ASSISTENTE DE BUSCA AUMENTADA POR GERAÇÃO PARA O ACERVO DA SIIEPE/UFPEL

NICOLAS SOSA MACHADO[1]; ANDERSON PRIEBE FERRUGEM[2], TATIANA AIRES TAVARES[3]:

[1]Universidade Federal de Pelotas – nsmachado@inf,ufpel.edu.br
[2]Universidade Federal de Pelotas – tatiana@inf.ufpel.edu.br
[2]Universidade Federal de Pelotas – ferrugem@inf.ufpel.edu.br

1. INTRODUÇÃO

A Semana Integrada de Ensino, Pesquisa e Extensão (SIIEPE) da Universidade Federal de Pelotas (UFPel) representa o principal evento acadêmico da instituição, consolidando anualmente um vasto repositório de conhecimento. Milhares de trabalhos são apresentados, abrangendo diversas áreas, e seus respectivos artigos são publicados nos anais do evento. Contudo, essa riqueza de informações permanece subutilizada devido à dificuldade de realizar buscas semânticas e transversais que conectem ideias e resultados dispersos ao longo de diferentes edições e áreas do conhecimento. Ferramentas de busca tradicionais, frequentemente falham em capturar o contexto e a nuance das questões de pesquisa, limitando o potencial de descoberta e de síntese do conhecimento acumulado.

Neste cenário, a Inteligência Artificial (IA), em particular os Grandes Modelos de Linguagem (LLMs), oferece uma oportunidade a ser explorada. A arquitetura de Busca Aumentada por Geração (RAG) surge como uma solução robusta para mitigar as limitações dos LLMs, como a geração de informações incorretas e a falta de acesso a dados específicos e atualizados (LEWIS et al., 2020). A propensão dos modelos de linguagem a gerar conteúdo plausível, porém factualmente incorreto, é um desafio amplamente documentado na área (JI et al., 2023). O RAG aprimora um LLM conectando-o a uma base de conhecimento externa, permitindo que ele fundamente suas respostas em fatos concretos extraídos de documentos relevantes.

Este trabalho justifica-se pela necessidade de um sistema de apoio acadêmico que democratize e otimize o acesso à produção científica da UFPel. Propõe-se o desenvolvimento de um assistente de chat inteligente, fundamentado na arquitetura RAG, capaz de responder a perguntas em linguagem natural sobre todo o acervo de artigos dos anais da SIIEPE. O objetivo é transformar um arquivo estático de documentos em uma base de conhecimento dinâmica e conversacional, facilitando a pesquisa para alunos, professores e pesquisadores, e promovendo novas conexões entre os trabalhos desenvolvidos na universidade.

2. METODOLOGIA

A arquitetura do sistema foi projetada de forma modular para garantir eficiência e escalabilidade, seguindo um fluxo de processamento de dados bem definido,

desde a coleta dos artigos até a geração da resposta ao usuário. O processo pode ser dividido nas seguintes etapas.

1. **Coleta Automatizada de Dados**: Um *worker* desenvolvido em Python é responsável por navegar programaticamente no portal de anais da SIIEPE. Ele itera sobre os anos, áreas do conhecimento e eventos, identificando e baixando os artigos científicos em formato PDF. Para cada artigo, o *worker* extrai metadados essenciais da página web, como título, autores, orientador e ano.
2. **Pré-processamento e Enriquecimento**: Antes da ingestão, cada PDF passa por uma etapa de enriquecimento. Uma nova primeira página contendo os metadados extraídos é gerada e inserida no documento original. Esta etapa é crucial, pois garante que informações contextuais importantes sejam indexadas junto ao conteúdo do artigo, viabilizando buscas filtradas e mais precisas.
3. **Segmentação e Vetorização (Embeddings)**: O texto completo de cada PDF enriquecido é extraído e segmentado em blocos de texto menores e sobrepostos (*chunks*). Esta segmentação é necessária para adequar o conteúdo aos limites de contexto dos modelos de linguagem. Cada *chunk* é então processado por um modelo de *embedding* que o converte em um vetor numérico de alta dimensão seguindo uma abordagem de pré-treinamento contrastivo (WANG et al., 2022). Esse vetor captura o significado semântico do texto, permitindo que trechos com significados similares sejam matematicamente próximos no espaço vetorial.
4. **Armazenamento em Banco de Dados Vetorial**: Os vetores gerados, juntamente com o texto original do *chunk* e os metadados do documento, são armazenados no banco de dados vetorial Qdrant. O Qdrant é otimizado para realizar buscas de similaridade em alta velocidade, permitindo encontrar os *chunks* de texto mais relevantes para uma determinada pergunta em um grande volume de dados.
5. **Recuperação e Geração de Resposta (Pipeline RAG)**: Esta é a etapa interativa com o usuário.
    - **Consulta do Usuário**: O usuário submete uma pergunta em linguagem natural através de uma interface web.
    - **Análise da Consulta**: A pergunta é processada por um LLM local (Qwen) para extrair possíveis filtros (autor, ano, área) e refinar a consulta principal para a busca semântica.
    - **Busca (Retrieval)**: A consulta refinada é convertida em um vetor de *embedding* usando o mesmo modelo da etapa 3. O sistema utiliza este vetor para consultar o Qdrant, que retorna os *k chunks* de texto mais semanticamente similares, respeitando os filtros extraídos. Esta abordagem, conhecida como *dense retrieval*, é fundamental para a eficácia de sistemas de pergunta e resposta em domínio aberto (KARPUKHIN et al., 2020).
    - **Aumento (Augmentation)**: Os *chunks* de texto recuperados são concatenados e formatados como um "contexto".
    - **Geração (Generation)**: O contexto e a pergunta original do usuário são enviados ao LLM com uma instrução específica: "Responda à pergunta baseando-se *exclusivamente* nas informações contidas no contexto fornecido". O LLM, então, sintetiza as informações dos trechos recuperados e gera uma resposta coesa e contextualizada.

3. RESULTADOS E DISCUSSÃO

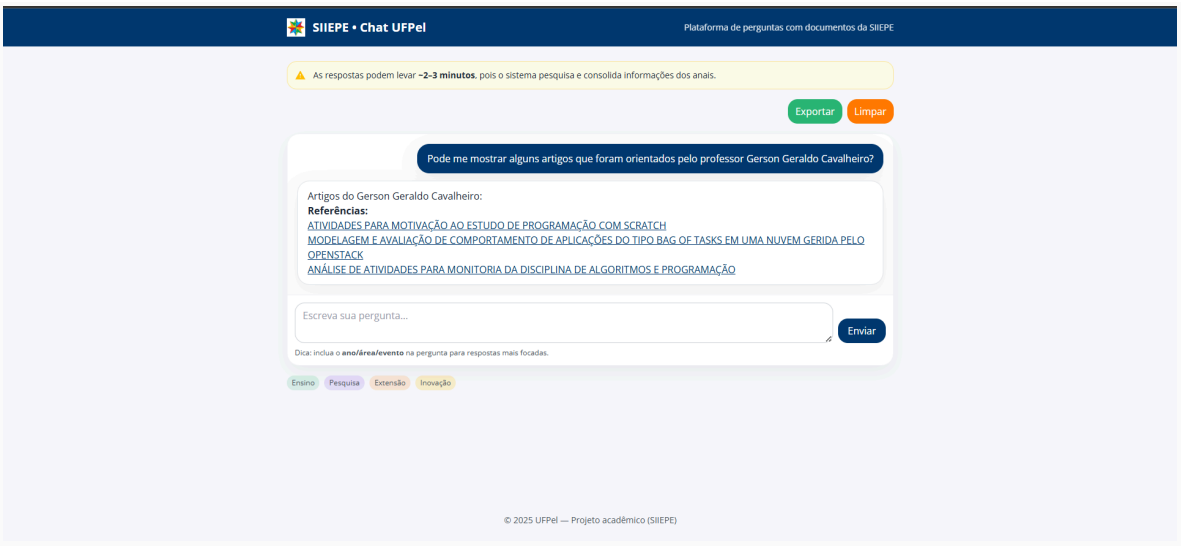
O desenvolvimento do projeto atingiu um estágio funcional, com o *pipeline* de coleta, processamento, armazenamento e consulta operando de forma integrada. O *worker* de coleta foi capaz de processar os anais de múltiplos anos, populando o banco de dados vetorial com os documentos segmentados e vetorizados.

Para viabilizar a interação do usuário com este *pipeline*, foi desenvolvida uma interface de chat web, conforme ilustrado na Figura 1. Através dela, o usuário pode submeter suas perguntas em linguagem natural de forma intuitiva. O sistema exibe as respostas geradas pelo LLM em um formato de conversação, apresentando também as fontes dos documentos originais que embasaram a resposta, o que permite a verificação e o aprofundamento na pesquisa.

A principal aplicação do sistema é a sua capacidade de responder a perguntas complexas que exigem a síntese de informações de múltiplos documentos. Esta funcionalidade representa um avanço significativo em relação à busca por palavras-chave, que retornaria apenas uma lista de documentos, exigindo que o usuário os lesse individualmente para extrair as informações. O sistema, portanto, atua como um assistente de pesquisa, realizando a primeira camada de análise e síntese de conteúdo.

Apesar dos resultados promissores, o sistema possui limitações inerentes à tecnologia. A qualidade da resposta final é diretamente dependente da qualidade dos trechos recuperados na busca vetorial. Se documentos irrelevantes forem recuperados, o LLM pode gerar uma resposta incorreta ou incompleta. Adicionalmente, a execução de LLMs locais, embora garanta privacidade, exige recursos computacionais significativos e pode resultar em um tempo de resposta maior em comparação com modelos comerciais em nuvem. A otimização do equilíbrio entre velocidade, custo e qualidade da resposta é um ponto central para a evolução do projeto.

Figura 1 – Captura de tela do sistema desenvolvido



4. CONCLUSÕES

Este trabalho demonstrou com sucesso a viabilidade e o potencial da aplicação de uma arquitetura de Busca Aumentada por Geração (RAG) para criar um assistente de chat inteligente sobre o acervo dos anais da SIIEPE/UFPel. O sistema transforma um vasto repositório de documentos PDF em uma base de conhecimento interativa e acessível, oferecendo à comunidade acadêmica uma ferramenta poderosa para a exploração e síntese da produção científica da instituição. Ao facilitar o acesso à informação e promover a descoberta de conexões entre trabalhos, o projeto contribui diretamente para a valorização do conhecimento gerado na universidade.

Como perspectivas futuras, o projeto visa a otimização contínua do *pipeline*. Isso inclui a experimentação com diferentes modelos de *embedding* e LLMs para aprimorar a acurácia da recuperação e a qualidade das respostas geradas. Pretende-se também desenvolver uma interface de usuário mais rica, com funcionalidades como histórico de conversas e a capacidade de visualizar os trechos dos documentos originais que fundamentaram a resposta. A longo prazo, a arquitetura desenvolvida poderá ser expandida para incluir outras fontes de dados acadêmicos da UFPel, como teses, dissertações e publicações de periódicos, consolidando um ecossistema de informação inteligente para toda a universidade.

5. REFERÊNCIAS BIBLIOGRÁFICAS

LEWIS, P. *et al.* Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: **Advances in Neural Information Processing Systems** 33 (NeurIPS), 2020, p. 9459–9474

JI, Z. et al. Survey of Hallucination in Natural Language Generation. **ACM Computing Surveys**, v. 55, n. 12, p. 1–38, 2023.

KARPUKHIN, V. et al. Dense Passage Retrieval for Open-Domain Question Answering. *In*: CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING (EMNLP), 2020, Online. **Proceedings...** [S. l.]: Association for Computational Linguistics, 2020. p. 6769–6781.

WANG, L. et al. Text Embeddings by Weakly-Supervised Contrastive Pre-training. **arXiv preprint arXiv:2212.03533**, 20