto os demais ficaram entre 6,0 e 6,2. Já na resistência a contrafactuais, o melhor desempenho foi do Mistral (6,4), seguido por Llama 3.1 e Zephyr (6,2), com Dolphin 3 atrás (5,0). A soma dessas quatro métricas

de robustez gera um score máximo teórico de 40 pontos. O Llama 3.1 alcançou a maior pontuação total (21,8), seguido por Mistral (18,6), Dolphin 3 (18,4) e Zephyr (18,0).

Quanto à qualidade das respostas, todos os modelos obtiveram altos níveis de fidedignidade, com médias superiores a 82%. O Mistral foi o mais fiel (87,6%), seguido de Llama 3.1 (82,7%), Zephyr (82,5%) e Dolphin 3 (82,4%). Esses valores indicam que, na maioria das vezes, as respostas dos modelos refletiram corretamente o conteúdo dos documentos, com poucas alucinações.

Já em relação à relevância, houve maior variação. Dolphin 3 teve o melhor desempenho (0,743), seguido por Llama 3.1 (0,668), Zephyr (0,530) e Mistral (0,494). Isso mostra que, apesar da fidelidade, Zephyr e Mistral tiveram dificuldade em apresentar respostas diretamente relacionadas à pergunta. O Dolphin 3 demonstrou melhor alinhamento entre pergunta e resposta, enquanto o Llama 3.1 ofereceu equilíbrio entre relevância e fidelidade.

Em resumo, Llama 3.1 apresentou a melhor robustez geral, sendo mais confiável em contextos com ruído ou ausência de dados. Dolphin 3 destacou-se por fornecer respostas diretas e pertinentes. Mistral foi o mais fiel ao conteúdo, embora menos assertivo, e Zephyr teve desempenho inferior nos critérios avaliados. Os resultados mostram que os modelos têm perfis distintos, com trade-offs entre fidelidade, precisão e completude, reforçando a importância de escolhas baseadas nos requisitos específicos de cada aplicação.

5.2 Tempo de Resposta e Relação com a Qualidade

Foi avaliado o tempo médio de resposta de cada modelo para verificar sua eficiência computacional e sua relação com a qualidade das respostas. Todos os modelos responderam às mesmas consultas sob condições idênticas de hardware, e o tempo médio por resposta foi calculado com base em cinco execuções por página.

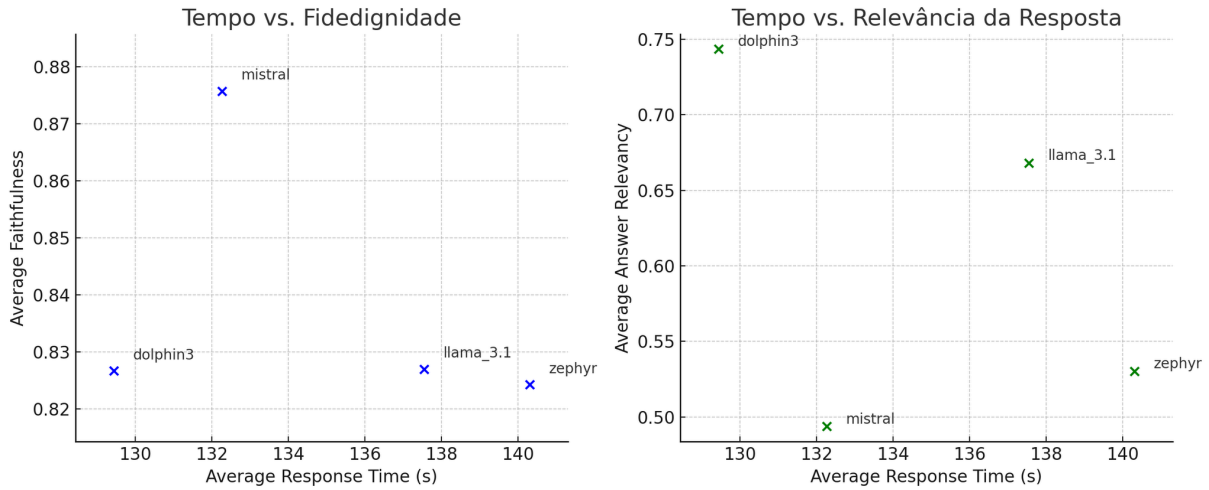
O Dolphin 3 foi o modelo mais rápido, com média de 129,4 segundos por resposta. Em seguida vieram o Mistral (132,3 s), o Llama 3.1 (137,5 s) e, por último, o Zephyr, com a maior latência (140,3 s). Apesar dessas diferenças de tempo, não houve correlação direta entre velocidade e qualidade. O Dolphin 3, mesmo sendo o mais rápido, obteve a maior relevância média (~0,74). Já o Zephyr, o mais lento, apresentou a pior relevância (~0,53). Isso demonstra que mais tempo de processamento não garantiu respostas melhores.

Em termos de fidedignidade, três modelos, Llama 3.1, Dolphin 3 e Zephyr apresentaram valores muito próximos (entre 0,82 e 0,83), enquanto o Mistral, com tempo intermediário, foi o mais fiel, com ~0,88. Isso reforça que a fidelidade está mais ligada ao comportamento interno do modelo do que ao tempo gasto para responder.

Quanto à relevância, houve maior variação. O Dolphin 3 foi o mais relevante, o Mistral ficou com um dos piores resultados (~0,50), apesar de ter tempo semelhante ao Dolphin, e o Llama 3.1 teve uma relevância moderada (~0,67).

Esses dados indicam que não há um padrão que relacione desempenho temporal à qualidade da resposta. A arquitetura e o treinamento de cada modelo parecem ter mais influência do que a velocidade de execução. Em termos práticos, o Dolphin 3 demonstrou ser o mais eficiente, entregando respostas rápidas e de alta qualidade, enquanto o Zephyr teve o pior desempenho combinado, sendo lento e com baixa relevância.

Por fim, embora as diferenças de tempo não tenham sido drásticas (todos entre 2 e 3 minutos por lote de 10 respostas), em aplicações reais, mesmo pequenas variações podem impactar a experiência do usuário. Dentro deste escopo experimental, no entanto, as diferenças foram discretas, e o Dolphin 3 se destacou como o modelo com melhor relação entre tempo e qualidade.



Fonte: Elaborado pelo autor

5.3 Desempenho Técnico e Limitações de Cada Modelo

Com base nos resultados quantitativos, pode-se discutir o desempenho técnico individual de cada modelo, ressaltando seus pontos fortes, limitações observadas e os impactos disso em suas respostas:

Llama 3.1 destacou-se por sua robustez a ruído e pela habilidade em reconhecer perguntas não respondíveis, obtendo o melhor desempenho na métrica Negative Rejection. Demonstrou boa fidelidade às fontes e resistência a contrafactuais, embora com respostas por vezes

conservadoras ou genéricas, o que pode ter afetado sua relevância média. É ideal para cenários que priorizam precisão factual e confiabilidade.

Dolphin 3 foi o modelo com maior relevância de resposta e eficiência na integração de informações, conseguindo reunir trechos dispersos com precisão. Foi também o mais rápido, sugerindo vantagens operacionais. No entanto, apresentou vulnerabilidade a ruídos e conteúdos enganosos, além de menor capacidade de reconhecer perguntas sem resposta, o que pode levar a respostas com informações não fundamentadas. Sua abordagem mais agressiva privilegia completude, mas com maior risco de alucinações.

Mistral obteve a maior fidelidade (faithfulness), evitando adicionar informações não presentes nas fontes e sendo bastante resistente a contrafactuais. Seu ponto fraco foi a baixa relevância das respostas, muitas vezes limitando-se a trechos fiéis mas pouco assertivos ou completos. Além disso, respostas genéricas em perguntas sem base documental também foram observadas. O modelo é adequado quando a exatidão textual é mais importante do que a clareza na resposta.

Zephyr apresentou o desempenho mais fraco e inconsistente. Foi o modelo com menor capacidade de rejeitar perguntas sem resposta, o que resultou em baixa fidelidade e relevância. Além disso, mostrou fragilidade frente a ruído e contrafactuais, além de alta latência, sem benefícios evidentes em qualidade. Seu uso exige melhorias substanciais, especialmente em verificação pós-geração e foco contextual.

6 Limitações e Ameaças a Validade

Este estudo apresenta limitações que restringem a generalização dos resultados. Foram analisadas apenas cinco páginas estáticas com número reduzido de perguntas, o que pode não refletir a variedade de cenários encontrados em aplicações reais. Páginas com formatos distintos ou conteúdos mais complexos podem impactar o desempenho dos modelos de forma diferente.

Além disso, foram testados apenas LLMs de código aberto; modelos proprietários podem apresentar comportamentos distintos. A avaliação combinou métricas automáticas e humanas, mas não substitui análises qualitativas mais detalhadas, como julgamentos especializados.

O experimento também assumiu a presença de documentos corretos, desconsiderando falhas no mecanismo de busca o que, na prática, pode afetar significativamente os resultados.

Para estudos futuros, recomenda-se ampliar o escopo (mais páginas, domínios e perguntas), incluir LLMs de diferentes portes, e aprimorar a integração entre recuperação e geração com técnicas

como reranqueamento, prompting avançado e filtragem de contexto. Também seria útil avaliar aspectos como toxicidade, coerência narrativa e realizar testes cegos com avaliadores humanos para validar a experiência de uso de forma mais realista.

7 Conclusão

Esta pesquisa atingiu seus objetivos ao oferecer uma análise comparativa detalhada do desempenho de modelos LLM em uma arquitetura RAG aplicada a páginas estáticas. Os resultados demonstraram que cada modelo apresenta um perfil técnico distinto, reforçando a ideia de que não há uma solução única ideal: robustez e relevância nem sempre caminham juntas. Um modelo pode ser altamente fiel ao contexto, mas pouco assertivo; outro pode oferecer respostas mais objetivas, porém com maior risco de lapsos factuais.

Encontrar o equilíbrio entre esses extremos é um desafio ainda em aberto. No entanto, estudos como este ajudam a mapear com mais precisão as forças e limitações de cada abordagem. Na prática, os dados indicam que, quando a prioridade é a confiabilidade factual, modelos como o **Llama** se destacam especialmente se combinados com ajustes linguísticos. Já quando se busca concisão e foco na resposta, modelos como o **Dolphin** são promissores, desde que respaldados por mecanismos de verificação pós-geração.

Além disso, reforça-se a importância da arquitetura RAG como ferramenta para ampliar o alcance dos LLMs, fornecendo contexto relevante e reduzindo lacunas de conhecimento. No entanto, essa abordagem também exige a consideração de novas métricas e desafios como robustez a ruído, integração de múltiplas evidências e detecção de perguntas não respondíveis que vão além das avaliações tradicionais de QA.

Espera-se que este trabalho sirva como base para estudos mais amplos e aprofundados no futuro. As sugestões levantadas desde o desenvolvimento de modelos mais equilibrados até a calibração de comportamentos, como a capacidade de reconhecer quando não há resposta possível têm o potencial de contribuir significativamente para a construção de sistemas de pergunta e resposta mais confiáveis, eficientes e úteis em contextos reais de aplicação.

REFERÊNCIAS

AGGARWAL, L. et al. Analyzing Chatgpt Based on Large Language Model from Industrial Perspective. Disponível em: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4563696.

ALBERTS, I. L. et al. Large language models (LLM) and ChatGPT: what will the impact on nuclear medicine be? European Journal of Nuclear Medicine and Molecular Imaging, v. 50, 9 mar. 2023.

ALMEIDA, F.; XEXÉO, G. Word Embeddings: A Survey. [S.l: s.n.]. Disponível em: https://arxiv.org/pdf/1901.09069.

BANERJEE, D. et al. Benchmarking LLM powered Chatbots: Methods and Metrics. Disponível em: http://arxiv.org/abs/2308.04624.

BORGEAUD, S. et al. Improving language models by retrieving from trillions of tokens. Disponível em: https://arxiv.org.

BRAND, J.; ISRAELI, A.; NGWE, D. Using GPT for Market Research. SSRN Electronic Journal, 2023.

CHANG, Y. et al. A Survey on Evaluation of Large Language Models. ACM Transactions on Intelligent Systems and Technology, v. 15, n. 3, 23 jan. 2024.

CHEN, J. et al. Benchmarking Large Language Models in Retrieval-Augmented Generation. Proceedings of the AAAI Conference on Artificial Intelligence, v. 38, n. 16, p. 17754–17762, 24 mar. 2024

DOGUCU, M.; ÇETINKAYA-RUNDEL, M. Web Scraping in the Statistics and Data Science Curriculum: Challenges and Opportunities. Journal of Statistics Education, p. 1–11, 4 ago. 2020.

ES, S. et al. RAGAS: Automated Evaluation of Retrieval Augmented Generation. [S.l: s.n.]. Disponível em: https://aclanthology.org/2024.eacl-demo.16.pdf. Acesso em: 26 mar. 2025.

HUANG, L. et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. ACM Transactions on Office Information Systems, 20 nov. 2024.

KASNECI, E. et al. ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, v. 103, n. 102274, 1 abr. 2023.

KHDER, M. Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application. International Journal of Advances in Soft Computing and its Applications, v. 13, n. 3, p. 145–168, 28 nov. 2021.

LANGCHAIN. Disponível em: https://www.langchain.com/.

LI, Z. et al. Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach. Disponível em: https://arxiv.org/abs/2407.16833. Acesso em: 5 ago. 2024.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST). TREC: Text REtrieval Conference. Disponível em: https://trec.nist.gov/.

PYTHON. Python. Disponível em: https://www.python.org/.

SINGH, K.; ZOU, J. New Evaluation Metrics Capture Quality Degradation due to LLM Watermarking. Disponível em: https://openreview.net/forum?id=PuhF0hyDq1. Acesso em: 24 nov. 2024.

TANG, Q. et al. QUILL: Query Intent with Large Language Models using Retrieval Augmentation and Multi-stage Distillation. Disponível em: https://arxiv.org/abs/2210.15718.

TANG, Q. et al. Self-Retrieval: Building an Information Retrieval System with One Large Language Model. Disponível em: https://arxiv.org/abs/2403.00801.

VASWANI, A. et al. Attention Is All You Need. [S.l: s.n.]. Disponível em: https://arxiv.org/pdf/1706.03762.

VIJAYARAGAVAN PICHIYAN et al. Web Scraping using Natural Language Processing: Exploiting Unstructured Text for Data Extraction and Analysis. Procedia Computer Science, v. 230, p. 193–202, 1 jan. 2023.

WEIGHTS & BIASES. Evaluating Large Language Models (LLMs) with Eleuther AI. Disponível em: https://wandb.ai.

XU, P. et al. Retrieval meets Long Context Large Language Models. arXiv (Cornell University), 4 out. 2023.

YANG, Y. et al. Can We Delegate Learning to Automation?: A Comparative Study of LLM Chatbots, Search Engines, and Books. Disponível em: https://export.arxiv.org/abs/2410.01396. Acesso em: 8 out. 2024.

YU, H. et al. Evaluation of Retrieval-Augmented Generation: A Survey. [S.l: s.n.]. Disponível em: https://arxiv.org/pdf/2405.07437.

ZHANG, P. et al. Retrieve Anything To Augment Large Language Models. Disponível em: https://arxiv.org/abs/2310.07554.

ZHAO, P. et al. Retrieval-Augmented Generation for AI-Generated Content: A Survey. Disponível em: https://arxiv.org/abs/2402.19473v1. Acesso em: 7 out. 2024.

ZHAO, W. X. et al. A Survey of Large Language Models. arXiv:2303.18223 [cs], 31 mar. 2023.