CESUPA

Ementa do Primeiro Projeto de Implementação do 2° Bimestre

(AV2): Mecanismos de Atenção e Arquiteturas Transformer

Sistema LLM + RAG para Consulta a Documentos Oficiais sobre Dengue

BELÉM

2025

CAUÊ MARTINS

GABRIEL ANTONIO

GABRIEL GALEGO

JOÃO VITOR FARIAS

MURILO GUIMARÃES

WALLACE RICARDO

Resumo

Este trabalho apresenta o desenvolvimento de um sistema de Recuperação Aumentada por Geração (RAG) aplicado ao domínio da dengue no Brasil, utilizando documentos oficiais do Ministério da Saúde como fonte primária. O pipeline implementado combina técnicas de recuperação lexical e semântica, re-ranqueamento vetorial e engenharia de prompts estruturados para aprimorar a exatidão e a confiabilidade das respostas. Avaliamos cinco configurações distintas, baseline sem recuperação, dois sistemas somente vetoriais, um híbrido BM25+embeddings, e uma versão otimizada com few-shot e prompts estruturados, medindo métricas de fidelidade, precisão de contexto e performance baseada em itens clínicos. Os resultados demonstram que as abordagens híbridas e orientadas por exemplos aprimoram significativamente a aderência ao conteúdo dos documentos oficiais e reduzem a incidência de alucinações. O trabalho reforça a aplicabilidade prática de RAG para domínios sensíveis, como saúde pública, onde a acurácia informacional é mandatória.

Palavras-chave: Dengue; Assistente virtual; LLM; RAG; LangChain.

BELÉM

2025

1. INTRODUÇÃO

O uso de grandes modelos de linguagem (LLMs) em aplicações críticas, especialmente na área da saúde, exige mecanismos que reduzam alucinações e mantenham o conteúdo gerado alinhado a fontes verificáveis. Embora modelos como Qwen, LLaMA e GPT apresentem alta fluência, sua capacidade factual permanece limitada quando não há acesso direto a bases documentais relevantes [1]. A técnica de Recuperação Aumentada por Geração (RAG) surge como alternativa essencial para integrar LLMs a corpora específicos, garantindo que as respostas estejam fundamentadas em trechos textuais confirmáveis [2]. No caso da dengue, cuja comunicação pública exige precisão terminológica e aderência às diretrizes clínicas oficiais, modelos não supervisionados podem reproduzir informações incompletas, ambíguas ou incorretas, aumentando riscos no uso prático.

Com base nesses desafios, este projeto implementa uma arquitetura RAG completa, utilizando documentos originais do Ministério da Saúde sobre manejo clínico, vigilância epidemiológica e orientações de prevenção. Sua contribuição consiste em demonstrar empiricamente como diferentes estratégias de recuperação, segmentação textual e engenharia de prompts impactam a qualidade das respostas e o alinhamento do modelo ao conteúdo oficial. O código desenvolvido, estruturado com LangChain, ChromaDB, BM25Retriever e o modelo Qwen2.5-1.5B, permite comparar diretamente as variantes de RAG e avaliar quantitativamente sua fidelidade, proporcionando uma análise rigorosa do impacto de cada componente do pipeline.

2. FUNDAMENTAÇÃO TEÓRICA

A concepção do sistema apoia-se nos princípios de Recuperação Aumentada por Geração, que consiste na combinação entre um módulo de recuperação de documentos e um modelo de linguagem responsável pela síntese final da resposta. O processo envolve, primeiramente, identificar trechos relevantes de documentos por meio de embeddings vetoriais de alta dimensionalidade, os quais permitem medir similaridade semântica entre consultas e conteúdos textuais. A literatura demonstra que RAG reduz significativamente a incidência de erros factuais, já que restringe a base de conhecimento utilizada pelo modelo para trechos textualmente comprováveis. Para isso, são utilizados métodos como BM25, que se baseia em frequência de termos, e modelos de embeddings como sentence-transformers, que capturam relações semânticas profundas.

No contexto deste trabalho, a etapa generativa é realizada com o modelo Qwen2.5-1.5B-Instruct, um LLM open-source direcionado a tarefas de instrução e relativamente leve para execução em ambientes como Google Colab. A escolha por um modelo de arquitetura causal decorre da necessidade de controlar o formato das respostas e garantir que a saída reflita estritamente o conteúdo recuperado. Os prompts estruturados utilizados nas versões mais avançadas (RAG4) representam técnicas de prompt engineering que reforçam a obediência ao contexto e incluem exemplos few-shot, contribuindo para respostas mais consistentes. As métricas de avaliação adotadas — como Context Precision, baseada em similaridade vetorial entre consulta e trechos recuperados, e Faithfulness, que mede se frases da resposta encontram suporte explícito no contexto — são fundamentadas em literatura de avaliação de RAG e visam medir a aderência e veracidade do modelo de forma sistemática.

3. METODOLOGIA

A metodologia foi estruturada em etapas que refletem diretamente o fluxo computacional descrito no código. Cada fase, desde o carregamento dos PDFs até o cálculo de métricas, foi implementada por funções específicas, projetadas para fornecer reprodutibilidade experimental e modularidade.

A primeira etapa consistiu na leitura de documentos oficiais, definidos na lista PDF_URLS, utilizando requisições HTTP. Os arquivos são armazenados, por meio de um download interno, no diretório dengue_pdfs/. Em seguida, PyPDFLoader é utilizado para transformar cada página de cada documento armazenado no diretório em instâncias independentes da classe Document, uma decisão coerente com a arquitetura RAG, já que a granularidade fina facilita a posterior divisão em chunks menores. A checagem automática garante um corpus mínimo de 10 PDFs ou 50 páginas, prevenindo cenários de treinamento com dados insuficientes. A segunda fase envolve segmentação textual por meio das duas instâncias de RecursiveCharacterTextSplitter. O código diferencia explicitamente chunk médio (1200 caracteres) e chunk pequeno (600 caracteres), criando duas bases vetoriais distintas (vstore_medium e vstore_small). Chunks médios tendem a capturar mais contexto clínico, especialmente em trechos densos de manuais de manejo, enquanto chunks menores facilitam a precisão na recuperação para consultas mais específicas. Essa decisão é crítica, pois as configurações RAG1, RAG2 e RAG3 dependem diretamente da granularidade definida.

A terceira etapa é responsável pela construção dos embeddings por meio da classe HuggingFaceEmbeddings. O modelo escolhido, paraphrase-multilingual-MiniLM-L12-v2, é otimizado para rapidez e possui suporte explícito à língua portuguesa, garantindo compatibilidade com textos médicos nacionais. Os embeddings resultantes são persistidos em instâncias independentes de ChromaDB, permitindo experimentações paralelas. Além disso, o código implementa um reranqueador manual baseado em similaridade coseno, operacionalizado nas funções embed_query, embed_texts, cosine_sim e principalmente rerank_by_embedding(), que redefine dinámicamente a ordem dos documentos com base na relevância. A etapa subsequente integra o componente lexical BM25Retriever, que opera sobre os docs_small. O sistema híbrido é então implementado por meio das funções hybrid_candidates() e hybrid_with_rerank(), que combinam resultados de BM25 e do retriever vetorial, removem duplicatas com base em hashing de conteúdo e aplicam reranqueamento final. Essa arquitetura é diretamente responsável pelos ganhos observados na configuração RAG3 e RAG4.

A geração textual é tratada com o modelo Qwen2.5-1.5B, carregado por meio das bibliotecas HuggingFace Transformers. A função to_chat_prompt() prepara o texto de entrada conforme o formato de conversa esperado pelo tokenizer. O pipeline de geração utiliza temperature=0 e do_sample=False, garantindo determinismo. As configurações experimentais são consolidadas na classe RagConfig, com cinco perfis distintos, manipulados dinamicamente pela função run_query(), que controla recuperação, formatação do prompt, geração da resposta e cálculo das métricas. Por fim, a avaliação é realizada utilizando as funções detect_items(), prf1(), context_precision_at_k() e faithfulness(). As perguntas de teste e os itens esperados são estruturados manualmente no código, simulando um conjunto de validação semântico específico para o domínio da dengue. O resultado é sintetizado nas tabelas geradas por summary_df e no gráfico exibido ao final do notebook.

4. RESULTADOS

Os resultados mostram diferenças marcantes entre as cinco configurações. O baseline apresenta, como esperado, respostas genéricas e não ancoradas no corpus, reforçando o problema da falta de aderência factual. As métricas de Faithfulness e Context Precision são rotuladas como NaN, pois o baseline não utiliza recuperação, inviabilizando comparação direta. A configuração RAG1, baseada em chunks médios e top-k=5, apresenta boa cobertura contextual, mas ocasionalmente insere trechos excessivamente amplos, dificultando a precisão. RAG2, com chunks menores e top-k=8, permite granularidade maior e maior

frequência de acertos, porém sofre com fragmentação excessiva em perguntas que requerem contexto mais amplo, como manejo clínico.

RAG3 representa um salto qualitativo relevante. A combinação BM25 + vetor e reranqueamento produz melhor alinhamento entre consulta e trechos clínicos específicos. Isso é evidenciado pelo aumento do Context Precision@k, que tende a superar significativamente RAG1 e RAG2, demonstrando que o sistema híbrido localiza melhor definições, listas de sinais de alarme e recomendações formais. RAG4 apresenta os melhores resultados globais. O prompt estruturado, somado ao few-shot, produz respostas mais objetivas, bem segmentadas e com menor margem de interpretação pelo modelo.

Os valores médios de F1, Faithfulness e Context Precision superam todas as demais configurações, especialmente ao listar sinais clínicos, medidas de prevenção e critérios de urgência. Em testes como "Quando procurar atendimento de urgência?", RAG4 identifica itens corretos com maior completude e organização, refletindo melhorias tanto no fluxo generativo quanto no processo de recuperação.

5. DISCUSSÃO

Os resultados confirmam que a arquitetura híbrida é particularmente adequada para o domínio da dengue. O conteúdo médico oficial apresenta muitas listas formais, expressões técnicas fixas e vocabulário padronizado. BM25, por ser sensível a termos específicos, localiza rapidamente trechos como "sinais de alarme" ou "manejo clínico". No entanto, esses trechos nem sempre representam as melhores explicações ou descrições mais ricas semânticas. Os embeddings complementam essa limitação ao recuperar fragmentos que capturam nuances contextuais. A abordagem RAG3 destaca-se porque integra o melhor dos dois mundos, mas ainda assim mostra limitações quando o LLM não possui instruções explícitas de como organizar a informação. Isso é corrigido em RAG4, onde o prompt estruturado e o few-shot restringem o comportamento do modelo, diminuindo significativamente a amplitude das interpretações possíveis. O código mostra que essa restrição é implementada manualmente, indicando que a engenharia de prompt desempenha papel tão importante quanto a recuperação de contexto.

O ranqueamento baseado em similaridade cosseno, associado aos embeddings, é decisivo para evitar que textos excessivamente longos ou irrelevantes dominem o contexto. A função rerank_by_embedding() reordena trechos com base em uma métrica direta entre a consulta e os conteúdos recuperados. Isso diminui ruídos e contribui para respostas mais específicas, especialmente quando o corpus contém páginas extensas com múltiplos temas.

Outro ponto relevante é a influência do tamanho dos chunks. O código demonstra que chunk médio beneficia perguntas complexas, enquanto chunk pequeno melhora consultas extremamente específicas. Esse equilíbrio explica por que o híbrido, que frequentemente recupera trechos de ambos os tamanhos, gera melhores resultados.

Por fim, a avaliação quantitativa implementada no código é simples, porém extremamente funcional. O uso de detecção com regex para itens clínicos não exige modelos adicionais e permite mensurar objetivamente se o LLM capturou os elementos essenciais exigidos em protocolos oficiais.

6. CONCLUSÃO

Este estudo demonstra que a combinação entre RAG híbrido e engenharia de prompts estruturados é altamente eficaz para sistemas de consulta baseados em documentos oficiais de saúde. A versão RAG4 supera com margem significativa as configurações anteriores, apresentando respostas mais completas, factuais e alinhadas às diretrizes brasileiras. O pipeline desenvolvido, totalmente aberto e reprodutível, serve como base sólida para aplicações similares em outras áreas sensíveis, como vigilância epidemiológica, educação em saúde ou orientação clínica preliminar. Conclui-se que RAG não apenas melhora a performance factual dos modelos, mas é indispensável sempre que o domínio exige rigor técnico e textual.

7. REFERÊNCIAS BIBLIOGRÁFICAS

[1] T. B. Brown et al., “Language Models Are Few-Shot Learners,” Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901, 2020.

[2] P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP,” in Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020.

[3] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau, “Okapi at TREC,” in Proceedings of the Text REtrieval Conference (TREC), 1995.

[4] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3980–3990, 2019.

[5] A. Vaswani et al., “Attention Is All You Need,” in Advances in Neural Information Processing Systems, vol. 30, pp. 6000–6010, 2017.