Proposta de um Banco de Dados Vetorial para a Análise Semântica de Atos Normativos do Conselho Nacional de Justiça (CNJ)

Resumo

O Conselho Nacional de Justiça publica, em média, XXX atos normativos por ano. O alcance dos efeitos normativos supera o âmbito interno do Poder Judiciário, alcançando pilares basilares da Sociedade, realizando Justiça em todos os seus aspectos. Este artigo descreve estatisticamente todos os atos normativos do CNJ e detalha o desenvolvimento de uma solução de engenharia de dados para transformar os atos normativos em um banco de dados vetorial. Utilizando técnicas de web scraping para a coleta de dados, processamento de linguagem natural (PLN) para a limpeza e estruturação do texto, e modelos de deep learning para a geração de embeddings semânticos, o projeto culmina na criação de um repositório otimizado para buscas por similaridade de conteúdo. A metodologia aqui apresentada demonstra um fluxo de trabalho completo, desde a extração de dados brutos do site do CNJ até sua inserção em um banco de dados PostgreSQL com a extensão PGVector, viabilizando futuras aplicações de inteligência artificial no domínio jurídico.

Introdução

O CNJ é uma instituição pública criada pela Emenda Constitucional nº 45, em 2004, com o objetivo de aprimorar o trabalho do Poder Judiciário brasileiro. Sua missão é promover o desenvolvimento do Judiciário em benefício da sociedade, atuando principalmente no controle e na transparência administrativa e processual.

Para cumprir suas atribuições, o Conselho edita diversos atos normativos que regulamentam a atuação dos órgãos judiciais. Essas normas, como Resoluções, Recomendações, Provimentos, Portarias, Instruções Normativas, e outros, estabelecem diretrizes e procedimentos em áreas como gestão processual, administrativa e financeira, além de políticas judiciárias de alcance nacional. Tais atos buscam a padronização e a eficiência dos serviços judiciais, garantindo maior transparência e efetividade na prestação jurisdicional em todo o país.

Diante do volume e da complexidade desses atos normativos, a busca e a análise de informações por meio de métodos tradicionais, como a pesquisa por palavras-chave, tornam-se ineficientes e limitadas. Imagine tentar encontrar informações em uma biblioteca gigantesca onde você só pode pesquisar pelos títulos exatos dos livros. Você perderia inúmeras obras relevantes que simplesmente usam palavras diferentes para descrever o mesmo assunto. É exatamente esse o desafio que enfrentamos com a imensa quantidade de atos normativos.

Para superar essa barreira, o projeto aborda a recuperação da informação através da similaridade semântica. Em vez de apenas ler as palavras, a consulta traduz a essência de cada ato normativo em uma espécie de endereço digital, uma representação numérica única chamada *embedding*. Ao organizar esses “endereços” em um banco de dados especializado (o PostgreSQL com a extensão PGVector), ganhamos uma capacidade peculiar: a de buscar por similaridade de ideias.

Objetivos

O objetivo principal deste trabalho é projetar, implementar e documentar a criação de um banco de dados vetorial com atualização automatizada. Os objetivos secundários são:

* Descrever estatisticamente a base de atos normativos do CNJ;
* Desenvolver um script de *web scraping* para extrair de forma automatizada os metadados e o conteúdo textual;
* Implementar funções para a limpeza e pré-processamento do texto jurídico, removendo ruídos e padronizando conteúdo;
* Criar uma estratégia de chunking com o propósito de dividir os documentos em fragmentos textuais lógicos e semanticamente relevantes;
* Selecionar um modelo pré-treinado de transformação de sentença para converter os textos em *embeddings* vetoriais de alta dimensão;
* Inserir as informações em um banco de dados vetorial PGVector.

Justificativa

A criação de um banco de dados vetorial para os normativos do CNJ justifica-se pela necessidade de modernizar o acesso e a análise de informações jurídicas. Servidores e magistrados do Judiciário, profissionais do direito, pesquisadores e gestores públicos poderiam se beneficiar de ferramentas de busca semântica que permitam encontrar normativos relacionados a um tema específico, mesmo que a consulta não utilize os termos exatos presentes no documento.

Esta infraestrutura de dados é a base para o desenvolvimento de sistemas avançados, como: (procurar referencia)

* Sistemas de Pergunta e Resposta (Q&A): Para responder a questões formuladas em linguagem natural sobre as regulamentações do CNJ.
* Análise de Tendências e Padrões: Para identificar a evolução de temas e entendimentos ao longo do tempo.
* Geração Aumentada por Recuperação (RAG): Para enriquecer modelos de linguagem generativos (LLMs) com conhecimento jurídico específico e atualizado, minimizando a ocorrência de "alucinações".

Abordagem Teórica

<http://repositorio.unb.br/handle/10482/52246>

Webscraping de conteúdo jurídico

PLN

Sentence transformers e embeddings

Banco de dados vetorial e PGVector

Metodologia

O método utilizado deste projeto se baseia em um script Python, contido em um notebook jupyter, disponibilizado no <>, que orquestra todo o processo em sete etapas principais:

Passo 1: Importação das bibliotecas

O ambiente é inicializado com a importação das bibliotecas necessárias: requests e BeautifulSoup para o scraping; psycopg2 para a conexão com o PostgreSQL; sentence\_transformers para carregar o modelo de IA; e outras bibliotecas auxiliares como re, unicodedata e dotenv para manipulação de texto e gerenciamento de credenciais.

Passo 2: Configurações Gerais

Um bloco centralizado define as variáveis de controle do script. Isso inclui a URL raiz do portal de atos do CNJ (https://atos.cnj.jus.br), limites de páginas e links a serem processados, o nome do modelo de embedding (sentence-transformers/LaBSE), e as credenciais de acesso ao banco de dados, carregadas de forma segura de um arquivo .env.

Passo 3: Funções de limpeza e Processamento de Texto

Este passo implementa o núcleo de PLN do projeto. A função parse\_and\_clean\_html\_content extrai o texto puro do HTML, removendo tags irrelevantes. A função clean\_legal\_text realiza uma limpeza fina, eliminando ruídos comuns em documentos legais. A função mais crítica, chunk\_structured\_legal\_text, utiliza expressões regulares para "escanear" o texto e dividi-lo em segmentos estruturados, identificando marcadores como "Preâmbulo", "Art.", "Parágrafo" e "CONSIDERANDO". Ela também extrai metadados importantes, como o nome da autoridade signatária do ato.

Passo 4: Funções de Acesso à Rede e ao Banco de Dados

As funções de entrada e saída (I/O) são isoladas neste bloco. fetch\_page\_content realiza as requisições HTTP para obter o conteúdo das páginas, com um mecanismo de atraso para evitar sobrecarregar o servidor do CNJ. connect\_to\_postgres estabelece a conexão com o banco de dados. A função create\_pg\_table prepara a infraestrutura no banco, criando a tabela de destino com uma coluna para o texto, uma para metadados em formato JSONB e a coluna embedding do tipo VECTOR. Por fim, insert\_data\_with\_embedding recebe um chunk de texto, gera seu embedding usando o modelo LaBSE e insere o registro completo na tabela.

Passo 5: Função Principal de Orquestração

A função run\_scraper atua como o maestro do processo. Ela inicia a criação da tabela no banco e entra em um laço de repetição para navegar pelas páginas de listagem de atos. Em cada página, extrai os links para os documentos individuais. Para cada documento, ela extrai os metadados e o conteúdo, aplica as funções de limpeza e chunking (Passo 3), e por fim, para cada chunk gerado, invoca a função de inserção (Passo 4) que gera o embedding e armazena os dados. A biblioteca tqdm é utilizada para exibir uma barra de progresso.

Passo 6: Inicialização e Execução

A célula final do notebook inicia o processo. Primeiramente, carrega o modelo SentenceTransformer na memória. Em seguida, estabelece a conexão com o banco de dados dentro de um bloco try...finally para garantir que a conexão seja sempre fechada, mesmo em caso de erros. Com o modelo carregado e a conexão ativa, a função run\_scraper é chamada para executar o pipeline completo.

Passo 7: Descrição e Avaliação

A função describe retorna as principais medidas resumo e gráficos que descrevem o banco de dados em questão. A avaliação foi realizada substituindo algumas palavras por uma máscara, e analisado se o sistema retornou com sucesso a palavra ocultada.

Resultados e Discussão

6. Conclusão