UNIVERSIDADE FEDERAL DO AGRESTE DE PERNAMBUCO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Analice da Silva Nascimento

José Matheus Nogueira Luciano

Luís Filipe de Barros Ferreira

RELATÓRIO DE PROJETO
RAG PARA CONSULTA DE ATIVIDADE LEGISLATIVA

Garanhuns-PE

2024

Analice da Silva Nascimento

José Matheus Nogueira Luciano

Luís Filipe de Barros Ferreira

RELATÓRIO DE PROJETO
RAG PARA CONSULTA DE ATIVIDADE LEGISLATIVA

Relatório do projeto de desenvolvimento de RAG para consulta de atividade legislativa apresentado ao curso de Bacharelado em Ciência da Computação como requisito parcial para obtenção da aprovação na disciplina de Tópicos Especiais em Inteligência Artificial.

Orientador: Luís Felipe Alves Pereira

Garanhuns– PE

2024

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de Recuperação Aumentada com Geração (RAG) para consulta de atividades legislativas, combinando técnicas de processamento de linguagem natural e aprendizado de máquina. O objetivo é melhorar a precisão e a relevância das informações recuperadas, facilitando a análise e a interpretação de documentos legislativos, com foco na consulta aos últimos projetos de lei aprovados pelo Congresso. Para isso, foram utilizadas metodologias avançadas de pré-processamento de dados, incluindo chunking e embeddings, bem como modelos de Large Language Models (LLMs) para garantir respostas mais coerentes e contextualizadas.

Palavras-Chave: Recuperação Aumentada com Geração, Processamento de Linguagem Natural, Large Language Models, Embeddings, Chunkings.

Sumário

Introdução

A complexidade e o volume crescente de dados legislativos tornam a busca por informações relevantes e atualizadas um desafio significativo. Sistemas tradicionais de recuperação de informação muitas vezes são insuficientes para fornecer resultados contextualizados e precisos, dificultando a compreensão das normas e suas implicações. Nesse contexto, o uso de técnicas avançadas de Inteligência Artificial, especialmente a Recuperação Aumentada com Geração (RAG), tem se mostrado uma abordagem promissora para melhorar a consulta e análise de documentos legislativos.

O presente trabalho propõe o desenvolvimento de um sistema baseado em RAG para consulta de atividades legislativas, com foco na recuperação dos últimos projetos de lei aprovados pelo Congresso, combinando diferentes metodologias de pré-processamento de dados, estratégias de chunkings, modelos de embeddings e modelos de Large Language Models (LLMs). A principal contribuição desta pesquisa é a implementação de um pipeline eficiente, capaz de segmentar documentos, gerar representações vetoriais precisas e empregar modelos de IA para aprimorar a recuperação e a geração de informações.

Ao longo deste trabalho, são abordadas as etapas de desenvolvimento do sistema, incluindo a seleção e análise de modelos de embeddings e LLMs, a avaliação do desempenho dos modelos e a análise dos resultados obtidos. O objetivo final é fornecer um mecanismo eficaz de consulta legislativa, reduzindo a carga cognitiva do usuário e otimizando o acesso às informações relevantes.

1. Metodologia e Arquitetura do Sistema

O fluxo de trabalho do sistema é ilustrado no Diagrama 1.1, abrangendo desde a extração de dados até a geração da resposta pela LLM.

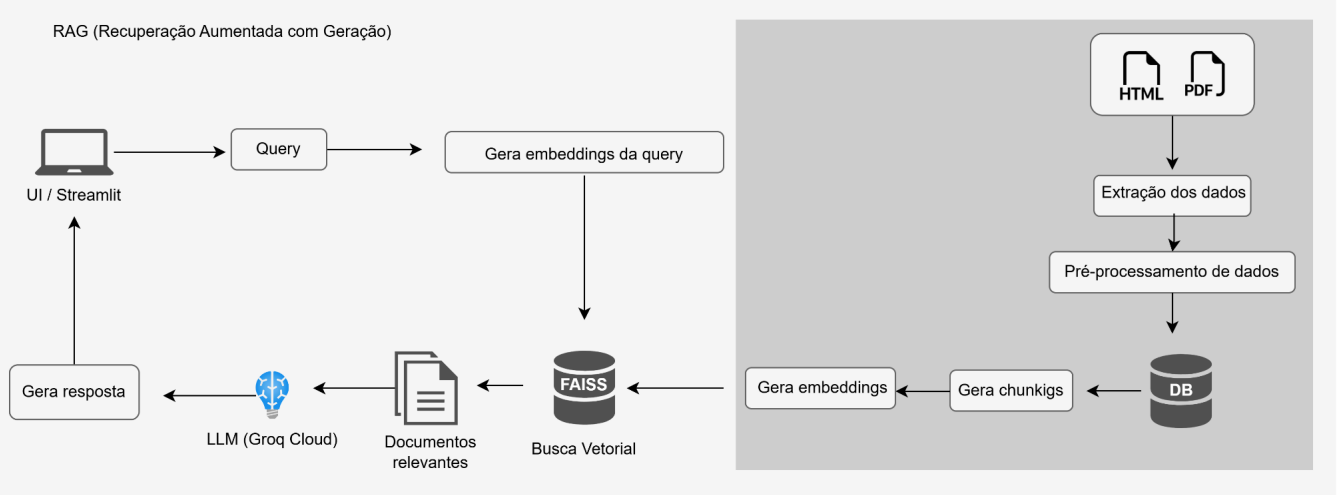

Diagrama 1.1 - Arquitetura/Fluxo de trabalho

Repositório do código-fonte do RAG para consulta legislativa: link

Foram utilizados três tipos de documentos: (i) arquivos HTML contendo leis; (ii) suas respectivas atividades legislativas (PL, PLN, PLC, PLP, PLS, MPV), no formato PDF; e (iii) seus possíveis vetos, no formato PDF.

todos disponíveis em: https://www.congressonacional.leg.br/materias/ultimas-leis-publicadas

Além disso, diversas configurações que serão detalhadas nas próximas seções, foram testadas para otimizar o processo de recuperação de informações, incluindo diferentes estratégias de pré-processamento, abordagens de segmentação textual (chunking) e variação de modelos de embeddings disponíveis no Hugging Face, bem como modelos de LLMs fornecidos pelo Groq.

A organização dos embeddings também foi analisada sob duas estratégias principais de separação de dados:

1. **Estratégia de armazenamento unificado**: nesta abordagem, os embeddings de leis, atividades legislativas e vetos foram gerados e armazenados em um único diretório. O objetivo era criar uma representação mais ampla e integrada dos documentos, permitindo a correlação entre diferentes tipos de informações. No entanto, esse método apresentou limitações significativas, resultando em uma recuperação desproporcional de documentos legislativos, favorecendo excessivamente as atividades legislativas em detrimento das leis e vetos. Assim, mesmo consultas

relacionadas a leis ou vetos frequentemente retornavam apenas documentos de atividades legislativas.

2. **Estratégia de segmentação por tipo de documento**: nesta abordagem, os embeddings foram armazenados separadamente em diretórios distintos para leis, atividades legislativas e vetos. Para evitar vieses na recuperação de documentos, foram testadas duas estratégias para balanceamento da relevância dos resultados:
    - **Estratégia baseada em similaridade direta**: a relevância dos documentos foi determinada com base na similaridade entre a consulta do usuário e os embeddings de cada diretório, priorizando aqueles com maior correspondência sem favorecer exclusivamente um único tipo de documento.
    - **Estratégia baseada em palavras-chave**: neste caso, a recuperação foi ponderada utilizando palavras-chave associadas a cada categoria documental, como "lei", "veto", "tramitação" e "atividade legislativa", permitindo um ajuste dinâmico da distribuição dos documentos recuperados.

Os resultados indicam que a estratégia de segmentação por tipo de documento, combinada com a abordagem de balanceamento baseada em similaridade direta, apresentou o melhor desempenho na recuperação de informações. Essa configuração manteve a capacidade de correlação entre diferentes tipos de documentos, ao mesmo tempo que reduziu os vieses observados na estratégia de armazenamento unificado.

Estratégias de Organização e Recuperação de Dados:

No processo de fragmentação do texto, é possível que um embedding seja gerado sem conter explicitamente o nome da lei, do projeto de lei (PL) ou do veto ao qual está relacionado. Esse problema pode comprometer a contextualização do embedding, reduzindo sua precisão na recuperação de informações com base em consultas específicas. Além da inclusão do número das leis, PLs e vetos, é fundamental indicar a que tipo de documento cada embedding pertence, pois um usuário pode gerar embeddings de diferentes tópicos para a mesma atividade legislativa.

Para mitigar esse problema, foram adicionados metadados no início de cada embedding, garantindo que as informações essenciais sobre o documento estivessem sempre presentes. A eficácia dessa estratégia foi avaliada por meio das métricas apresentadas abaixo:

estratégia	jaccard
Metadados fora dos chunking e armazenamento unificado	0.0247
Metadados dentro dos chunkings e armazenamento unificado	0.0137
Metadados dentro dos chunkings e segmentação por tipo de documento	0.0882

Os resultados demonstram que a abordagem de incluir metadados diretamente nos chunkings e segmentar por tipo de documento proporciona um ganho significativo na recuperação da informação, garantindo um melhor contexto e maior precisão na busca legislativa.

2. Pré-processamento de dados

A fase de pré-processamento dos dados, isto é, utilizar as ferramentas para redução de ruídos e transformação para linguagem natural, é uma etapa que influencia diretamente na qualidade e otimização do processamento da IA. De modo específico, a prática se faz muito útil em contextos como tradução, resumo e associação entre textos, dada sua capacidade de destrinchar, desde o início, as complexas estruturas gramaticais e semânticas.

Nas circunstâncias do projeto, porém, o contexto relativo e a formatação jurídica que os textos apresentam são essenciais para a preservação completa do sentido e posterior reconstrução das informações a serem divididas pelo método de chunking. Portanto, práticas como o stemming, lemmatizing e tokenização se mostraram ineficazes, visto que têm por objetivo retirar possíveis flexões verbo-nominais e tratar os tokens individualmente.

Das práticas, realizamos a normalização do texto para padronizar sua estrutura e facilitar a busca semântica. Isso incluiu a substituição de abreviações comuns, como "art." por "artigo", garantindo maior clareza para os modelos de linguagem. Também foram removidos caracteres especiais e espaçamentos desnecessários, reduzindo o ruído nos dados e melhorando a consistência do texto.

Por fim, dado que a origem de parte dos documentos vem de arquivos html e a outra do formato pdf, foram utilizadas bibliotecas para a retirada de tags e sinalizações ambíguas (“\”), utilizadas na linguagem.

3. Chunkings

O método de Chunking (também referido como segmentação) é o processo de dividir um texto em unidades menores, conhecidas como *chunks*. As partes podem pequenas frases ou textos completos, a depender da aplicação. É de suma importância que a técnica utilizada seja eficaz em segmentar o bloco em partes semanticamente suficientes para que, no processo de recuperação desses dados, não haja perdas grandes de sentido e o processamento da LLM seja mais rápido.

Para medir o desempenho de cada uma das estratégias aplicadas durante o projeto, foram levados em consideração a similaridade dentro do bloco, a quantidade de tokens e a fragmentação de informações em relação a uma mesma query e modelo de embedding. A query em questão se refere ao veto nº 34 e, por consequência, os documentos analisados são extraídos da base de dados dos vetos.

3.1. Estratégias utilizadas

Estratégia	Quantidade de chunks	Índice de Jaccard	Média de Similaridade	Tamanho médio de chunks	Desvio padrão	Semântica
Splitting (Paragraph-based chunking)	168	0.0526	0.6868	489.7976	443.5004	4.0000
Token-based chunking (grammatical-based)	489	0.0000	0.8482*	134.7075	137.2934	0.0000
Recursive chunking (fixed-size + overlapping)	651	0.0312	0.6056	148.2703	11.8708	2.0000
Semantic chunking (statistical chunker)	277	0.0385	0.6919	323.6390	345.3801	4.5000
Dynamic chunking	67	0.0000	0.6856	1343.5373	807.6146	3.5000

* Apesar da alta taxa de similaridade, grande parte dos chunks foram ignorados no cálculo por possuírem pouca ou nenhuma informação.

Tabela 3.1 - Estratégias de chunking

3.2. Explicação das métricas utilizadas:

Quantidade de chunks: A quantidade de blocos gerados a partir dos textos da base.
Índice de Jaccard: Mede a similaridade entre os documentos recuperados através dos embeddings e os documentos referentes ao tópico em questão. É interessante usá-lo como parâmetro de comparação nas formas de chunkenização pois aponta, de forma quantitativa, a qualidade semântica do chunk. Ele pode ser calculado através da fórmula $Jaccard = \frac{A \cap B}{A \cup B}$, onde A e B são conjuntos de tamanho k, sendo A gerado pelo retrieval e B o *gold set* - conjunto com as informações esperadas.
Média de Similaridade: A similaridade, aqui calculada através do cosseno dos vetores, é calculada entre os elementos de um chunk, separados por ponto. A média é feita apenas com os blocos que possuem mais de uma sentença, entendendo que aqueles possuem apenas uma possuem sentido completo.
Tamanho médio e desvio padrão: O tamanho do chunk é calculado em caracteres e então é feita a média de todos os chunks de todos os arquivos. Já o desvio padrão é a média da diferença desse tamanho médio.
Semântica: Avaliado de forma manual, foram analisados a estrutura dos textos em relação ao corpo completo e a presença de elementos irrelevantes ou mal distribuídos, de forma que prejudique a recuperação posteriormente, e então foi atribuída uma nota de 0 a 5.

4. Embeddings

Embeddings são representações numéricas de dados usados para capturar relações semânticas e estruturais em espaços de alta dimensionalidade. No processamento de

linguagem natural (PLN), por exemplo, embeddings de palavras representam termos em um espaço vetorial, de forma que palavras com significados semelhantes fiquem próximas umas das outras.

Nesta seção, foi avaliado modelos de embeddings aplicados a um conjunto de dados jurídicos, analisando seu desempenho em termos de similaridade, distância entre vetores e recuperação de informação.

4.1. Modelos utilizados:

Modelo	Tamanho (G)	Tempo de execução (segundos)	Dimensão dos embeddings
stjiris/bert-large-portuguese-cased-legal-mlm-sts-v1.0	1, 3	3706.12	1024
stjiris/bert-large-portuguese-cased-legal-mlm-sts-v1	1, 3	3854.95	1024
sentence-transformers/LaBSE	1,8	1192.95	768
sentence-transformers/paraphrase-multilingual-mpnet-base-v2	0.458	776.89	768
alfaneo/bertimbau-base-portuguese-sts	0.417	1038.05	768
stjiris/bert-large-portuguese-cased-legal-tsdae-gpl-nli-sts-MetaKD-v1	1, 3	3728.26	1024
alfaneo/jurisbert-base-portuguese-sts	0.417	851.68	768
PORTULAN/serafim-100m-portuguese-pt-sentence-encoder-ir	0.417	860.19	768
BAAI/bge-small-en-v1.5	0.133	351.41	384

Tabela 4.1 - Modelos de embeddings

4.2. Desempenho dos modelos de embeddings:

Modelos	Similaridade cosseno	Distância Euclidiana média	Precisão@5	Recall@5	MRR@10	NDCG@10
stjiris/bert-large-portuguese-cased-legal-mlm-sts-v1.	0.4531	1.0302	0.2000	0.2500	1.0000	1.0000

0						
stjiris/bert-large-portuguese-cased-legal-mlm-sts-v1	0.4276	1.0547	0.2000	0.2500	1.0000	1.0000
sentence-transformers/LaBSE	0.5068	0.9765	0.2000	0.2500	0.3333	0.5000
sentence-transformers/paraphrase-multilingual-mpnet-base-v2	0.5336	0.9442	0.2000	0.2500	1.0000	1.0000
alfaneo/bertimbau-base-portuguese-sts	0.6446	0.8280	0.0000	0.0000	0.0000	0.0000
stjiris/bert-large-portuguese-cased-legal-tsdae-gpl-nli-sts-MetaKD-v1	0.4269	1.0534	0.2000	0.2500	1.0000	1.0000
alfaneo/jurisbert-base-portuguese-sts	0.6422	0.8236	0.0000	0.0000	0.0000	0.0000
PORTULAN/serafim-100m-portuguese-pt-sentence-encoder-ir	0.5744	0.9067	0.2000	0.2500	1.0000	1.0000
BAAI/bge-small-en-v1.5	0.7439	0.7054	0.2000	0.2500	1.0000	1.0000

Tabela 4.2 - Desempenho dos modelos de embeddings

4.3. Explicação das métricas utilizadas:

Similaridade Cosseno: Mede o alinhamento entre os vetores. Define-se como o cosseno do ângulo entre dois vetores u e v no espaço vetorial:

$$cos(\theta) = \frac{u.v}{||u||\,||v||}$$

Varia entre -1 (vetores totalmente opostos) e 1 (vetores idênticos).

Distância Euclidiana Média: Mede a separação entre os vetores no espaço vetorial. Calculada como a norma L_2, entre dois vetores u e v:

$$d(u, v) = ||u - v||_2 = \sqrt{\sum_i (u_i - v_i)^2}$$

valores menores indicam maior proximidade.
Precisão@K: Mede a proporção de vizinhos recuperados que são relevantes dentro do top-K resultados. Define-se como a fração dos K primeiros resultados que são relevantes:

$$Precisão@K = \frac{|relevantes\ recuperados \cap top{-}K\ resultados}{k}$$

Quanto maior o valor, melhor o posicionamento dos resultados relevantes.

Recall@K: Mede a proporção de vizinhos relevantes encontrados em relação ao total de vizinhos relevantes existentes:

$$Recall@K = \frac{|relevantes\ recuperados \cap top{-}K\ resultados}{|total\ de\ relevantes|}$$

MRR (Mean Reciprocal Rank): Mede a posição média do primeiro resultado relevante, priorizando posições mais altas no ranking. Define-se como a média da recíproca da posição rir_iri do primeiro item relevante para cada consulta:

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{r_i}$$

NDCG (Normalized Discounted Cumulative Gain): Avalia a qualidade do ranking ponderando a relevância dos itens pela posição i:

$$DCG@K = \sum_{i=1}^{K} \frac{relevância(i)}{log_2(i+1)}$$

Normalizado pelo valor ideal IDCG@K, obtendo:

$$NDCG@K = \frac{DCG@K}{IDCG@K}$$

4.4. Análise Visual dos resultados:

Os gráficos a seguir ilustram a distribuição das similaridades e distâncias entre embeddings:

Gráficos dos modelos de embeddings

5. Large Language Models (LLMs)

LLMs são modelos de inteligência artificial treinados em grandes volumes de texto para entender e gerar linguagem natural. Eles utilizam arquiteturas avançadas de deep learning para processar e prever palavras com base em contexto.

Nesta seção, foram avaliados LLMs aplicados a um conjunto de dados jurídicos, analisando seu desempenho em termos de precisão e relevância das respostas geradas e o tempo médio de resposta.

Para a avaliação foram elaboradas perguntas e respostas de referência para as mesmas com base nos dados jurídicos fornecidos. Foram elaboradas três tipos de perguntas para cada um dos três tipos de documentos fornecidos: leis, vetos e atividade legislativa dos projetos de lei. Quanto aos tipos de perguntas elas foram separadas em:

Consultas diretas: Testam a recuperação exata da informação e a precisão da base de dados.
Perguntas interpretativas: Avaliam a capacidade da LLM de resumir e explicar o conteúdo.
Perguntas contextuais: Testam se a RAG consegue buscar múltiplas fontes e relacionar informações.

As respostas de referência foram elaboradas com base nos dados fornecidos em conjunto com o auxílio de inteligências artificiais, para algumas perguntas foram passadas as informações relacionadas a ela e logo após foi realizada a mesma pergunta à IA, a melhor resposta com base na conformidade com os dados presentes nos arquivos foi selecionada como resposta de referência para a pergunta em questão.

5.1. Desempenho dos LLMs:

Modelo	Tempo médio de resposta (segundos)	BERTScore médio (F1)	Precisão média	Relevância média	Clareza média	Completude média
llama3-8b-8192	5.39	0.6860	2.3	3.25	3.7	2.3
llama-3.3-70b-versatile	7.51	0.6454	2.15	2.75	3.2	2.15
gemma2-9b-it	4.66	0.6839	2.1	2.5	3.1	2.1
qwen-2.5-32b	5.45	0.6727	2.1	2.7	3.1	2.1
deepseek-r1-distill-llama-70b	15.80	0.6424	2.0	2.5	2.95	2.0

Tabela 5.1 - Desempenho dos LLMs

5.2. Explicação das métricas utilizadas:

Tempo Médio de Resposta: O tempo médio de resposta mede o tempo, em segundos, que o LLM leva para gerar uma resposta após receber um prompt. Essa métrica é fundamental para avaliar a eficiência do modelo.

BERTScore Médio (F1): Avalia a qualidade das respostas geradas, utilizando a similaridade semântica entre as palavras. Ao contrário de métricas tradicionais, que comparam palavras exatas, o BERTScore usa embeddings de palavras para medir o quanto as respostas geradas se alinham com a resposta de referência em termos de significado. O **F1** combina duas métricas fundamentais: **precisão** e **recall**. A **precisão** é a proporção de palavras geradas que são relevantes (ou correspondem ao significado da referência). O **recall** é a proporção de palavras relevantes da referência que estão presentes na geração. O **F1** reflete o equilíbrio entre essas duas métricas, sendo útil para entender a qualidade global da resposta.
O BERTScore (F1) varia entre **0** e **1**. Um valor próximo de **1** indica uma alta similaridade semântica entre a resposta gerada e a referência, enquanto valores próximos de **0** sugerem baixa correspondência.

Avaliação Humana: Foi utilizada para avaliar a qualidade das respostas geradas com notas de 1 a 5 em relação aos seguintes critérios: **precisão**, **relevância**, **clareza** e **completude**.
Precisão: Refere-se à exatidão das informações contidas na resposta, se a resposta contém informações corretas.
Relevância: Avalia o grau de correspondência entre a resposta e a pergunta, se a resposta está relacionada à pergunta.
Clareza: Foca na capacidade da resposta de ser facilmente entendida, se ela é clara e compreensível.
Completude: Considera se a resposta abrange todos os aspectos da pergunta, se ela está completa ou incompleta.

5.3. Análise visual dos resultados:

Os gráficos a seguir ilustram a comparação entre as métricas utilizadas para os modelos testados:

Gráficos das métricas dos LLMs

Conclusão

O desenvolvimento do sistema de Geração Aumentada de Recuperação (RAG) para consulta de atividades legislativas mostrou-se uma solução eficaz para aprimorar a busca e interpretação de documentos legais. A combinação de técnicas de chunking, embeddings e modelos de LLMs permitiu melhorar a precisão e a relevância das respostas, tornando a consulta legislativa mais intuitiva e eficiente.

Os resultados obtidos demonstraram que a abordagem proposta foi capaz de superar limitações de métodos tradicionais de recuperação de informações, proporcionando uma melhor organização e acessibilidade aos dados legislativos. No entanto, desafios ainda permanecem, como a necessidade de aprimorar o balanceamento entre precisão e cobertura da informação recuperada.

Como trabalho futuro, sugere-se a ampliação do conjunto de dados utilizado, a incorporação de técnicas de fine-tuning para modelos de linguagem e a avaliação de novas arquiteturas de IA que possam aprimorar ainda mais a qualidade das respostas geradas. Assim, espera-se que este projeto contribua significativamente para o desenvolvimento de ferramentas mais eficazes no âmbito da consulta e análise legislativa.