Desenvolvimento de um Sistema Inteligente de Recuperação de Informações sobre o Bacharelado em Inteligência Artificial.

Participantes: André Cerqueira Castro, Dayane Rodrigues, Hugo Pessoni e Pedro Rabelo Mendonça.

Introdução:

(a) Justificativa

Nosso projeto visa facilitar o acesso a informações sobre o curso de Bacharelado em Inteligência Artificial (BIA) da UFG, fornecendo assim uma aplicação que responde a dúvidas de estudantes, vestibulandos e interessados de outras áreas. Esse sistema permitirá que informações como duração do curso, grade curricular, carga horária e outras questões comuns sejam acessadas de forma fácil e automatizada.

(b) Objetivos

O objetivo principal é desenvolver uma aplicação RAG (Recuperação Aumentada com Geração) que ofereça recuperação eficiente de informações sobre o BIA. Compararemos o Graph RAG e o RAG tradicional para identificar o método de melhor desempenho no contexto educacional. O RAG tradicional usa uma base vetorial para recuperação, o que favorece consultas amplas e contextuais; já o Graph RAG utiliza um banco de dados gráfico, permitindo relacionar entidades e explorar conexões estruturadas entre temas. No nicho educacional, o Graph RAG pode oferecer respostas mais contextuais e interligadas, mas demanda maior complexidade na configuração e manutenção. O RAG tradicional, por outro lado, é mais simples e ágil, mas pode perder precisão em respostas que dependem de relações complexas entre os dados. Nossa meta é criar uma aplicação robusta, com precisão e relevância nas respostas fornecidas.

Proposta de Pesquisa:

(a) Método Proposto

- **Definição do Problema:** A necessidade de fornecer informações sobre o curso de BIA de maneira acessível e responsiva. A aplicação deverá responder perguntas relacionadas a diversos aspectos do curso usando técnicas avançadas de recuperação e geração de conteúdo.
- **Dados Disponíveis:** Os dados a serem utilizados incluem:
  - Projeto Pedagógico do Curso (PPC) do BIA.
  - Transcrição da entrevista dada pelo coordenador do curso, Anderson Soares, sobre o BIA no quadro Guia das Profissões do Brasil Escola.
  - Crawler de fontes web, incluindo o site oficial do BIA e notícias sobre a criação do curso. Exemplos: Notícia de criação do curso Site oficial do BIA Notícia UFG sobre a criação do curso.

- **Metodologia/Técnicas/Ferramentas Utilizadas:**
  - **Frameworks para o RAG:** Usaremos o LlamaIndex como *framework* para construir *pipelines* RAG, já que ele facilita o desenvolvimento e integração das técnicas de recuperação e geração.
  - **Banco de Dados Vetorial:** Utilizaremos o Qdrant como *vectorstore* para o RAG tradicional, uma vez que possibilita uso gratuito, atendendo ao perfil do projeto e permitindo busca híbrida. Em segundo plano, consideraremos o Chroma, que além de facilidade de uso local, também oferece opção gratuita.
  - **Graph Database:** Para o Graph RAG, será utilizado o Neo4j, que permite uso gratuito para nosso perfil de projeto e é amplamente reconhecido na área. Em caso de necessidade, o Nebula será considerado como opção secundária, embora exija um processo de solicitação para acesso gratuito, o que o torna um pouco mais burocrático.
  - **Recuperação Híbrida:** Utilizaremos uma combinação de busca esparsa e densa (Hybrid Search) para aproveitar os pontos fortes de cada técnica: a busca esparsa (ex., BM25, TF-IDF) é eficaz em identificar correspondências exatas em textos, enquanto a busca densa (ex., DPR, ColBERT) utiliza *embeddings* para captar relações semânticas e contextuais, resultando em respostas mais abrangentes e relevantes. Com essa abordagem, podemos equilibrar precisão e contexto, melhorando a qualidade das respostas. O Qdrant se destaca como *vectorstore* ideal para essa aplicação, oferecendo suporte gratuito para *hybrid search*, o que facilita nossa implementação e permite um uso otimizado dos recursos.
  - **Reranking:** A reordenação dos resultados será realizada, quando possível, por um modelo de *reranking* (ex., Cohere), para assegurar maior precisão nas respostas recuperadas. O *reranking* melhora a qualidade dos resultados ao ordenar as respostas mais relevantes para o topo. Verificaremos a possibilidade de uso gratuito do Cohere ou de outras opções de *reranking*, dado que ainda precisamos confirmar se há disponibilidade de teste gratuito.
  - **LLM (Modelo de Linguagem de Grande Escala):** O uso de um modelo da OpenAI será explorado caso tenhamos acesso a uma chave da API. Caso as limitações financeiras sejam uma barreira, rodaremos o LLM localmente utilizando o Ollama, que suporta o modelo phi3 3.8b, capaz de rodar em configurações modestas com resultados satisfatórios em português. Se tivermos mais recursos computacionais, o modelo Llama3 8b será considerado.
- **Perfil do Agente:** O sistema buscará preparar um perfil de tutor de curso, que responda de forma precisa e clara a perguntas sobre o BIA. Inicialmente, aplicaremos técnicas de *prompt engineering* no template RAG para estabelecer esse perfil do agente. Caso essa abordagem não produza resultados satisfatórios, consideraremos realizar um fine-tuning do modelo, conforme sugerido por Diogo, embora seja necessário verificar os custos e limitações dessa opção antes de sua implementação.
- **Ferramentas para Dados:**
  - **Crawler de Dados:** Ferramentas como UseScraper ou ScrapingBee serão utilizadas para capturar informações de sites e fontes externas. Ambas ferramentas possibilitam o uso gratuito em suas versões de teste, o que é suficiente para nosso escopo inicial.

- ➢ **Transcrição de Vídeos:** Utilizaremos as ferramentas TubRipper para salvar o áudio do vídeo da entrevista do Anderson Soares sobre o BIA e o Sonix para transcrevê-lo, assim possibilitando seu uso no dataset.

(b) Plano de Trabalho

1. **Etapa 1 - Coleta de Dados e Pré-Processamento:** Recolheremos o PPC, transcrições de entrevistas e conteúdo da web. Os dados serão limpos, organizados e estruturados para uso.
2. **Etapa 2 - Implementação dos Modelos de Recuperação e Geração:** Configuraremos o Qdrant e o Neo4j, assim como as técnicas de busca híbrida. Modelos de *reranking* serão testados para validar a relevância dos resultados.
3. **Etapa 3 - Avaliação e Comparação de Desempenho:** O desempenho do RAG tradicional e Graph RAG será comparado usando métricas de similaridade (cosseno, distância de Manhattan) e de *reranking*. A avaliação será apoiada por LLMs externos que verificarão a qualidade das respostas geradas.
4. **Etapa 4 - Ajustes Finais e Prompt Engineering:** Otimizaremos o agente usando *prompt engineering* para aprimorar a precisão e relevância das respostas. Se necessário, consideraremos o fine-tuning com base nas métricas observadas.

Métricas

Para avaliação, utilizaremos:

- ❖ **Scores de Similaridade, Distância:** Como não dispomos de dados anotados, avaliaremos o desempenho do Graph RAG e do RAG tradicional comparando *scores* de recuperação, incluindo similaridade de cosseno e distância de Manhattan, para medir a proximidade entre consultas e respostas.
- ❖ **Avaliação de Geração com LLM:** Para avaliar o *pipeline* RAG completo, utilizaremos LLMs de alto desempenho, como o modelo 4o, para comparar as respostas geradas. Esse processo envolverá a análise da pergunta, do contexto recuperado e da geração final, seguindo um protocolo estabelecido de avaliação para medir precisão e relevância nas respostas.

Conclusão

O desenvolvimento de uma aplicação RAG para responder perguntas sobre o Bacharelado em Inteligência Artificial representa uma solução eficaz para facilitar o acesso a informações educacionais. Através da comparação entre Graph RAG e RAG tradicional, visamos identificar a abordagem mais precisa e relevante para esse domínio. Com o uso de *frameworks* e ferramentas acessíveis e estratégias de avaliação rigorosas, esperamos oferecer uma plataforma robusta, acessível e ajustada às necessidades dos usuários, promovendo a disseminação de conhecimento sobre o curso.