Projeto Residência em TIC

ChatPDF com Inteligência Artificial

**Turma 2**: ENIO – LINDY – LUIZ – PATRICIA
Orientador: Professor Jackson - UNOESC

Chapecó
2024

Resumo

Nós desenvolvemos o **ChatPDF_DotSet**, um aplicativo web projetado para atender às necessidades específicas da empresa DotSet. Inspirados pela limitação dos chatPDFs disponíveis no mercado, que geralmente exigem a compra de créditos após um número limitado de perguntas, nós decidimos criar uma solução que fosse acessível e eficiente para os clientes da DotSet. Esses clientes frequentemente enfrentam dúvidas ao responder questões importantes e contam com a orientação de consultores. Esses consultores, por sua vez, são auxiliados por arquivos em formato PDF para fornecer suporte adequado.

O ChatPDF_DotSet foi concebido para resolver esse desafio, utilizando inteligência artificial para analisar e extrair informações de documentos PDF, facilitando o trabalho dos consultores. Nossa aplicação é alimentada pela API OpenAI, que disponibiliza Grandes Modelos de Linguagem (LLMs) capazes de interpretar e responder perguntas baseadas no conteúdo dos PDFs. Essas LLMs são compostas por redes neurais sofisticadas, com codificadores e decodificadores que compreendem e extraem significado dos textos, reconhecendo as relações entre palavras e frases.

Para implementar essa solução, nós utilizamos a linguagem de programação Python, que oferece várias bibliotecas para integração com a API da OpenAI, bem como para a leitura e extração de dados de documentos PDF e a comunicação via diversos protocolos. O resultado é uma ferramenta prática que aprimora a eficiência e a precisão no suporte aos clientes da DotSet.

Palavras Chaves: Inteligência Artificial; Modelos de Linguagens Grandes; ChatPDF

## 1. Execução do ChatPDF_DotSet

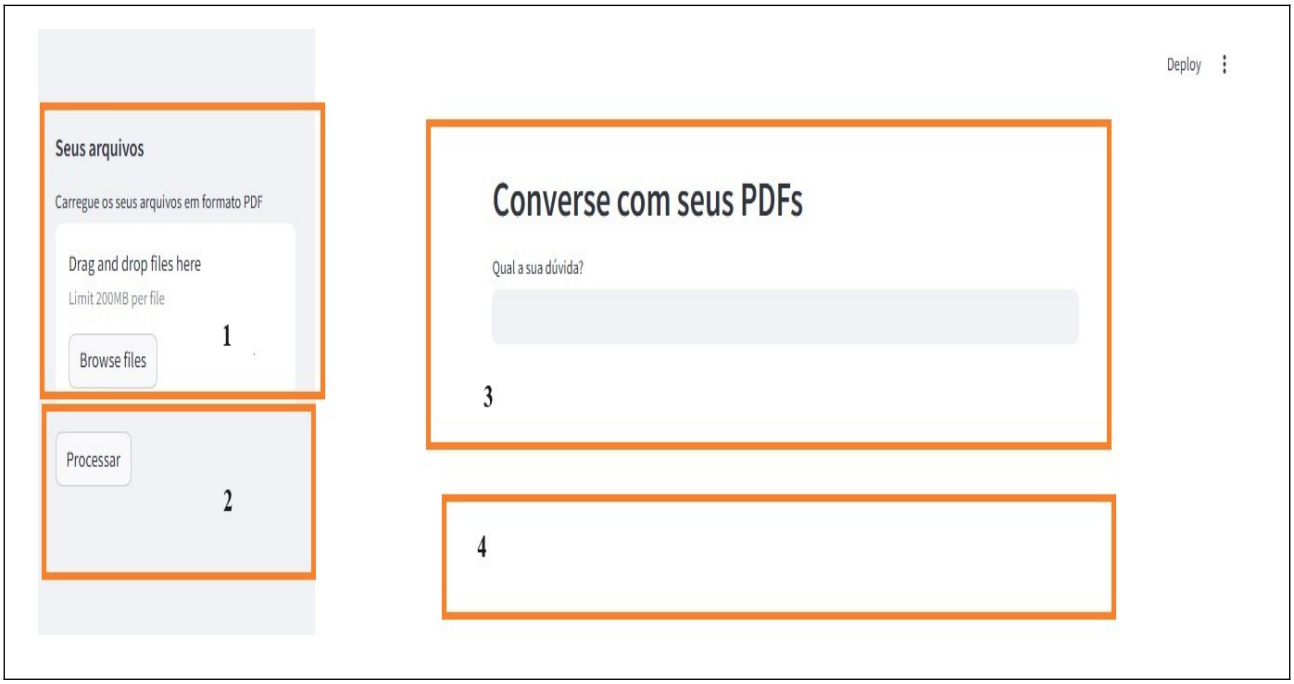


Figura 1 – Interface do Usário

A Figura 1 apresenta a interface com o usuário do ChatPDF_DotSet:

1) Botão Brose files: O usuário utiliza este botão para fazer o upload do documento PDF desejado;
2) Botão Processar: Após o download o usuário clica neste botão então todo o processo sobre o documento PDF é executado (leitura, extração, divisão em partes e transformação em códigos);
3) Campo de Texto: o usuário digita sua dúvida e aciona a tecla Enter. Então a resposta é buscada no documento PDF;
4) Apresentação de Perguntas e Respostas: São dispostas as dúvidas apresentadas e as respctivas respostas a estas dúvidas.

## 2. Detalhes da Implementação do ChatPDF_DotSet

A Figura 2 ilustra o fluxo de execução do ChatPDF_DotSet. Para a implementação, utilizamos a linguagem de programação Python, sendo os principais arquivos envolvidos text.py, chatbot.py e app.py.

Após o usuário clicar no botão "Processar", conforme descrito na Seção 1, o arquivo text.py é acionado. Esse arquivo realiza a leitura e extração do documento PDF por meio da função Process_Files(Files), que espera como argumento um arquivo no formato PDF. Em seguida, a

função Create_text_chunks(text) divide o documento PDF em partes menores, que chamamos de *chunks*, cujo tamanho pode ser ajustado conforme necessário pelos desenvolvedores ou administradores do sistema. Para essas operações, o arquivo text.py utiliza as bibliotecas PyPDF2 e langchain.text_splitter, respectivamente.

Na sequência, o arquivo chatbot.py é acionado, onde a função create_vectorstore(chunks) gera o espaço vetorial dos referidos *chunks*. A função create_conversation_chain(vectorstore) recebe como argumento esse espaço vetorial e retorna uma cadeia de conversação, denominada *conversation_chain*, que é armazenada temporariamente em memória volátil durante a execução da aplicação.

As respostas às perguntas do usuário são fornecidas por meio de cálculos de similaridade realizados pelo pacote FAISS (Facebook AI Similarity Search), implementado dentro da função create_vectorstore(chunks) no arquivo chatbot.py. Para a execução do chatbot.py, são importadas classes do pacote LangChain do Python.

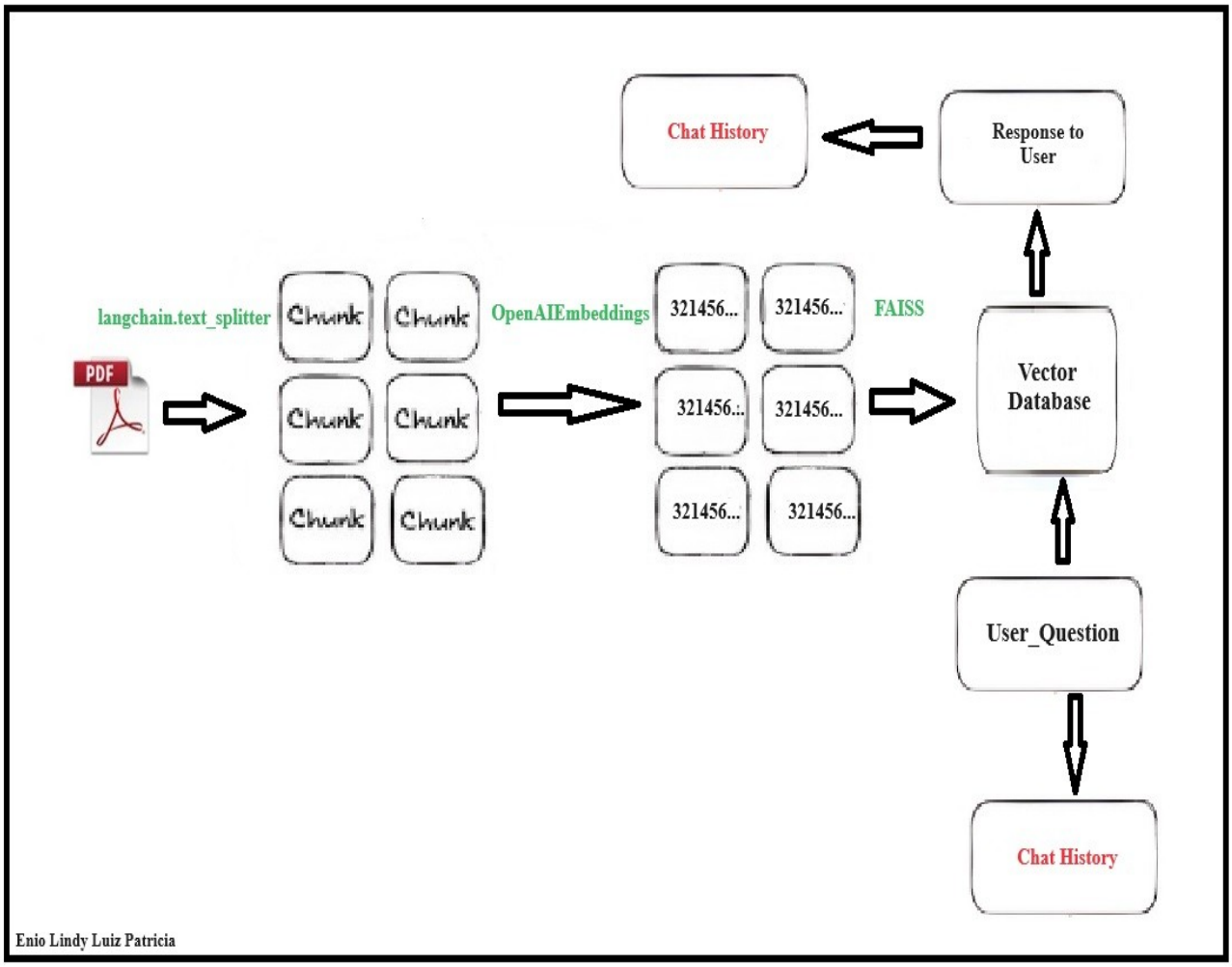


Figura 2 - fluxo de execução do ChatPDF_DotSet

O terceiro arquivo, app.py, é responsável por gerenciar a interface com o usuário, realizando as chamadas das funções presentes em text.py e chatbot.py, passando como argumento a estas funções as entradas do usuário (documento PDF e perguntas) e, em seguida, apresentando as respostas ao usuário.

A chave secreta, que a OpenAI, identifica seus usuários, é informada no arquivo .env.

**3. Referências**


Build your own Notion chatbot - Show the Community! - Streamlit. Último Acesso: 14/08/2024

Overview - OpenAI API. Último Acesso: 14/08/2024.

Streamlit documentation. Último Acesso: 14/08/2024.

3.12.5 Documentation (python.org). Último Acesso: 14/08/2024.

Quickstart tutorial - OpenAI API , Último Acesso: 14/08/2024.