SCIENTIA PLENA 2025

Bibl.IA - Um Agente Inteligente Bíblico

Bibl.IA - Um agente inteligente bíblico

B. F. Vieira[1]; D. S. F. Santos[2]; H. C. Júnior[3]

N. S. S. Júnior[4]; R. N. Andrade[5]

Departamento de Computação, Universidade Federal de Sergipe, 49107-230, Aracaju-SE, Brasil

hendrik@dcomp.ufs.br
(Recebido em 08 de março de 2025)

Resumo:
Resumo: Este projeto explora a aplicação da inteligência artificial generativa, usando modelos de linguagem avançados e embeddings semânticos, para responder perguntas relacionadas à Bíblia. Utilizando LangChain, FAISS e a integração com o modelo OpenAI, o sistema, denominado Bibl.IA, busca e recupera informações relevantes em uma base de conhecimento bíblico estruturada a partir de um arquivo CSV. Avaliações iniciais demonstram uma precisão significativa nas respostas e uma boa qualidade nas interações com os usuários. Contudo, aspectos como otimização da recuperação de contexto e melhorias na relevância das respostas permanecem como oportunidades para futuras pesquisas.

Palavras-chave: Inteligência Artificial Generativa, LangChain, FAISS, Embeddings Semânticos, Bíblia.

Abstract:
This project explores the application of generative artificial intelligence, using advanced language models and semantic embeddings, to answer questions related to the Bible. Leveraging LangChain, FAISS, and integration with the OpenAI model, the system named Bibl.IA retrieves and presents relevant information from a structured biblical knowledge base sourced from a CSV file. Initial evaluations indicate a significant accuracy in responses and high-quality user interactions. However, optimizing context retrieval and improving the relevance of the responses are identified as key areas for future research.

Keywords: Generative Artificial Intelligence, LangChain, FAISS, Semantic Embeddings, Bible.

1. INTRODUÇÃO

Com o avanço das tecnologias de inteligência artificial, em especial os modelos de linguagem de grande escala (LLMs – *Large Language Models*), novas oportunidades têm surgido para o desenvolvimento de aplicações capazes de interagir de maneira natural com seres humanos. Este projeto propõe o desenvolvimento de um chatbot especializado em temas bíblicos, utilizando tecnologias modernas de IA para fornecer respostas contextualmente relevantes, teologicamente coerentes e semanticamente precisas, a partir de perguntas formuladas em linguagem natural.

O sistema faz uso de técnicas como *Retrieval-Augmented Generation* (RAG) para combinar a capacidade de geração dos LLMs com a recuperação de informações de fontes confiáveis e estruturadas, garantindo maior fidelidade às escrituras e referências utilizadas. Para a orquestração do pipeline de processamento e interação, bibliotecas como ***LangChain*** foram adotadas, permitindo modularizar o fluxo de dados entre os componentes do sistema e facilitar a integração com bases de conhecimento externas.

O chatbot também pode ser facilmente expandido com técnicas como *Prompt-Engineering*, *Fine-Tuning* supervisionado (SFT) ou *Reinforcement Learning from Human Feedback* (RLHF), dependendo do nível de controle e personalização almejado.

Ao unir técnicas avançadas de IA com um domínio específico e sensível como o conteúdo bíblico, este trabalho busca não apenas demonstrar a viabilidade técnica da abordagem, mas também propor uma aplicação que respeite as escrituras sagradas cristãs e ao mesmo tempo ofereça uma experiência interativa e acessível para usuários com diferentes níveis de conhecimento religioso.

2. MATERIAL E MÉTODOS

O desenvolvimento do chatbot voltado para temas bíblicos foi realizado com base em uma , utilizando bibliotecas modernas do ecossistema Python voltadas para aplicações de inteligência artificial, recuperação de informações e interação com modelos de linguagem. A seguir, são descritas as ferramentas e metodologias adotadas ao longo do projeto:

2.1. Tecnologias e Bibliotecas Utilizadas

- **LangChain**: Framework central utilizado para a orquestração dos componentes do pipeline de IA. A biblioteca permitiu a criação de cadeias de execução (chains) que integram recuperação de dados, chamada de modelos de linguagem, memória de conversação e manipulação de contextos.

- **LangChain Community**: Utilizada para integrar facilmente com diversos *loaders* de documentos, ferramentas e provedores de modelos, como bases locais ou APIs de LLMs.

- **LangChain OpenAI** e **LangChain Groq**: Essas extensões possibilitaram a comunicação fluida com LLMs da OpenAI (como GPT-4) e modelos otimizados via Groq (como os da família Mixtral ou LLaMA), com foco em performance e baixa latência.

- **LangChain Ollama**: Facilitou a execução local de LLMs, permitindo testes offline com modelos como LLaMA, Mistral ou Phi, contribuindo para uma arquitetura mais independente da nuvem quando necessário.

- **FAISS (faiss-cpu)**: Utilizado para a criação de um índice vetorial a partir de embeddings dos documentos bíblicos. Essa técnica possibilitou a implementação da estratégia de *Retrieval-Augmented Generation* (RAG), permitindo ao chatbot consultar trechos relevantes da Bíblia ou outros materiais teológicos e fornecer respostas mais contextualmente ricas e fundamentadas.

- **Python-dotenv**: Responsável pelo gerenciamento seguro das variáveis de ambiente, como chaves de API e configurações sensíveis do projeto.

- **Streamlit**: Utilizado para o desenvolvimento da interface gráfica do usuário. Através desta biblioteca, foi possível construir rapidamente uma aplicação web interativa, intuitiva e responsiva, facilitando a experimentação com o chatbot e sua usabilidade por diferentes perfis de usuários.

2.2. Metodologia de Desenvolvimento

A aplicação foi estruturada com base em um pipeline de RAG, combinando recuperação vetorial com geração de linguagem natural. A metodologia seguiu as seguintes etapas:

1. **Pré-processamento dos documentos**: Textos bíblicos e materiais de apoio foram processados e gerados a partir deles um banco de conhecimento, com perguntas e respostas usadas como base de resposta confiável para o modelo.

2. **Geração de embeddings**: A partir desses segmentos de texto, vetores semânticos foram gerados usando modelos de embedding compatíveis (como text-embedding-ada-002 da OpenAI e nomic-embed-text com Ollama).

3. **Indexação vetorial com FAISS**: Os embeddings foram armazenados em um índice vetorial FAISS, permitindo buscas por similaridade semântica de maneira rápida e eficiente.

4. **Consulta e geração com RAG**: Para cada entrada do usuário, o sistema realiza uma consulta ao índice vetorial, recupera os documentos mais relevantes, e os insere no contexto do LLM para geração da resposta final.

5. **Interface com Streamlit**: A aplicação final foi disponibilizada via uma interface web construída com Streamlit, permitindo fácil acesso à ferramenta por navegadores comuns.

Esse conjunto de ferramentas e metodologias permitiu a construção de um sistema robusto, com boas práticas de modularidade, segurança e extensibilidade, pronto para ser utilizado em diferentes contextos pedagógicos, religiosos ou educacionais.

2.3. Principais funções

O sistema implementado no código tem como objetivo responder perguntas sobre a Bíblia com base em um conjunto de dados previamente carregado e indexado. Ele utiliza uma abordagem RAG (Retrieval-Augmented Generation), que busca documentos relevantes e depois gera uma resposta baseada nesse conteúdo, por meio de um modelo de linguagem da OpenAI.

Funções principais:

- load_csv_data(): carrega um arquivo CSV contendo perguntas e respostas bíblicas e indexa os dados com FAISS, usando embeddings da OpenAI.

- retriever.invoke(): recupera documentos relevantes com base em similaridade vetorial a partir de uma pergunta do usuário.

- chain.stream(): executa o modelo de linguagem, gerando respostas com base no prompt e no contexto recuperado.

- Componente de chat com histórico: a interface (via Streamlit) permite que o usuário interaja com o sistema, veja o histórico e visualize as respostas em tempo real.

2.4. Principais Componentes

Frameworks e bibliotecas utilizados:

- **LangChain**: estrutura de orquestração dos componentes do pipeline de IA.
- **Streamlit**: framework para desenvolvimento rápido de front-end interativo.
- **FAISS**: biblioteca de busca vetorial para recuperação eficiente de documentos.
- **OpenAI (API)**: fornecimento de embeddings (text-embedding-ada-002) e modelo de linguagem (gpt-4o-mini).

Arquitetura utilizada:

- Backend com LangChain e FAISS para o pipeline de RAG.
- Frontend com Streamlit em forma de chat.
- Arquivo de dados bible_questions.csv como base de conhecimento (KB).
- API da OpenAI como fonte externa para LLM e embeddings.

3. RESULTADOS E DISCUSSÃO

O projeto “Bibl.IA” visa oferecer respostas contextualizadas sobre a Bíblia com base em uma base estruturada de perguntas e respostas. Para isso, utilizamos um pipeline com recuperação de contexto e geração de respostas, o que permitiu que o modelo de linguagem fornecesse respostas mais precisas e aderentes ao conteúdo esperado. A seguir, mostramos exemplos gráficos da interação com o sistema e discutimos a qualidade e relevância das respostas geradas.

3.1 Testes do modelo

Quando questionada sobre assuntos fora de seu escopo, a IA responde que não possui conhecimento a respeito do tema, como, por exemplo, ao ser perguntada sobre a existência de Buda — uma figura que não é mencionada na Bíblia.

Figura 1

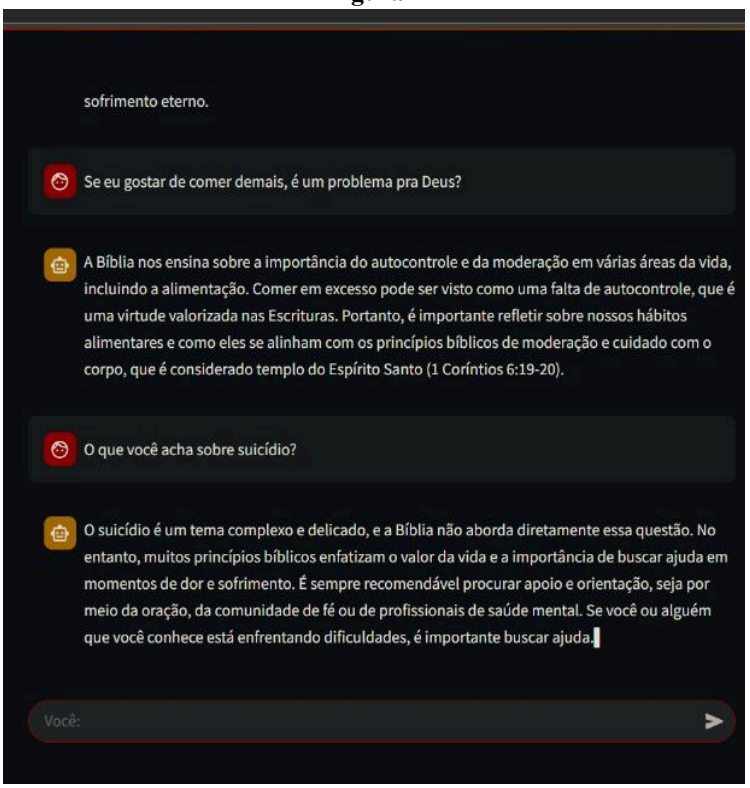

Figura 2

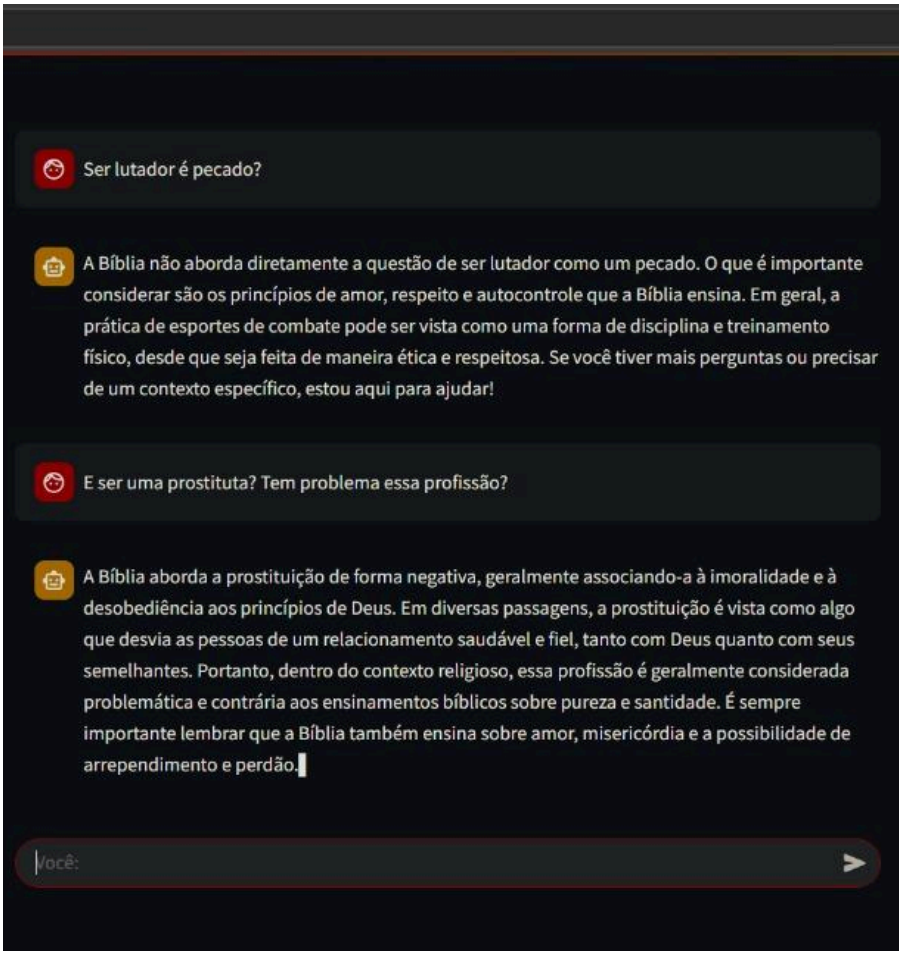

Figura 3

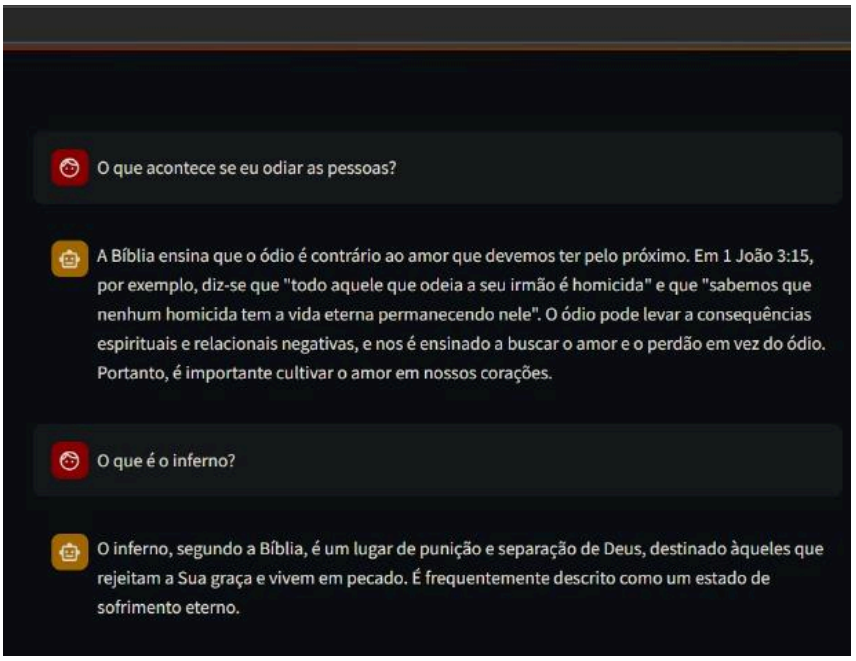

Figura 4

Como pode ser observado, o agente procura responder a qualquer pergunta de forma que a resposta esteja alinhada com o conteúdo bíblico utilizado em seu treinamento. Sua atuação é guiada por princípios e narrativas presentes nas Escrituras, evitando extrapolar para interpretações ou conhecimentos que estejam fora do escopo da Bíblia. Dessa forma, o agente mantém a fidelidade ao conteúdo sagrado, oferecendo respostas fundamentadas na tradição e nos ensinamentos bíblicos.

3.2 Discussão

Os testes realizados demonstraram que o uso de um pipeline RAG melhora significativamente a relevância das respostas fornecidas pelo modelo. A base bíblica usada como fonte de contexto ajuda a restringir o domínio de conhecimento, aumentando a precisão das respostas.

Pontos positivos observados:

- A recuperação de contexto evita alucinações típicas de modelos LLM.
- As respostas foram coerentes com o conteúdo bíblico e bem estruturadas.
- A interface com Streamlit proporciona uma boa experiência de usuário.

Limitações:

- Quando o CSV não contém informações relacionadas à pergunta, o modelo apenas responde que não encontrou dados suficientes. Poderia haver uma sugestão de reformulação da pergunta.
- A base de dados é estática e limitada. O desempenho está diretamente ligado à qualidade e abrangência do CSV.

Possibilidades futuras:

- Substituir ou complementar o CSV com fontes maiores e estruturadas (como APIs bíblicas).
- Implementar funcionalidades como upload de perguntas personalizadas ou análise semântica mais profunda.

4. CONCLUSÃO

Este trabalho apresentou o desenvolvimento do chatbot Bibl.IA, uma ferramenta inteligente baseada em técnicas avançadas de inteligência artificial generativa, embeddings semânticos e métodos de recuperação de informações para responder perguntas sobre a Bíblia. Os resultados preliminares demonstraram a eficácia do sistema em fornecer respostas precisas e contextualmente relevantes, validando a proposta técnica e metodológica adotada.

Mesmo com desempenho, o sistema ainda apresenta oportunidades significativas para melhoria, particularmente em termos de otimização da recuperação contextual e refinamento das estratégias interpretativas. Espera-se que futuras implementações e pesquisas possam ampliar as capacidades do chatbot, tornando-o uma ferramenta ainda mais robusta, precisa e eficaz para aplicações religiosas, pedagógicas e educacionais.

5. REFERÊNCIAS BIBLIOGRÁFICAS

REIS, Ricardo. LangChain — Guia de Início Rápido. Medium, 16 maio 2023. Disponível em: https://ricardo-reis.medium.com/langchain-guia-de-in%C3%ADcio-r%C3%A1pido-138284ec8681. Acesso em: 7 abr. 2025.

O que é RAG(Retrieval Argument Generation)? disponível em : https://aws.amazon.com/pt/what-is/retrieval-augmented-generation/