

Relatório - Construção de um chatbot baseado em IA para responder dúvidas sobre o vestibular da Unicamp 2024.

Jonathan do Ouro

Large language models, como o Gpt-3, possuem conhecimento obtidos em seu treinamento e ficam armazenados em seus parâmetros, assim eles possuem um conhecimento congelado no tempo, tornando difícil realizar tarefas de domínio específico sem um fine-tuning. Dessa forma, para criar um chatbot que responda perguntas sobre o vestibular da Unicamp é utilizado a técnica de Retrieval Augmented Generation (RAG), que integra informações de domínio com um large language model, assim como demonstrado na figura 1.

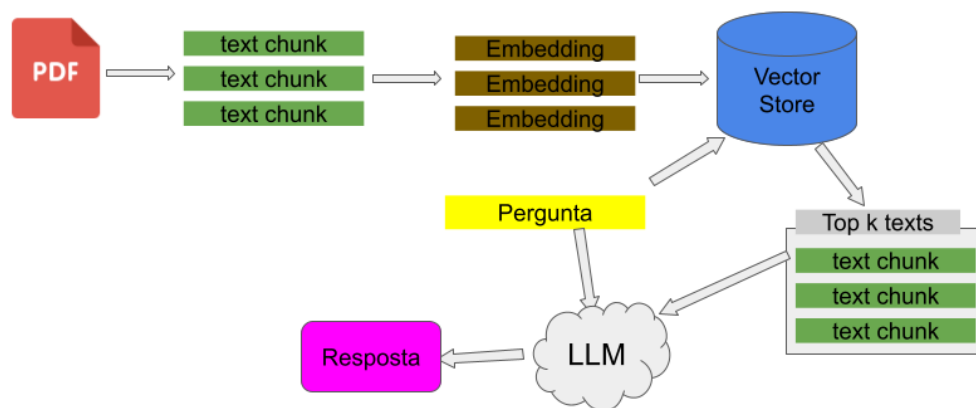


Figura 1: Arquitetura de RAG utilizada para implementar o chatbot

Cada componente do sistema afeta a qualidade da resposta final, portanto é essencial realizar experimentos e avaliar com métricas a influência de cada um dos componentes nos resultados finais a fim de obter uma configuração ideal para o chatbot.

Dessa forma, foi realizado experimentos para encontrar uma boa configuração do sistema de retrieval. Nesses experimentos foi fixado o banco de vetores Chroma Db e averiguado a influência da função que quebra o texto grande em pequenos pedaços e os modelos geradores de embeddings, os componentes experimentados são:

1. Funções para dividir o texto completo da resolução do vestibular Unicamp em pedaços pequenos
 - a. [NLTKTextSplitter](#)
 - b. [RecursiveCharacterTextSplitter](#)
 - c. [llmsherpa](#)
2. Modelo gerador de embeddings
 - a. [text-embedding-ada-002](#)
 - b. [multilingual-e5-large](#)

Para determinar qual das configurações era a melhor, foi desenvolvido um dataset que contém em cada tupla uma pergunta, trechos de textos relevantes para a pergunta e sua respectiva resposta. Assim, para cada pergunta foi feita uma busca semântica que retornava os textos mais similares, com esses textos da busca e os trechos relevantes associados a pergunta era calculado mean average precision, normalized discounted cumulative gain, recall at 3, precision at 3 e f1 score. Os resultados estão na figura 2.

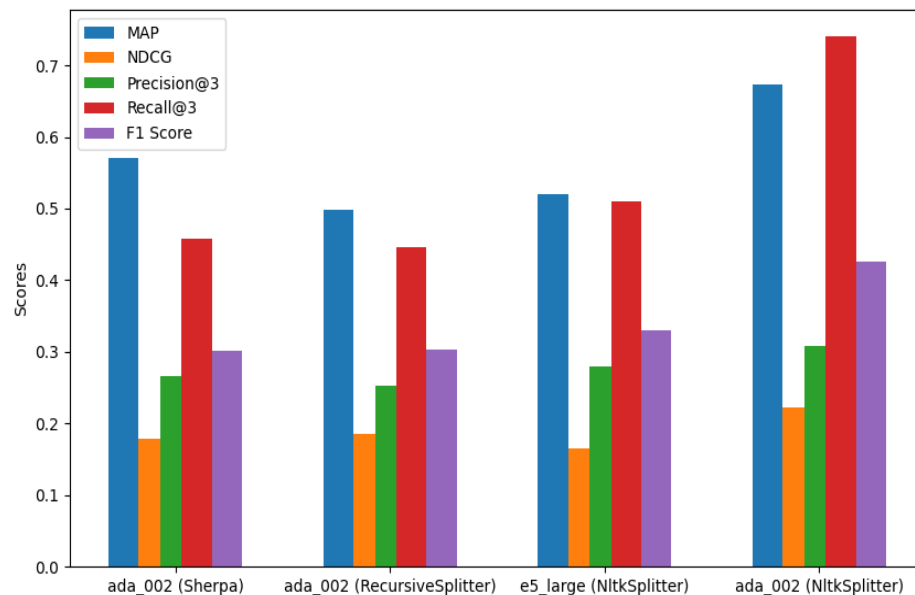


Figura 2: Comparação de Métricas entre Diferentes Embeddings e Text Splitters no sistema de text retrieval

O uso do chat-gpt para esse processo foi fundamental, pois ajudou muito a implementar as funções que calculam as métricas e interpretar os resultados, além é claro de ter em primeiro lugar recomendado tais métricas. Assim, ficou definido utilizar o text splitter com NLTK e o modelo de embedding ada 002 da Open AI.

Uma vez definido o sistema de retrieval, foi avaliado a qualidade das respostas geradas pelo gpt-3.5-turbo, onde a avaliação foi feita utilizando o mesmo dataset mas dessa vez considerando os pares de perguntas e respostas.

O chat-gpt sugeriu algumas métricas como precision, recall e f1 score para computar a qualidade da resposta gerada em relação a esperada, e pesquisando encontrei um paper que introduz uma métrica chamada SAS (Semantic Answer Similarity) baseada em um cross encoder que mede a similaridade entre duas respostas.

Utilizando Langchain para fazer prompts e utilizar o gpt-3.5-turbo foi feita a avaliação do chatbot e os resultados estão na figura 3.

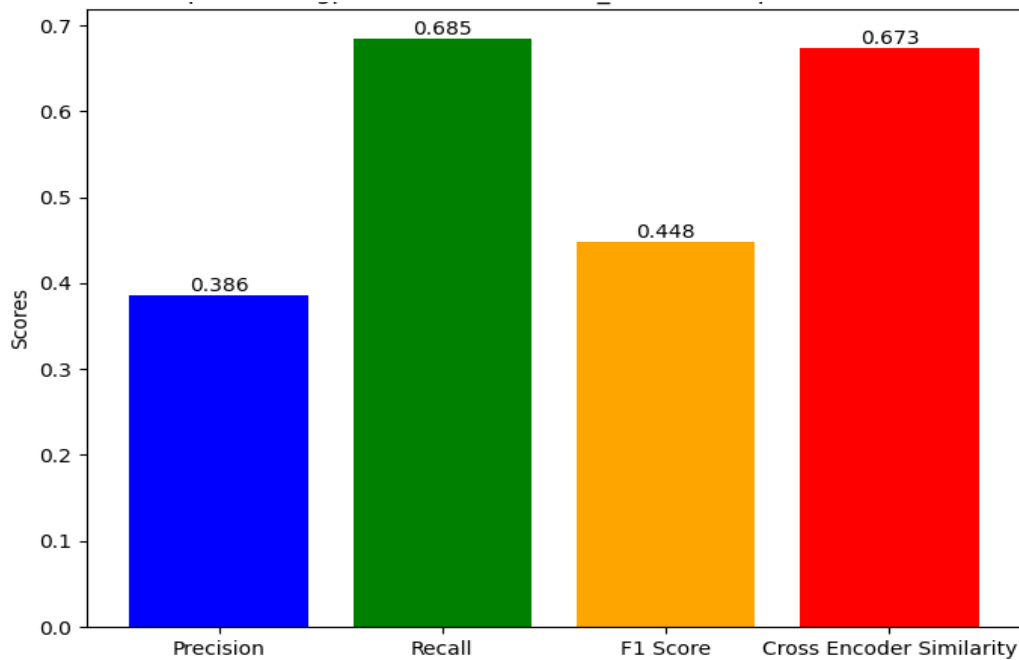


Figura 3: Avaliação do chatbot

Dessa forma o chatbot final é construído com o Text Splitter da biblioteca NLTK, gerador de embeddings ada 002 da Open AI, banco de vetores Chroma DB e gpt-3.5-turbo como large language model.

Os experimentos estão disponíveis em um notebook do google colab e a implementação final é esta na principalmente na pasta Domain e no arquivo main.py

Para utilizar o chatbot acesse esse link:

Ou se preferir rodar localmente, pode clonar o repositório, instalar todas dependências do arquivo requirements.txt e rodar o comando `'streamlit run main.py'` na raiz do projeto.

1. Como o chatbot é estruturado, necessidade de RAG, diagrama do RAG
 - a. Dizer que cada parte influencia a resposta final
 - b. Explicar a necessidade de encontrar a melhor combinação de componentes através de experimentos
2. Descrever como o chatbot foi implementado
 - a. Primeira parte do sistema de retrieval
 - i. Métricas utilizadas para avaliar
 - ii. Quais experimentos
 1. Splitters
 2. Embeddings
 - iii. Dataset utilizado: forma como ele é utilizado para avaliar
 - iv. Resultados com gráficos
 - b. Sistema de respostas
 - i. Dataset para avaliar
 - ii. Métricas utilizadas: paper cross encoder
 - iii. Experimentos
 - iv. Resultados com gráficos
3. Conjunto final de componentes
 - a. Listar componentes utilizados
 - b. Gráfico com métricas finais
 - c. Como testar o chatbot
 - i. On-line
 - ii. LocalHost