UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Aplicação de RAG para identificação de atos judiciais similares na Justiça Eleitoral

Ramon Gouveia Rodrigues

Monografia - MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

Ramon Gouveia Rodrigues

Aplicação de RAG para identificação de atos judiciais similares na Justiça Eleitoral

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Ricardo Marcondes Marcacini

Versão original

São Carlos

2024

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

G719a

Gouveia Rodrigues, Ramon
Aplicação de RAG para identificação de atos judiciais similares na Justiça Eleitoral / Ramon Gouveia Rodrigues; orientador Ricardo Marcondes Marcacini. -- São Carlos, 2024.
65 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2024.

1. Retrieval-Augmented Generation - RAG. 2. Large Language Model - LLM. 3. Justiça Eleitoral. 4. Similaridade de documentos. 5. Resumo. I. Marcondes Marcacini, Ricardo, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Ramon Gouveia Rodrigues

Application of RAG for the identification of similar judicial acts in Electoral Justice

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Advisor: Prof. Dr. Ricardo Marcondes Marcacini

Original version

São Carlos

2024

Este trabalho é dedicado à minha querida família, que sempre acreditou em mim e me motivou a seguir em frente. Dedico também aos pesquisadores de TI, pelo revolucionário momento em que vivemos graças aos avanços em Inteligência Artificial.

AGRADECIMENTOS

A realização deste trabalho só foi possível graças ao apoio e à colaboração de diversas pessoas e instituições, que, direta ou indiretamente, contribuíram significativamente para o seu desenvolvimento.

Em primeiro lugar, expresso minha profunda gratidão à minha família, especialmente aos meus pais e à minha irmã, pelo apoio incondicional e pela confiança em meu potencial. Seus incentivos foram fundamentais para que eu perseverasse ao longo desta jornada.

À minha namorada, registro meu sincero agradecimento por todo o amor, compreensão e paciência. Esteve ao meu lado nos momentos mais desafiadores, e sua presença constante foi uma fonte inestimável de inspiração e força.

Agradeço também ao meu orientador, Prof. Dr. Ricardo Marcacini, por suas valiosas sugestões e orientação. Seu vasto conhecimento e experiência foram essenciais para a realização deste trabalho. São raros os estudiosos de Inteligência Artificial que conseguem unir uma prática tão sólida a um conhecimento tão atualizado das inovações da área.

Manifesto também minha gratidão aos meus colegas de curso, cujos debates, trocas de ideias e apoio mútuo tornaram essa jornada mais leve e intelectualmente enriquecedora.

Agradeço ainda aos patrocínios recebidos ao longo deste MBA, que incluem a bolsa de estudos concedida pela Fundação para o Incremento da Pesquisa e do Aperfeiçoamento Industrial (FIPAI) e a infraestrutura computacional fornecida por meio do *Developer Program Members* da Nvidia, essencial para a execução dos experimentos deste trabalho.

Por fim, agradeço ao Instituto de Ciências Matemáticas e de Computação (ICMC) da Universidade de São Paulo (USP) por me proporcionar a oportunidade de realizar este trabalho na melhor universidade da América Latina e uma das mais prestigiadas do mundo. Um agradecimento especial à coordenadora Profa. Dra. Solange Rezende, assim como a todo o corpo docente e administrativo.

A todos, meu muito obrigado!

"A Inteligência Artificial será a maior conquista da humanidade.
Mas também pode ser a última."
Stephen Hawking

RESUMO

G. Rodrigues, Ramon **Application of RAG for the identification of similar judicial acts in Electoral Justice**. 2024. 65 p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

O sistema jurídico brasileiro enfrenta desafios de eficiência nas tramitações processuais devido à carência de ferramentas tecnológicas adequadas. A Inteligência Artificial (IA) emerge como uma solução promissora para otimizar esses processos. Este trabalho propôs uma *Retrieval-Augmented Generation* (RAG) para identificar atos judiciais similares na Justiça Eleitoral (JE), visando aumentar a eficiência dessa tarefa. A metodologia incluiu experimentos para avaliar a eficácia da RAG em aprimorar as respostas de sistemas tradicionais de recuperação de informação. Foi utilizado o Elasticsearch como sistema tradicional e o Llama 3.1 como *Large Language Model* para refinar as respostas. Os resultados mostraram que a RAG proposta melhorou significativamente a qualidade das respostas, especialmente nos resumos dos atos judiciais. A entropia foi nula em todos os experimentos ao reordenar os documentos mais similares, e os resumos gerados apresentaram uma pontuação BLANC média de 0,138, destacando-se pela fluidez, compreensibilidade, informatividade e concisão com que os textos foram gerados. A consistência dos resultados foi mantida entre diferentes tribunais eleitorais, com a observação de que resumos de textos mais curtos tendem a ser melhores. Assim, a conclusão foi de que a adoção de uma RAG para a busca de atos judiciais similares oferece ganhos significativos em relação aos sistemas tradicionais, permitindo identificar documentos realmente similares e fornecendo respostas resumidas que facilitam a organização e celeridade para o usuário final.

Palavras-chave: *Retrieval-Augmented Generation* (RAG). *Large Language Model* (LLM). Justiça Eleitoral. Similaridade de documentos. Resumo.

ABSTRACT

G. Rodrigues, Ramon **Application of RAG for the identification of similar judicial acts in Electoral Justice**. 2024. 65 p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

The Brazilian legal system faces efficiency challenges in procedural processing due to the lack of adequate technological tools. Artificial Intelligence (AI) emerges as a promising solution to optimize these processes. This study proposed a Retrieval-Augmented Generation (RAG) approach to identify similar judicial acts in the Electoral Justice (JE), aiming to enhance the efficiency of this task. The methodology included experiments to assess the effectiveness of the RAG in improving the responses of traditional information retrieval systems. Elasticsearch was used as the traditional system, and Llama 3.1 was employed as the Large Language Model to refine the responses. The results showed that the proposed RAG significantly improved the quality of responses, particularly in the summaries of judicial acts. The entropy was zero in all experiments when reordering the most similar documents, and the generated summaries achieved an average BLANC score of 0,138, standing out for the fluency, comprehensibility, informativeness, and conciseness with which the texts were generated. The consistency of results was maintained across different electoral courts, with the observation that summaries of shorter texts tend to be better. Thus, the conclusion was that adoption of a RAG for searching similar judicial acts offers significant advantages over traditional systems, allowing for the identification of truly similar documents and providing summarized responses that facilitate organization and efficiency for the end user.

Keywords: *Retrieval-Augmented Generation* (RAG). *Large Language Model* (LLM). Electoral Justice. Document similarity. Summary

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
abnTeX	ABsurdas Normas para TeX
AILA	*Artificial Intelligence for Legal Assistance*
BERT	*Bidirectional Encoder Representations from Transformers*
BLANC	*Bacronymic Language model Approach for summary quality estimatioN*
CLIR	*Cross-Language Information Retrieval*
CNJ	Conselho Nacional de Justiça
COLIEE	*Competition on Legal Information Extraction/Entailment*
CSJT	Conselho Superior da Justiça do Trabalho
DLRM	Modelo Diversificado de Recuperação de Casos Jurídicos
DSL	*Domain-Specific Language*
GPT	*Generative Pretrained Transformer*
IA	Inteligência Artificial
IBGE	Instituto Brasileiro de Geografia e Estatística
JE	Justiça Eleitoral
JSON	*JavaScript Object Notation*
LaTeX	Lamport TeX
LIR	Legal Information Retrieval
LLM	*Large Language Model*
ML	*Machine Learning*
PC	Prestação de Contas
PLN	Processamento de Línguas Naturais
RAG	*Retrieval-Augmented Generation*
RCAND	Registro de Candidatura

RE	Recurso Eleitoral
REC	Recurso
RESPE	Recurso Especial Eleitoral
RNN	Rede Neural Recorrente
RSLP	Removedor de Sufixos da Língua Portuguesa
SAILER	*Structure-Aware pre-traIned language model for LEgal case Retrieval*
SVM	*Support Vector Machine*
STF	Supremo Tribunal Federal
STJ	Superior Tribunal de Justiça
STS	*Short Sentence Similarity*
TJBA	Tribunal de Justiça da Bahia
TJCE	Tribunal de Justiça do Ceará
TJES	Tribunal de Justiça do Estado de Sergipe
TJMG	Tribunal de Justiça do Estado de Minas Gerais
TJPB	Tribunal de Justiça do Estado da Paraíba
TRF2	Tribunal Regional Federal da 2ª Região
TRE-BA	Tribunal Regional Eleitoral da Bahia
TRE-GO	Tribunal Regional Eleitoral de Goiás
TRE-MG	Tribunal Regional Eleitoral de Minas Gerais
TRE-RJ	Tribunal Regional Eleitoral do Rio de Janeiro
TRE-SP	Tribunal Regional Eleitoral de São Paulo
TREs	Tribunais Regionais Eleitorais
TSE	Tribunal Superior Eleitoral
USPSC	Campus USP de São Carlos
USP	Universidade de São Paulo

SUMÁRIO

1 INTRODUÇÃO

Nos últimos anos, o Poder Judiciário tem enfrentado desafios significativos relacionados ao crescente volume de processos, culminando em uma sobrecarga de trabalho para os profissionais jurídicos, como magistrados, servidores e advogados. A natureza complexa e a diversidade de atos judiciais tornam árdua a tarefa de lidar eficientemente com a vasta quantidade de informações disponíveis.

Segundo o relatório Justiça em Números 2023 ((CNJ), 2023), criado pelo Conselho Nacional de Justiça (CNJ) para mensurar o Poder Judiciário, a quantidade de casos novos pós-Codiv-19 saltou de 25,8 para 31,5 milhões em apenas 2 anos, um aumento de 22%, conforme Figura 1.

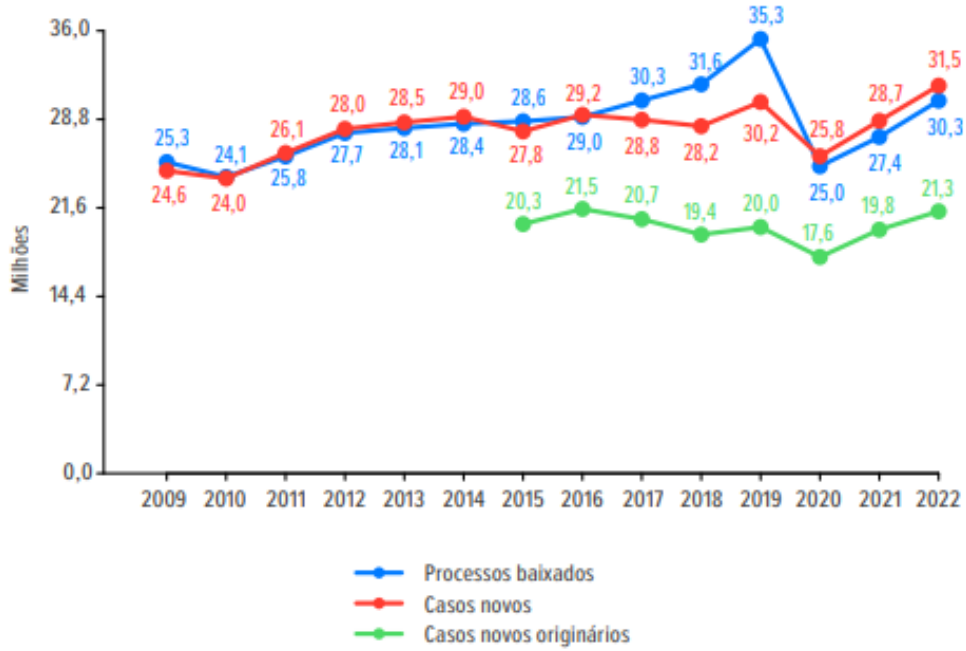

Figura 1 – Evolução do volume de processos no Poder Judiciário durante os anos.

Fonte: (CNJ) (2023)

.

Além disso, a quantidade de casos novos pós-Covid-19 tem consistentemente ultrapassado a quantidade de processos baixados. Isso sugere que, ao longo dos últimos anos, o Poder Judiciário tem enfrentado dificuldades em julgar um número maior de processos em comparação com os que ingressam nos tribunais, o que mostra o grande aumento da quantidade de documentos jurídicos sendo produzidos.

O impacto disso é mostrado em Rocha (2011), que ao analisar o uso dos serviços de

biblioteca pelos assessores dos ministros do Supremo Tribunal Federal (STF), identificou que as jurisprudência, decisões de um tribunal sobre um determinado assunto, são difíceis de serem encontradas por eles, o que poderia comprometer as sentenças proferidas.

Nessa mesma linha, o trabalho em Sansone e Sperlí (2022) levantou o estado da arte de sistemas de Recuperação de Informações Jurídicas (Legal Information Retrieval - LIR) e concluiu que o resumo, a busca e a compreensão de documentos jurídicos ainda são desafios em aberto na literatura. Também reforçou a necessidade do desenvolvimento de metodologias de processamento de documentos e de extração de informações úteis para melhorar a recuperação de informações relevantes nesse domínio.

Nesse cenário, a Inteligência Artificial (IA) emerge como uma ferramenta crucial para aprimorar o funcionamento do sistema judiciário, proporcionando ganhos substanciais em termos de eficiência, custo e qualidade na prestação jurisdicional. Segundo um levantamento feito em 2022 pelo CNJ, o número de iniciativas de IA no judiciário brasileiro aumentou 171% em relação a 2021, totalizando 111 projetos desenvolvidos ou em desenvolvimento nos tribunais ((CNJ), 2022b).

Entre esses projetos, podem ser citados, como exemplos, o Janus, idealizado pelo Tribunal Regional Eleitoral da Bahia (TRE-BA), que visa automatizar tarefas repetitivas dos julgamentos de pedidos de candidatura e prestação de contas eleitorais; o Gemini, coordenado pelo Conselho Superior da Justiça do Trabalho (CSJT), que agrupa processos por similaridade de tema nas unidades de primeiro e segundo grau da Justiça do Trabalho; e a Sofia, assistente virtual de atendimento do Tribunal de Justiça da Bahia (TJBA) que utiliza IA na triagem automática de processos ((CNJ), 2022a).

No entanto, o Poder Judiciário ainda carece de uma ferramenta de IA que auxilie na identificação e análise de atos judiciais. A vastidão de jurisprudências acumuladas ao longo dos anos exige algoritmos de IA que possam identificar padrões, relações e similaridades entre elas. Ao empregar técnicas de Processamento de Línguas Naturais (PLN) e aprendizado de máquina para realizar essa tarefa, a IA não apenas viabiliza uma gestão mais eficiente do volume de processos, mas também aprimora a consistência e a uniformidade nas decisões, promovendo, assim, uma jurisprudência mais coesa e previsível.

Além disso, a crescente demanda por transparência e acesso à informação no âmbito judicial torna essencial a implementação de abordagens inovadoras. Ferramentas de IA voltadas para o tratamento de atos judiciais não apenas simplificam a recuperação de informações relevantes, mas também possibilitam uma compreensão mais rápida e precisa das jurisprudências. Ademais, a integração da IA no processo decisório judicial moderniza as práticas existentes, fortalecendo a confiança na justiça e promovendo um sistema mais transparente e eficaz.

Nesse sentido, os Modelos de Linguagem de Grande Escala (*Large Language*

Models - LLMs) são uma alternativa promissora para o tratamento de atos judiciais e o aprimoramento das buscas de sistemas de recuperação informação (Shu *et al.*, 2024). Recentemente, diferentes LLMs de código aberto, como o LLama 3.1 (Dubey *et al.*, 2024), têm sido compartilhados, ampliando a capacidade de interpretação de jurisprudências (Surden, 2023). É nesse contexto que este trabalho é proposto, especialmente com uso de LLMs para *Retrieval-Augmented Generation* (RAG).

O processo de RAG consiste, primeiramente, na utilização de um sistema de recuperação de informações para buscar em uma base de dados judicial e identificar os documentos mais relevantes para uma consulta específica. Em seguida, o LLM utiliza esses documentos como contexto para gerar uma resposta textual aprimorada, seja destacando as informações mais relevantes ou produzindo uma sumarização que potencializa a experiência do usuário.

Sendo assim, o objetivo geral deste trabalho é propor uma RAG que seja capaz de identificar atos judiciais similares da Justiça Eleitoral (JE). Para atender a esse objetivo geral, este estudo possui os seguintes objetivos específicos:

1. Avaliar se a implementação de uma RAG pode melhorar a qualidade das respostas de um sistema de recuperação tradicional quando utilizado para identificar atos judiciais similares;

2. Avaliar se o desempenho de um modelo de LLM para identificar similaridades e melhorar a apresentação dos resultados das buscas de textos jurídicos em português é significativamente impactado quando se altera o tribunal (TSE ou TREs) de origem desses textos.

Ao alcançar esses objetivos, espera-se que este estudo contribua significativamente para o avanço da eficiência do Poder Judiciário, em específico, dos processos da JE. A expectativa é que a exploração dos benefícios práticos da IA na análise de atos judiciais, promova uma justiça mais eficiente, coesa, transparente e adaptada aos desafios contemporâneos que permeiam os sistemas jurídicos.

Este trabalho está organizado da seguinte forma: no Capítulo 1 é contextualizado o problema e apresentado o objetivo deste estudo. O Capítulo 2 apresenta a fundamentação teórica, incluindo trabalhos relacionados à recuperação de informações jurídicas, identificação de documentos similares e o uso da IA no domínio jurídico. No Capítulo 3 é detalhada a metodologia proposta, bem como as métricas utilizadas para avaliar os experimentos. O Capítulo 4 descreve a execução dos experimentos e os resultados obtidos. Por fim, o Capítulo 5 apresenta as conclusões do trabalho baseadas nos resultados discutidos no capítulo anterior.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo fornece um panorama dos avanços mais recentes no uso da IA para a recuperação de informações jurídicas. Inicialmente, são mencionados trabalhos que buscam solucionar os principais desafios do meio jurídico. Posteriormente, são apresentadas pesquisas recentes que exploram a similaridade de documentos. Em seguida, são citados trabalhos que fazem uso de LLMs. Por fim, partindo para uma visão nacional, são mencionados trabalhos brasileiros que aplicam IA em textos em português do domínio jurídico.

2.1 Recuperação de Informações Jurídicas - LIR

Recuperação de Informações Jurídicas (*Legal Information Retrieval* - LIR), cujo nome mais apropriado poderia ser Recuperação e Documentos Jurídicos (Bar-Hillel, 1962; Bourne, 1963), é a subárea da Ciência da Computação que estuda técnicas para encontrar, com a máxima precisão possível, documentos jurídicos, como decisões, estatutos, regulamentos, doutrinas e pareceres, em uma grande base de dados (Fraenkel, 1969). Esta subárea enfrenta uma série de desafios, muitos dos quais são abordados em competições como a *Competition on Legal Information Extraction/Entailment* (COLIEE) (Alberta, 2022) e a *Artificial Intelligence for Legal Assistance* (AILA) (Parikh *et al.*, 2021).

A COLIEE, em particular, apresenta quatro tarefas como desafios: a Tarefa 1 consiste em identificar jurisprudências de apoio com base em uma jurisprudência fornecida; a Tarefa 2 envolve a identificação de parágrafos de jurisprudências relevantes para a resolução de um novo caso; a Tarefa 3 busca estatutos pertinentes para uma questão jurídica específica; e a Tarefa 4 determina se um código civil recuperado da base de dados tem implicações em um caso específico. Já a AILA possui duas tarefas, a primeira consiste em classificar cada sentença de um caso legal com um rótulo de 1 a 7 do nível de retórica; e a segunda busca gerar sumários dos documentos jurídicos.

Entre os trabalhos recentes que buscam solucionar esses desafios usando IA, tem-se, por exemplo, o Šavelka e Ashley (2021), que propôs uma solução usando PLN para classificar quais sentenças, extraídas de uma base de jurisprudência, são úteis para um dispositivo legal específico. Há também o Kanapala *et al.* (2022), que descreveu uma nova abordagem combinando os modelos Hiemstra, BM25 e PL2F para buscar artigos de direito civil relevantes para uma prova de direito utilizando sistemas de votação por maioria para elencar os artigos mais relevantes. Outro trabalho similar, Parashar, Mittal e Mehta (2023) também propôs algoritmos de ranqueamento para ordenar dispositivos legais de acordo com sua relevância para casos específicos.

2.2 Similaridade no domínio jurídico

Muitos artigos de LIR se baseiam na similaridade para encontrar documentos relevantes no domínio jurídico. Segundo Jurafsky e Martin (2009), a similaridade de documentos no âmbito de PLN refere-se à avaliação de quão semelhantes são dois conjuntos de texto em termos de conteúdo, tema ou significado. Essa similaridade pode ser determinada por meio de várias técnicas que analisam e comparam as características linguísticas dos textos, como frequência de palavras (*Term Frequency - Inverse Data Frequency* - TF-IDF), distribuição semântica (Word2Vec, BERT), bem como a estrutura e contexto da linguagem (analise sintática e semântica).

Com o propósito de medir a similaridade, algoritmos de PLN quantificam as relações entre os termos e conceitos presentes nos documentos, possibilitando, assim, a identificação de padrões, co-ocorrências e outros atributos textuais relevantes que contribuem para um entendimento mais profundo da correlação entre textos (Jurafsky; Martin, 2009).

Em Westermann, Savelka e Benyekhlef (2021) e Kim, Rabelo e Goebel (2019), por exemplo, foram desenvolvidos métodos utilizando SVM (*Support Vector Machine*) e TF-IDF para encontrar casos jurídicos similares com base na pontuação de similaridade de seus parágrafos. Enquanto Yang (2020) identificou o nível de distribuição dos termos, calculou a similaridade semântica entre eles e identificou as co-ocorrências para encontrar documentos similares em uma base de dados jurídica.

Outros trabalhos se propuseram a comparar diferentes abordagens para encontrar a similaridade de documentos. É o caso, por exemplo, de Chatterjee *et al.* (2023), que comparou métodos que variaram desde a similaridade entre todos os termos até a similaridade apenas de termos jurídicos, cocitações e links bibliográficos. A abordagem que se destacou foi a de similaridade apenas de termos jurídicos. Já Wagh e Anand (2017), ao comparar abordagens de similaridade baseadas em cosseno e em citações, identificou que a abordagem baseada em citações é mais robusta para identificar casos similares.

Por outro lado, de acordo com Zhang *et al.* (2023), a simples similaridade não é adequada para uma busca eficaz de documentos jurídicos. O estudo ressalta que os usuários muitas vezes procuram não apenas conteúdos similares, mas sim subtópicos específicos relacionados ao tema principal da pesquisa. Diante disso, foi desenvolvido o Modelo Diversificado de Recuperação de Casos Jurídicos (DLRM), que não apenas considera a relevância do tópico principal, mas também as relações entre os subtópicos. Resultados experimentais indicaram que o DLRM superou outros modelos, como BM25, MMR, IA-select, exIA-select e M2DIV, destacando sua eficácia na recuperação de documentos jurídicos relevantes em diferentes contextos e subtemas.

Além disso, segundo Bonab, Sarwar e Allan (2020), quando um modelo de é desenvolvido e treinado em um idioma específico, frequentemente observa-se que seu

desempenho é degradado ao aplicá-lo em textos de outro idioma. Por isso, uma série de estudos tem se dedicado ao desafio do Recuperação de Informação entre Idiomas (*Cross-Language Information Retrieval* - CLIR), especialmente no contexto jurídico.

Em Zhebel, Zubarev e Sochenkov (2020), várias abordagens para CLIR foram discutidas, abrangendo desde métodos tradicionais, como aqueles baseados em mediadores, até abordagens mais contemporâneas, como as baseados em distribuição semântica. Em outro estudo, Zhang e Zhao (2020), foram apresentadas técnicas de tradução que incorporam *embedding* e expansão de consulta. Enquanto em Conneau e Lample (2019) foi proposto o método Smart Shuffling, o qual utiliza a similaridade estatística entre termos para construir um dicionário denso.

2.3 *Large Language Models* - LLMs

Felizmente, com o avanço da IA, modelos modernos como os LLMs tem surgido, facilitando uma compreensão mais profunda dos textos e, por conseguinte, a análise de suas similaridades. Entre os LLMs mais renomados, tem-se o *Generative Pretrained Transformer* (GPT) (Radford *et al.*, 2018) da OpenAI [1] e o BERT (*Bidirectional Encoder Representations from Transformers*) (Devlin *et al.*, 2019) da Google [2], ambos baseados na arquitetura Transformer (Vaswani *et al.*, 2017), que constitui o que há de mais avançado em PLN.

Estes modelos são treinados em extensivos conjuntos de dados textuais, o que permite que aprendam uma grande variedade de padrões de linguagem, desde a sintaxe até nuances semânticas complexas, tornando-os muito flexíveis e poderosos (Vaswani *et al.*, 2017). A introdução do GPT-3 por Brown *et al.* (2020), por exemplo, demonstrou que modelos maiores e bem treinados podem aprender a realizar tarefas para as quais não foram explicitamente treinados, uma técnica conhecida como aprendizagem de poucos exemplos ou "*few-shot learning*".

Por isso, estudos recentes têm investigado o uso de LLM no domínio jurídico para a recuperação de informação. Por exemplo, em Bento e Teive (2023), foram comparados algoritmos tradicionais de *Machine Learning*, como *Naive Bayes*, Árvore de Decisão, Random Forest e SVM, com o ChatGPT (*Transformer*/BERT) na classificação de documentos jurídicos em português. Os resultados revelaram que o modelo baseado em *Transformer* obteve melhores resultados, alcançando uma acurácia acima de 70% na classificação de textos jurídicos. O GPT 3.5, mesmo não sendo treinado no domínio jurídico, conseguiu obter 20% dos melhores F1-Scores médios. Além disso, esse estudo também mostrou que os algoritmos obtiveram melhores resultados do que a classificação humana, que alcançou apenas 60% de acurácia.

[1] https://openai.com/
[2] https://www.google.com/

Em outra pesquisa, Cui *et al.* (2023), foi criado um *chatbot* baseado no OpenLLAMA, denominado ChatLaw, com a combinação de LLM e banco de dados vetorial voltado para textos jurídicos em chinês. Esse modelo possibilitou não apenas a extração de características, mas também a prevenção de alucinações e o cálculo da similaridade entre os textos.

Um trabalho semelhante descrito em Prasad, Boughanem e Dkaki (2024) propôs o MESc, um *framework* hierárquico baseado em *Deep Learning*, com o intuito de prever decisões. Em comparação com LLMs de bilhões de parâmetros, como GPT-Neo e GPT-J, o MESc demonstrou um desempenho significativamente superior, alcançando pelo menos 2 pontos a mais do que esses modelos de última geração.

Por outro lado, o estudo apresentado em Li *et al.* (2023) destacou que modelos pré-treinados de LLM de propósito geral não são adequados para lidar com textos jurídicos, apontando a necessidade de um modelo especializado. Assim, foi proposto o modelo *Structure-Aware pre-traIned language model for LEgal case Retrieval* (SAILER), capaz de extrair informações estruturadas dos documentos e, assim, direcionar a atenção para os aspectos específicos do domínio jurídico.

Semelhantemente, o estudo descrito em Shaghaghian *et al.* (2020) apresentou como os modelos de LLM de propósito geral podem ser customizados para obter um melhor desempenho em textos jurídicos. Os experimentos conduzidos evidenciaram que as tarefas a nível de *token* alcançaram resultados superiores nos modelos pré-treinados em domínios genéricos, enquanto a customização do modelo para um domínio específico beneficiou as tarefas a nível de sentença.

2.4 IA aplicada ao domínio jurídico brasileiro

Dado o imperativo de adaptar modelos, sejam de LLM ou os mais tradicionais, para um contexto específico visando alcançar melhores resultados, são citados abaixo alguns trabalhos de IA voltados para o domínio jurídico brasileiro.

O estudo descrito em Souza *et al.* (2021), por exemplo, comparou diferentes abordagens de pré-processamento, *stemmers*, modelos de linguagem e variantes do BM25 na recuperação de documentos da Câmara dos Deputados do Brasil. Os resultados indicaram que o RSLP e o Savoy Stemming para a redução da dimensionalidade aprimoraram o *pipeline* de busca. Além disso, foi identificado que a combinação de unigramas com bigramas proporciona melhorias significativas nos resultados do BM25.

Outro estudo, Oliveira e Junior (2018), realizou experimentos de lematização em jurisprudências do Tribunal de Justiça do Estado de Sergipe (TJSE). Os resultados indicaram que o Removedor de Sufixos da Língua Portuguesa (RSLP) alcançou a maior redução na dimensionalidade, e que uma abordagem de lematização menos agressiva apresentou o melhor custo-benefício, aumentando a eficácia mesmo ao reduzir a dimensionalidade.

Por sua vez, em Aguiar *et al.* (2021) foram investigadas diversas técnicas de classificação textual e combinações de *embeddings* extraídos de modelos desenvolvidos para o idioma português. Treinando esses modelos utilizando dados do Tribunal de Justiça do Ceará (TJCE), o modelo BERT obteve o melhor desempenho, alcançando um F1-score de 88%. Além disso, concluiu-se também que a representação dos documentos por meio de *embeddings* gerados pelo BERT, juntamente com a arquitetura de contextos bidirecionais, possibilita capturar o contexto específico do domínio jurídico.

Analisando técnicas de similaridade em jurisprudências do Superior Tribunal de Justiça, o trabalho em Gomes (2021) concluiu que os modelos tradicionais (TF-IDF e BM25) não apresentaram ganhos significativos em relação aos modelos semânticos baseados em predição (Word2Vec e BERT). No entanto, entre os modelos semânticos, o Word2Vec destacou-se estatisticamente em comparação com o BERT. Esse último, possivelmente prejudicado pela falta de uma base histórica de *Short Sentence Similarity* (STS) no domínio jurídico para ajuste do modelo.

Já o estudo em Noguti, Vellasques e Oliveira (2020) propôs uma técnica de PLN para classificar textos jurídicos do Ministério Público do Estado do Paraná em diferentes áreas do direito. Foram avaliados Modelos Lineares, *Boosted Trees* e Redes Neurais. Os melhores resultados foram alcançados com o uso do Word2Vec na Rede Neural Recorrente (RNN) LSTM. Além disso, foram investigadas diferentes abordagens na etapa de pré-processamento para representar os termos, revelando que técnicas semânticas mais simples alcançaram resultados semelhantes ou até melhores do que as abordagens mais complexas.

No estudo mencionado em Oliveira e Nascimento (2022) diversos modelos baseados em Transformers, como BERT, GPT-2 e RoBERTa, foram comparados, revelando um desempenho superior em relação às técnicas tradicionais de PLN. Destacou-se o modelo RoBERTa, que apresentou os melhores resultados devido à sua especialização para o idioma português.

Em Tosta (2022), ao constatar que acórdãos de diversos tribunais brasileiros (STF, STJ, TRF2, TJPB e TJMG) possuíam estruturas diferentes, foram utilizadas ferramentas de *Machine Learning* (ML) para segmentá-los automaticamente em cinco partes, buscando padronizá-los em uma mesma estrutura. Os resultados indicaram que os modelos treinados com dados segmentados alcançaram melhores resultados do que os modelos treinados especificamente para cada tribunal. Além disso, embora o BERT, que compreende apenas o texto, tenha apresentado bons resultados na segmentação dos documentos, a adição de características de layout e imagem melhorou significativamente a classificação dos segmentos.

Sendo assim, com base na revisão bibliográfica realizada, embora haja estudos que usam IA para auxiliar na busca de documentos similares em uma base do domínio jurídico, não foram encontrados trabalhos que adotam modelos de RAG para aprimorar

os resultados de sistemas tradicionais de busca, especialmente com documentos da Justiça Eleitoral e em português. Assim, este estudo se destaca como pioneiro neste assunto.

3 METODOLOGIA

Este capítulo descreve a metodologia utilizada neste trabalho, que se fundamenta em um *pipeline* de uma RAG para atingir o objetivo proposto. Em linhas gerais, a RAG é uma técnica que combina LLM com Recuperação de Informação. Antes de gerar um texto, a RAG primeiro recupera informações relevantes de um *dataset* usando uma consulta específica. Depois, essas informações recuperadas são então alimentadas em um LLM gerador de texto que utiliza tanto as informações recuperadas quanto o contexto da consulta para produzir um texto mais preciso como resposta (Lewis *et al.*, 2020).

Nas seções seguintes, é apresentada uma visão geral do *pipeline* proposto destacando cada uma de suas etapa. Em seguida, é apresentada uma visão geral dados utilizados, incluindo sua fonte e forma de coleta. Por fim, são especificadas as métricas para avaliar o *pipeline* e a qualidade dos resultados obtidos.

3.1 *Pipeline* da metodologia proposta

O *pipeline* proposto é ilustrado na Figura 2, sendo cada uma de suas etapas descritas a seguir.

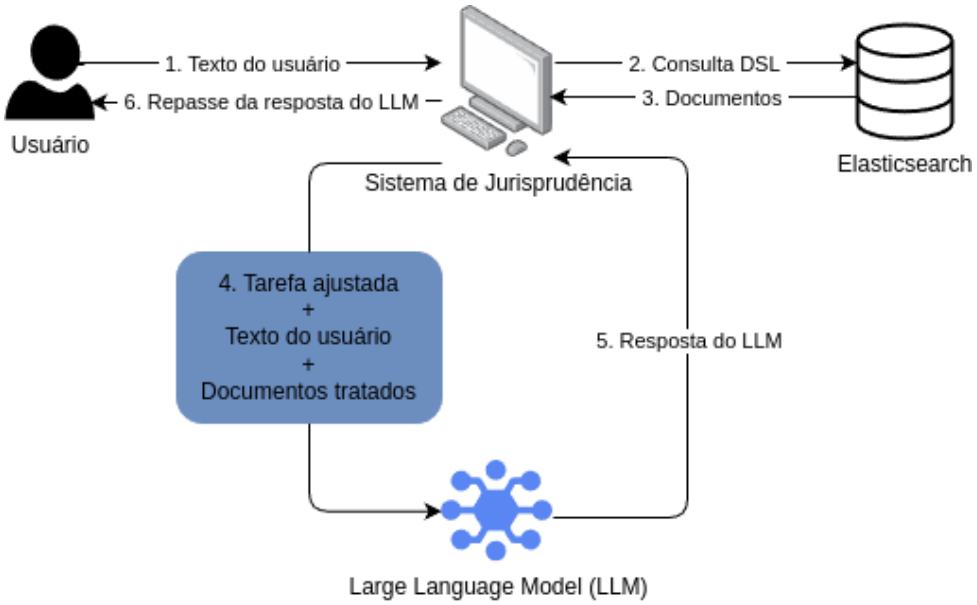

Figura 2 – *Pipeline* proposto.

3.1.1 Texto do usuário

O processo se inicia com o usuário inserindo seu texto de pesquisa no Sistema de Jurisprudência (https://jurisprudencia.tse.jus.br/). Esse texto pode ser uma pergunta ou palavras-chave relacionadas à jurisprudência.

Por exemplo:

texto_usuario = "Quais decisões são similares à decisão abaixo:

'Trata–se de novo pedido de propaganda partidária formulado pelo partido MOVIMENTO DEMOCRÁTICO BRASILEIRO – MDB, para veiculação de inserções no primeiro semestre de 2023, considerando que na publicação do deferimento, algumas datas já estavam prejudicadas. O partido foi intimado para indicar as inserções na forma do art. 14 da Resolução TSE n.23.679/2022 (ID 4508649). A agremiação apresentou as novas datas e inserções (ID 4508681). A Secretaria Judiciária certificou que os dias indicados pelo requerente foram reservados e atendem ao disposto na legislação (Lei n. 14.291/2022). Diante do exposto, DEFIRO o pedido de propaganda partidária gratuita formulado pelo partido MOVIMENTO DEMOCRÁTICO BRASILEIRO, para veiculação de inserções nas datas e horários mencionados na petição de ID 4508681. Intime–se. Datado e assinado eletronicamente. Juíza CAROLYNNE SOUZA DE MACÊDO OLIVEIRA Relatora'"

3.1.2 Consulta *Domain-Specific Language* (DSL)

O Sistema de Jurisprudência usa Elasticsearch para armazenar os dados. Ele é um banco de dados não relacional que armazena as informações em documentos no formato *JavaScript Object Notation* (JSON). Para realizar consultas, utiliza-se a DSL, uma linguagem específica de domínio que permite a formulação de consultas complexas e precisas. Os resultados das consultas são retornados e ordenados por relevância pelo próprio motor de busca do Elasticsearch. Também é possível especificar o número máximo de documentos a serem retornados utilizando o atributo "*size*".

Por exemplo:

```
{
  "size":20,
  "query": {
    "query_string": {
      "query": texto_usuario,
    }
  }
}
```

, onde *texto_usuario* é o texto contendo o que o usuário pretende buscar.

3.1.3 Documentos

Por meio da consulta DSL, o Elasticsearch retorna para o Sistema de Jurisprudência os documentos mais relevantes em formato JSON. Essa relevância é calculada por meio de

diversos fatores, como o TF-IDF e o BM25 (B.V., 2024).

Por exemplo:


```
[{
    "siglaUF": "AC",
    "publicacoes": [
        {
            "numeroPublicacao": "79",
            "nomeFontePublicacao": "Diário da Justiça Eletrônico",
            "numeroPagina": null,
            "siglaFontePublicacao": "DJE",
            "dataPublicacao": "08/05/2023",
            "numeroVolume": null
        }
    ],
    "numeroDecisao": null,
    "siglaClasse": "PropPart",
    "textoDecisao": "JUSTIÇA ELEITORAL TRIBUNAL REGIONAL ELEITORAL DO ACRE...",
    "descricaoTipoDecisao": "Decisão monocrática",
    "tipoDecisaoColegiado": "Sem anotação",
    "numeroUnicoFormatado": "0601566-30.2022.6.01.0000",
    "siglaTribunalJE": "TRE-AC",
    "referenciasLegislativas": null,
    "anoEleicao": 2022,
    "dataDecisao": "05/05/2023",
    "etiquetas": null,
    "textoEmenta": null,
    "codigoDecisao": 3247351,
    "descricaoClasse": "PROPAGANDA PARTIDÁRIA",
    "origemDecisao": "PJE",
    "naturezaDocumento": "Sem anotação",
    "assuntos": null,
    "nomeMunicipio": "RIO BRANCO"
}]
```


3.1.4 Tarefa ajustada

Neste momento, no lugar de já enviar as decisões retornadas pelo Elasticsearch ao usuário, é utilizado um LLM para tentar proporcionar uma resposta mais precisa ao usuário. Para isso, uma tarefa pré-definida é enviada ao LLM juntamente com os documentos retornados pelo Elasticsearch e o texto inicial fornecido pelo usuário.

Por exemplo:

```
"Sumarize em 3 linhas cada decisão em @DOCUMENTOS
e retorne apenas aquelas que respondem a pergunta:
+
'{texto_usuario}'
+
===
```

```
@DOCUMENTOS={documentos}",
```

onde *texto_usuario* é o texto fornecido pelo usuário na etapa 1 e *documentos* são os documentos retornados pelo Elasticsearch.

3.1.5 Resposta do LLM

A resposta do LLM é então repassada para o Sistema de Jurisprudência para que este possa repassar para o usuário final.

Por exemplo:

['- Decisão 3242215: Trata-se de novo pedido de propaganda partidária formulado pelo partido X, para veiculação de inserções em datas específicas. O partido foi intimado a indicar as inserções, apresentou as novas datas e a Secretaria Judiciária certificou que atendem à legislação. O pedido foi deferido, autorizando a veiculação das inserções nas datas e horários mencionados.',

3.1.6 Repasse da resposta do LLM

O Sistema de Jurisprudência repassa para o usuário final a resposta do LLM com base nos documentos retornados do Elasticsearch e uma tarefa específica pré-definida do LLM. Essa resposta deve ser mais precisa e relevante do que a consulta original do usuário, pois o LLM consegue extrair informações relevantes do texto capazes de melhorar a qualidade da resposta, como o contexto, as entidades e aspectos de escrita.

A etapa principal do *pipeline* da Figura 2 é a etapa 4, onde será feito um ajuste do LLM para uma tarefa particular. Por isso, ela é melhor detalhada na próxima seção.

3.2 Detalhamento da tarefa do LLM

O propósito desta etapa é explorar os benefícios do LLM para proporcionar uma resposta mais rica ao usuário final. Dessa forma, considerando que o LLM opera com base no paradigma de pergunta e resposta, é fundamental determinar qual é a pergunta mais adequada para se obter a melhor resposta.

Por isso, nesta etapa devem ser realizados experimentos com tarefas específicas para o LLM visando identificar qual delas se adapta melhor à complexidade e às nuances do domínio jurídico. Esses experimentos são cruciais para avaliar a capacidade do LLM em compreender e interpretar corretamente questões legais. Ao explorar diferentes tipos de perguntas e respostas, é possível identificar padrões, pontos fortes e fracos, bem como oportunidades de melhoria.

Para isso, o texto de entrada do LLM será composto por três componentes:

1. A tarefa refinada: tarefa esperada que o LLM execute;

2. O texto do usuário: texto de pesquisa inserido pelo usuário no Sistema de Jurisprudência;

3. Documentos: documentos retornados pelo Elasticsearch com base no texto de pesquisa do usuário.

Embora o Elasticsearch conte com um poderoso motor de busca baseado no Lucene [1] e possua recursos que, quando utilizados adequadamente, podem retornar resultados bastante satisfatórios para o usuário, ele não possui a capacidade de gerar texto, uma habilidade na qual os LLMs se destacam. Assim, a combinação do poder de recuperação de informação do Elasticsearch com o poder de geração de textos dos LLMs pode entregar resultados mais relevantes ao usuário.

Porém, em um grande volume de dados, a necessidade de eficiência torna inviável para o LLM processar todos os documentos retornados pelo Elasticsearch. Assim, é preciso estabelecer um equilíbrio entre o volume de documentos e o tempo que o LLM precisaria para processá-los e gerar uma reposta relevante em tempo hábil.

Além disso, será crucial realizar uma análise detalhada e um pré-processamento adequado dos documentos antes de submetê-los ao LLM. Durante essa análise, é essencial identificar os atributos mais relevantes para a tarefa em questão, bem como os seus tamanhos. Esse trabalho não apenas otimiza a capacidade do LLM de interpretar as informações corretamente, mas também contribui para a eficiência da RAG.

Ademais, para esta etapa do *pipeline* também será escolhido o LLM que fará parte da RAG. É fundamental que o modelo escolhido tenha bom desempenho para gerar textos em português e, de preferência, consiga lidar com textos longos.

Vale destacar que, dentro da RAG, o *in-context learning* é fundamental. Isso significa que o sistema utiliza exemplos específicos dos documentos recuperados para adaptar as respostas ao contexto jurídico. Essa abordagem garante que as respostas sejam relevantes e estejam alinhadas com as nuances dos casos legais e das leis aplicáveis. Além disso, a avaliação de diferentes *prompts* e a integração das respostas com os textos originais recuperados pelo Elasticsearch afetam o resultado, sendo um objeto de estudo dentro desta metodologia.

3.3 Métricas de avaliação experimental

Serão adotadas as métrica Entropia e *Bacronymic Language model Approach for summary quality estimatioN* (BLANC) nos experimentos realizados desta pesquisa. A

[1] https://lucene.apache.org/

primeira avalia a ordenação dos documentos similares mais relevantes, já a segunda, avalia a qualidade dos resumos gerados.

3.3.1 Entropia

A Entropia foi proposta inicialmente por Shannon (1948). Ela foi definida como uma medida da quantidade de informação contida em uma mensagem, ou seja, quanto maior a entropia, maior a quantidade de informação e menor a previsibilidade dos símbolos que compõem a mensagem.

No contexto de recuperação de informação, Entropia é uma medida estatística que quantifica a incerteza ou imprevisibilidade associada a um conjunto de dados (Manning; Raghavan; Schütze, 2008). A entropia pode ser usada para avaliar a diversidade dos documentos retornados por um sistema de busca. Por exemplo, ela pode quantificar a incerteza ou a imprevisibilidade das classes de documentos recuperados. Sua fórmula é (Manning; Raghavan; Schütze, 2008):

$$H(P) = -\sum_{x \in X} P(x) \log_2 P(x),$$

onde:

- $H(P)$ é a entropia do conjunto de documentos P;
- X é o conjunto de todos os possíveis tipos de documentos;
- $P(x)$ é a probabilidade de um documento x ser de um determinado tipo;
- $\log_2$ é o logaritmo na base 2.

Assim, se todos os documentos retornados forem do mesmo tipo, a entropia será 0, indicando que não há diversidade nos tipos de documentos. Por outro lado, se os documentos retornados forem de diferentes tipos com probabilidades iguais, a entropia será máxima, indicando alta diversidade e incerteza sobre o tipo de documento que será retornado.

3.3.2 BLANC

A BLANC, proposta por Vasilyev, Dharnidharka e Bohannon (2020), é uma abordagem objetiva para avaliar a qualidade de resumos textuais gerados automaticamente. Baseia-se na aplicação do modelo BERT pré-treinado para prever termos ocultos no texto original. A precisão com que o BERT preenche essas lacunas é comparada à precisão obtida quando o resumo é usado como suporte para compreender o texto original. Quanto maior a precisão alcançada com o auxílio do resumo, mais eficaz ele é em refletir o entendimento

do texto original, resultando em um *score* maior. A métrica BLANC é definida como (Vasilyev *et al.*, 2020):

$$BLANC = \frac{N_{help} - N_{base}}{N_{total}},$$

onde:

- N_{help} é o número de *tokens* (palavras ou conjunto de palavras) ocultos descobertos pelo modelo com a ajuda do resumo;
- N_{base} é o número de *tokens* ocultos descobertos pelo modelo sem a ajuda do resumo;
- N_{total} é o número total de *tokens* ocultos.

Dessa forma, um valor mais alto indica que o resumo é mais adequado, enquanto um valor de 0 representa o menor nível de adequação. Um *score* muito baixo pode sugerir que o resumo não é claro o suficiente para que o BERT preencha as lacunas corretamente, enquanto um *score* mais alto indica que o resumo captura melhor as nuances e ideias do texto original. Vale ressaltar que, por não ser uma métrica de similaridade, é possível que um resumo com baixa similaridade em relação ao texto original ainda possa receber um *score* alto de qualidade por ainda assim conseguir transmitir as ideias do texto original com outras palavras.

Uma grande vantagem dessa abordagem em relação a outras mais populares, como a ROUGE (Lin, 2004), é que ela não exige um resumo de referência produzido por humanos para avaliar a qualidade do resumo gerado automaticamente. É uma abordagem totalmente automatizada, sem necessidade de intervenção humana. Isso a torna mais fácil de ser reproduzida, pois a necessidade de interação humana geralmente resulta em processos mais lentos, caros e com recursos limitados.

4 AVALIAÇÃO EXPERIMENTAL

Neste capítulo é apresentada uma análise detalhada dos experimentos realizados com base na metodologia descrita anteriormente. O desempenho do *pipeline* da RAG proposta será testado em cinco conjuntos de dados, cada um representando um tribunal da JE específico, assim será possível atender aos dois objetivos específicos deste estudo, quais são:

1. Avaliar se a implementação de uma RAG pode melhorar a qualidade das respostas de um sistema de recuperação tradicional quando utilizado para identificar atos judiciais similares;

2. Avaliar se o desempenho de um modelo de LLM para identificar similaridades e melhorar a apresentação dos resultados das buscas de textos jurídicos em português é significativamente impactado quando se altera o tribunal (TSE ou TREs) de origem desses textos.

4.1 Configuração experimental

Para conduzir os experimentos, foi desenvolvida uma aplicação[1] na linguagem Python para automatizar e gerenciar o processo experimental com maior eficiência e precisão. Utilizando bibliotecas especializadas, a aplicação não apenas facilita a execução de experimentos repetitivos, mas também garante exatidão na análise de dados.

O ambiente computacional no qual a aplicação desenvolvida foi executada possui as seguintes especificações de *hardware* e *software*:

1. *Hardware* (Desktop Dell Inc. XPS 8500):
   a) Memória de 12GB
   b) Intel® Core™ i7-3770 × 8
   c) Disco rígido de 3TB
   d) Placa de vídeo NVIDIA GeForce GT 640
2. *Software*
   a) Ubuntu 24.04 LTS
   b) Elasticsearch 8.14.2
   c) Kibana 8.14.2

[1] https://github.com/ramonrodrigues/tcc-mba-usp

d) Docker 27.1.1

e) Docker Desktop 4.32.0

f) Visual Studio Code 1.92.1

g) Python 3.12.3

Sendo forma, a aplicação desenvolvida e o ambiente computacional constituem o conjunto necessário para automatizar todas as etapas exigidas para as execuções dos experimentos deste trabalho.

4.1.1 *Dataset*

Os dados utilizados para os experimentos são atos judiciais da JE, compreendendo tanto o Tribunal Superior Eleitoral (TSE), quando os Tribunais Regionais Eleitorais (TREs). Eles foram obtidos via Lei de Acesso à Informação, Lei nº 12.527/2011, em 02/02/2024, por meio de uma solicitação ao TSE, órgão que centraliza os atos judiciais de todos os tribunais da JE.

Ao todo, são 1.168.557 atos judiciais em formato JSON, distribuídos entre Acórdãos, Decisões sem resolução, Decisões monocráticas e Resoluções, cada um com os seguintes atributos:

- "codigoDecisao": identificador do ato judicial;
- "siglaClasse": sigla da classe processual;
- "textoDecisao": o texto do ato judicial;
- "siglaUF": sigla da Unidade Federativa do ato judicial;
- "descricaoClasse": nome da classe processual;
- "dataDecisao": a data do ato judicial;
- "nomeMunicipio": nome do município do ato judicial;
- "publicacoes": detalhes das publicações do ato judicial, incluindo o número da publicação, a fonte da publicação, a sigla da fonte de publicação, a data de publicação, o volume e o número da página;
- "numeroDecisao": número da decisão, quando possuir;
- "descricaoTipoDecisao": o tipo do ato judicial;
- "tipoDecisaoColegiado": o tipo do ato judicial quando este for colegiado;
- "numeroUnicoFormatado": o número único do processo;

- "siglaTribunalJE": a sigla do tribunal da Justiça Eleitoral;
- "referenciasLegislativas": as referências legislativas do ato judicial;
- "anoEleicao": o ano de eleição ao qual se refere o ato judicial;
- "etiquetas": marcadores anotados pela área negocial;
- "textoEmenta": o texto da ementa da decisão;
- "origemDecisao": sistema de origem do ato judicial;
- "naturezaDocumento": natureza do documento;
- "assuntos": lista de assuntos do processo.

Conforme Figura 3, mais da metade, 63,87% (746.370) são Acórdãos, seguidos de Decisões monocráticas, 29,15% (340.685), sendo as classes judicais Recurso Eleitoral (RE, 227.768), Prestação de Contas (PC, 76.958), Registro de Candidatura (RCAND, 46.178) e Recurso Especial Eleitoral (RESPE, 33.732) as mais predominantes.

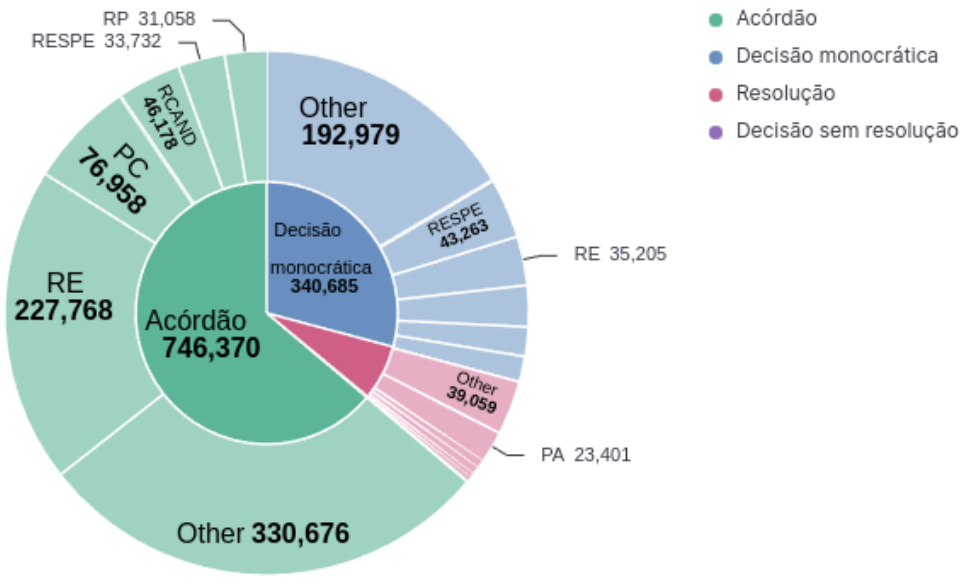

Figura 3 – Quantidade de classes judiciais por tipo de documento.

Já pela Figura 4 é possível perceber que o TSE é o tribunal da JE que mais possui atos judiciais publicados (205.419), seguido do TRE-SP com praticamente metade da quantidade de atos do TSE (121.736). Também é notável pela figura que o tipo de ato judicial mais predominante nos tribunais é o Acórdão, com exceção do TSE, onde a maioria dos atos, 56,94% (116.986), é formada por Decisões monocráticas.

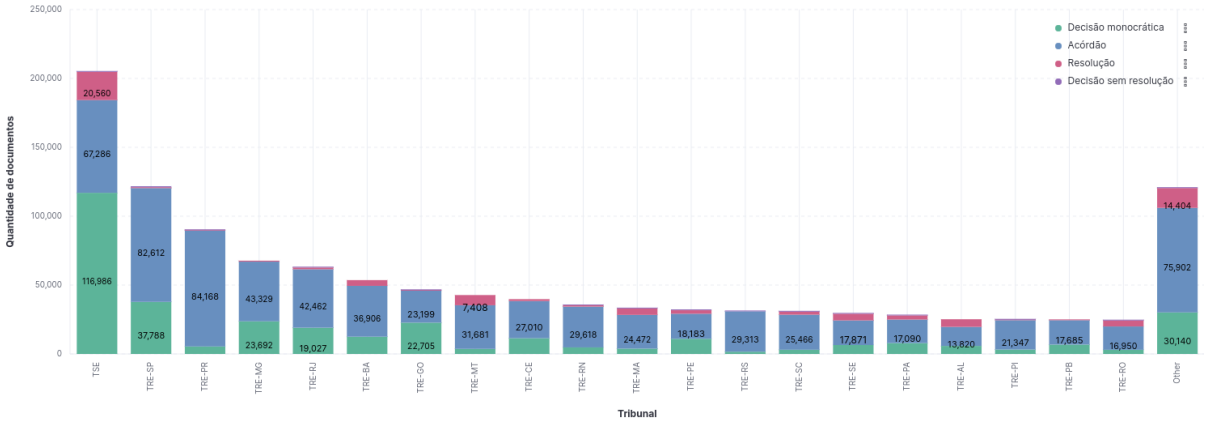

Figura 4 – Quantidade de documentos por tribunal.

Já em relação à distribuição das classes judiciais, a Figura 5 mostra não haver um padrão entre os tribunais da JE. Há classes que só são utilizadas por alguns, como a classe Recurso (REC), que é utilizada apenas pelo TRE-SP. Além disso, é possível perceber pela nomenclatura de algumas classes que alguns tribunais carecem de saneamento. Por exemplo, o TRE-PR possui classes cujas siglas são "2"e "N/A", ou seja, sem nenhuma significância jurídica e em desconformidade com a padronização de classes judiciais definida pelo Conselho Nacional de Justiça (CNJ)[2]. O TSE é o tribunal que possui mais classes em conformidade com o CNJ.

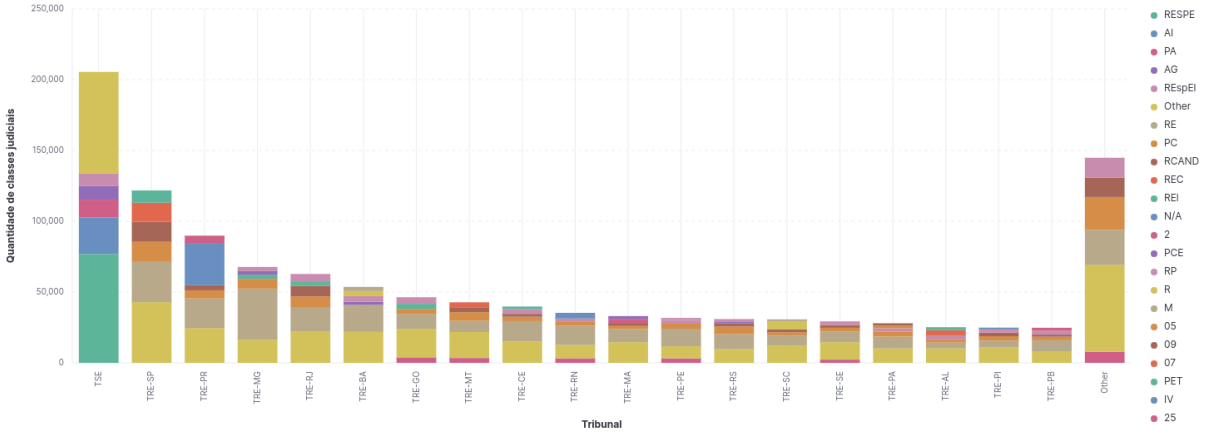

Figura 5 – Quantidade de classes judiciais por tribunal.

É importante destacar que todos esses dados são públicos e estão disponíveis para consulta no sistema de Pesquisa de Jurisprudência do TSE, acessível pelo endereço https://jurisprudencia.tse.jus.br/; e no sistema de Pesquisa de Jurisprudência dos TREs,

[2] https://www.cnj.jus.br/sgt/consulta_publica_classes.php

acessível pelo endereço https://jurisprudencia-tres.tse.jus.br/; ambos implementados e disponibilizados pelo TSE.

4.1.1.1 Indexação

Para armazenar os dados coletados, foi adotado o Elasticsearch como base de dados. Para isso, o Elasticsearch foi implantando em um contêiner Docker e foi utilizado o Docker Desktop como ferramenta para simplificar o seu gerenciamento. Além de possuir um motor de busca poderoso e amplamente conhecido por seu desempenho, o Elasticsearch trabalha com dados em formato JSON, o que facilita a indexação dos dados coletados, já que estão nesse mesmo formato.

Antes da indexação, os dados foram pré-processados para torná-los mais limpos e, consequentemente, mais fáceis de serem interpretados por máquina. Inicialmente, foram removidas as *tags* HTML, pois poderiam prejudicar o entendimento semântico dos dados. Posteriormente, foram eliminados espaços vazios gerados por quebras de linha (\n), tabulações (\t) e *carriage return* (\r) excessivos, resíduos de formatações anteriores e que não agregam valor à análise.

4.1.1.2 Consulta

Após a indexação dos documentos, foi criada a consulta DSL abaixo, que será detalhada a seguir, para que os dados fossem buscados do Elasticsearch:

```
{
  "size": 20,
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "textoDecisao": textoDecisao
          }
        },
        {
          "term": {
            "descricaoTipoDecisao.keyword": "Decisão monocrática"
          }
        },
        {
          "term": {
            "siglaTribunalJE.keyword": tribunal
          }
        }
      ]
    }
  },
  "_source": ["siglaTribunalJE", "codigoDecisao", "siglaClasse", "textoDecisao"]
}
```

A restrição "*size*" : 20 foi adicionada para limitar a resposta do Elasticsearch a apenas 20 documentos. Isso visa fornecer à RAG um conjunto de 20 atos judiciais, o que é suficiente para que ela consiga selecionar os mais relevantes solicitados para o usuário.

A cláusula *match* foi a escolhida para a busca porque ela permite realizar consultas simples utilizando o analisador padrão do Elasticsearch. Essa cláusula avalia a relevância dos termos encontrados nos documentos, favorecendo aqueles que contêm termos correspondentes ao texto da busca, neste caso, ao texto da variável *textoDecisao*.

Embora a cláusula *query_string* seja uma alternativa válida, sua capacidade de realizar consultas complexas com operadores lógicos e caracteres especiais pode causar equívocos de sintaxe quando o texto puro da decisão é informado na variável *textoDecisao*. Por exemplo, se o ato judicial contiver aspas duplas ou parênteses, esses caracteres podem ser interpretados pela *query_string* como operadores da expressão de busca, dificultando a busca exata pelo conteúdo desejado ou até mesmo gerando erros de sintaxe na consulta. Portanto, para evitar esses empecilhos e garantir uma busca mais direta e eficiente, foi adotada a cláusula *match*.

Já a restrição "*descricaoTipoDecisao.keyword*" : "*Decisãomonocrática*" foi adotada após identificar que apenas esse tipo de decisão possui conteúdo textual adequado para este estudo, podendo ter um tamanho acima 1024 *tokens*. Nos demais tipos, muitos não apresentam texto da decisão, e aqueles que o fazem, geralmente, contêm textos curtos e insuficientes para análise por um modelo de linguagem natural. Por essa razão, esses outros tipos foram excluídos da análise, seguem alguns de seus exemplos:

- Acórdão:
  - "Por maioria, indeferir o pedido."
  - "Por unanimidade, não conhecer do recurso."
  - "RecursoHomologaçãoDesistência"
- Resolução:
  - "À unanimidade de votos, a Corte aprovou a Resolução, nos termos do voto do Relator."
  - "O Tribunal, por unanimidade, autorizou o encaminhamento da lista.Presidência do Ministro Sepúlveda Pertence. Presentes os Ministros Celso de Mello, Carlos Velloso, Pedro Acioli, Américo Luz, Vilas Boas, Hugo Gueiros e o Dr. Geraldo Brindeiro, Vice-Procurador-Geral Eleitoral. Indexação: Lista tríplice. Preenchimento. Juiz titular. Jurista. Encaminhamento. Executivo."
  - "Prestação de contasDocumentoAplicaçãoDecreto legislativo"
- Decisão sem resolução:

  - "O Tribunal, por unanimidade, não conheceu do pedido, nos termos do voto do Relator."
  - "O Tribunal, à unanimidade, deferiu a renovação de requisição, de 9/6/10 a 8/6/11, nos termos do voto do Relator."
  - "O Tribunal, por unanimidade, deferiu, em parte, a proposta, nos termos do voto do Relator."

Por fim, foram restringidos na consulta apenas os seguintes atributos para retorno, com os demais sendo descartados por não serem relevantes para este estudo:

1. "siglaTribunalJE": sigla do tribunal da Justiça Eleitoral;
2. "codigoDecisao": identificador da decisão;
3. "siglaClasse": sigla da classe judicial da decisão. É utilizado para calcular a entropia;
4. "textoDecisao": o próprio texto da decisão.

A consulta DSL descrita nesta sessão foi criada para que a RAG possa retornar os 20 documentos da base do Elasticsearch mais similares à decisão informada pelo usuário, neste caso, a variável *textoDecisao*. Porém, para a execução dos cinco experimentos, é necessário definir qual tribunal representará cada experimento e qual decisão deste tribunal será utilizada para preencher a variável *textoDecisao*. Assim, os critérios definidos foram:

- Tribunais: os cinco com o maior número de decisões monocráticas: TSE (116.986 monocráticas), TRE-SP (37.788 monocráticas), TRE-MG (23.692 monocráticas), TRE-GO (22.705 monocráticas) e TRE-RJ (19.027 monocráticas);
- Decisão de cada tribunal: a mais recente considerando o atributo "dataDecisao".

4.1.2 LLM

O LLM escolhido para esse estudo foi o modelo Llama 3.1 405B Instruct[3]. Lançado recentemente, julho de 2024, ele é o modelo gerador de texto mais avançado da família Llama 3.1, que inclui modelos menores, com 8 e 70 bilhões de parâmetros. Porém, o com 405 bilhões, segundo descrito por Dubey *et al.* (2024), possui desempenho superior aos demais da mesma família, especialmente em tarefas que envolvem o idioma português.

Entre os fatores principais que motivaram a escolha desse modelo está o fato de ele ser *open source*, o que a torna acessível. Mesmo sendo gratuito, seu desempenho é comparável ao de modelos pagos e de referência no estado da arte, como o (OpenAI *et al.*,

[3] https://huggingface.co/meta-llama/Meta-Llama-3.1-405B-Instruct

2024). Além disso, ele oferece suporte a múltiplos idiomas, incluindo o português, o que é essencial para este trabalho, que utiliza documentos escritos nesse idioma como base de dados. Outro aspecto relevante é sua capacidade de processar textos longos, suportando cotextos de até 128.000 tokens, o que contribui para a geração de resumos de documentos extensos, como algumas decisões monocráticas.

Apesar de ser gratuito, a utilização do Llama 3.1 405B Instruct exige uma infraestrutura robusta para o seu processamento. Com bilhões de parâmetros, o treinamento desse modelo demandou mais de 39 milhões de horas de processamento (AI, 2024), o que torna inviável sua utilização em um computador convencional. Felizmente, a Nvidia, por meio do *Developer Program Members* (Corporation, 2024), disponibiliza sua infraestrutura para que esse e outros modelos possam ser utilizados via API para fins acadêmicos, o que viabilizou os experimentos deste trabalho.

4.1.2.1 Tarefas

Para a construção da RAG, é fundamental definir as tarefas adequadamente que serão repassadas para o LLM. Portanto, durante essa definição, foram testados vários casos buscando ter a resposta mais adequada do LLM. Entre os exemplos testados, estão:

- "*Sumarize em 3 linhas cada decisão em @DOCUMENTOS e retorne apenas as 5 mais relevantes que respondem a seguinte pergunta:*

  Quais decisões são similares à decisão abaixo:

  [texo_decisao]

  ===

  @DOCUMENTOS=[documentos]"

- "*Sumarize em até 3 linhas cada decisão em @DOCUMENTOS e retorne apenas as 5 mais relevantes que respondem a seguinte pergunta:*

  Quais decisões são similares à decisão abaixo:

  [texo_decisao]

  ===

  @DOCUMENTOS=[documentos]"

- "*Com base nas decisões em @DOCUMENTOS, responda à pergunta abaixo, resumindo cada decisão da resposta em até 3 linhas.*

  Pergunta: quais são as 5 decisões mais similares à decisão abaixo?

  [texo_decisao]

  ===

@DOCUMENTOS=[documentos]"

Porém, observou-se o LLM não gerava bons resumos em alguns casos, com textos truncados ou difíceis de serem interpretados. A causa disso foi que estavam sendo repassadas ao LLM não apenas uma tarefa, mais duas, a primeira que seria responder à pergunta do usuário, e a segunda, que seria para resumir cada documento da resposta da pergunta. Sendo assim, após vários testes, concluiu-se que seria mais vantajoso informar ao LLM duas tarefas, uma baseada na outra, conforma abaixo:

1. "*Com base nos documentos em @DOCUMENTOS, responda à pergunta abaixo. Na resposta, retorne apenas um array com o 'codigoDecisao' dos documentos retornados.*

   Pergunta: Retorne as 5 decisões mais similares à decisão abaixo:

   [texo_decisao]

   ===

   @DOCUMENTOS=[documentos]"

2. Para cada código de documento retornado no *array* da tarefa anterior, busca-se o texto daquela decisão e instrui a tarefa:

   "*Resuma em um parágrafo de 5 linhas o texto abaixo. Comece com 'RESUMO:':*

   [texto]".

   Após isso, retira-se o 'RESUMO:' da resposta, restando apenas o próprio resumo da decisão.

Com essas duas tarefas, foi possível, primeiramente, identificar as decisões mais similares com base na interpretação textual do LLM. Em seguida, cada decisão selecionada foi resumida de forma a permitir ao usuário uma compreensão rápida e clara do seu conteúdo.

A divisão dessas tarefas também simplificou a execução dos experimentos. Se houvesse apenas uma tarefa combinando ordenação e resumo das decisões, seria necessário a implementação de um reconhecimento de padrão textual para extrair automaticamente cada resumo da resposta e avaliar sua qualidade. Isso não seria o ideal, pois o LLM tende a gerar respostas com padrões textuais variados, especialmente em respostas longas.

Por outro lado, a primeira tarefa, por ser simples e curta, garantiu a obtenção de um *array* no formato correto. Assim, não foi necessário identificar qualquer padrão no texto, apenas converter a resposta do LLM de *string* para lista (*array*). Isso foi suficiente para que se pudesse percorrer a lista e buscar o texto de cada decisão pelo seu código. Com isso, foi possível analisar a qualidade do resumo gerado pela segunda tarefa, sem necessidade de tratamento adicional nas respostas do LLM.

4.2 Métricas

As métricas utilizadas nos experimentos deste trabalho foram a Entropia e o BLANC. Os cálculos foram realizados em cada experimento com o auxílio dos pacotes *scipy.stats* e *blanc.BlancHelp* do Python.

A entropia foi calculada em dois momentos distintos. No primeiro, para avaliar o grau de desordem dos 20 resultados retornados pelo Elasticsearch considerando a classe judicial de cada documento especificada pelo atributo "siglaClasse". Um resultado que retorna muitos documentos com a mesma classe indica um baixo grau de desordem, sugerindo que as decisões são mais propensas a serem similares entre si. Por outro lado, muitos documentos com classes distintas indicam alta desordem, levando a crer que, apesar de serem retornados na consulta por terem termos similares, são documentos que tratam de temas distintos.

No segundo momento, a entropia foi calculada com base nas classes judiciais dos cinco documentos retornados pelo LLM visando determinar se o modelo consegue reorganizar os 20 documentos retornados pelo Elasticsearch de uma maneira que responsa melhor a pergunta do usuário. Após isso, para fins de comparação com o LLM, também foi calculada a entropia das cinco primeiras decisões retornadas pelo Elasticsearch.

Já para analisar a qualidade dos resumos gerados pelo LLM utilizando a métrica BLANC, foi criada uma planilha contendo o texto original da decisão e o texto resumido pelo LLM para cada uma das cinco decisões retornadas. Em seguida, uma função em Python leu a planilha e, linha por linha, passou o texto original e o texto resumido como parâmetros para a função de cálculo do *score* BLANC. Os resultados foram então armazenados em uma cópia da planilha original.

4.3 Resultados e discussão

Com base nas ferramentas e configurações descritas nas seções anteriores, foram realizados os cinco experimentos para avaliar a RAG proposta neste trabalho, cada um focando em um dos tribunais. Em cada experimento, os resultados de uma simples busca no Elasticsearch foram comparados com os resultados fornecidos pela RAG. A análise considerou tanto a entropia quanto a qualidade dos resumos gerados pelo LLM da RAG.

4.3.1 Entropias

Os resultados das entropias podem ser visualizados na Tabela 1, onde é possível perceber que, quando analisados os 20 documentos retornados pelo Elasticsearch, o tribunal que exibiu a maior entropia foi o TSE, com 1,59; enquanto o TRE-RJ apresentou a menor entropia, zero. Observa-se também um indicativo de correlação entre a entropia e a

quantidade de dados analisados, pois os tribunais com um maior volume de decisões monocráticas apresentaram maior entropia, conforme ordenação da tabela.

			Elasticsearch		**LLM**
#	**Tribunal**	**Total de documentos**	**Entropia dos 20 documentos**	**Entropia dos 5 primeiros documentos**	**Entropia dos 5 documentos**
1	TSE	116.986	1,59	0	0
2	TRE-SP	37.788	0,97	0	0
3	TRE-MG	23.692	0,46	0	0
4	TRE-GO	22.705	0,72	0,97	0
5	TRE-RJ	19.027	0	0	0

Tabela 1 – Comparação da entropia do resultado do Elasticsearch com a entropia do resultado do LLM em cada um dos experimentos.

Quando analisadas as entropias dos cinco primeiros documentos do Elasticsearh, Tabela 1, apenas o experimento 4 (TRE-GO) apresentou entropia acima de zero, 0,97. Isso indica que o Elasticsearch consegue ter um bom ranqueamento dos documentos mais relevantes. Na Tabela 2 é possível ver quais foram as classes dos cinco primeiros documentos retornados pelo Elasticsearch no experimento 4. Nos demais experimentos, os cinco primeiros documentos sempre foram da mesma classe.

#	**Tribunal**	**Classe judicial**	**Classes judiciais dos 20 documentos retornados pelo Elasticsearch**	**Classes judiciais dos 5 documentos retornados pelo LLM**
1	TSE	PetCiv	PetCiv, PetCiv, PetCiv, PetCiv, PetCiv, PetCiv, PetCiv, PetCiv, RESPE, MSCiv, RESPE, RESPE, RESPE, MS, RESPE, RESPE, RESPE, RESPE, MS, PetCiv	PetCiv, PetCiv, PetCiv, PetCiv, PetCiv
2	TRE-SP	AJDesCargEle	AJDesCargEle, AJDesCargEle, AJDesCargEle, AJDesCargEle, AJDesCargEle, AJDesCargEle, AJDesCargEle, AJDesCargEle, PET, PET, PET, PET, PET, PET, PET, PET, PET, PET	PET, PET, PET, PET, PET
3	TRE-MG	PCE	PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE, ED, PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE, ED, PCE	PCE, PCE, PCE, PCE, PCE
4	TRE-GO	PCE	PCE, REl, REl, PCE, REl, PCE, REl, REl, PCE, REl, REl, REl, REl, REl, REl, REl, REl, REl, REl	REl, REl, REl, REl, REl
5	TRE-RJ	PCE	PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE, PCE	PCE, PCE, PCE, PCE, PCE

Tabela 2 – Comparação das classes judiciais dos documentos retornados pelo Elasticsearch e pelo LLM.

Em relação à entropia do LLM, todos os experimentos mostraram uma entropia zero para os retornos do LLM, o que indica que os resultados do modelo estão bem organizados em 100% dos casos, superando os resultados do Elasticsearch neste aspecto.

Porém, ao analisar o experimento com o TRE-GO, único caso onde o LLM apresentou uma entropia melhor do que o Elasticsearch, é possível perceber pela Tabela 2, que as decisões retornadas pelo LLM, apesar de serem de uma mesma classe ("REl"), é de uma classe diferente da classe da decisão ("PCE") informada na pesquisa do experimento.

Outro ponto relevante é que, conforme Tabela 3, o Elasticsearch sempre ranqueou a própria decisão informada na busca como a primeira no *ranking* de resultados, em todos os experimentos. Isso evidencia a eficácia do Elasticsearch em encontrar documentos específicos. Em contraste, o LLM trouxe a decisão alvo como primeira do *ranking* apenas no Experimento 1 (TSE), 3 (TRE-MG) e 5 (TRE-RJ). Isso pode indicar que o LLM tentou não buscar a mesma decisão informada como parâmetro, mas sim as suas similares, conforme solicitado a ela pela tarefa.

#	Tribunal	Código do documento	Códigos dos 20 documentos retornados pelo Elasticsearch	Códigos dos 5 documentos retornados pelo LLM
1	TSE	3295663	3295663, 3265859, 3265891, 3260564, 3253422, 3253377, 3257475, 3265836, 486684, 3249448, 494421, 483120, 495648, 511941, 497781, 493031, 487896, 497022, 520663, 2754291	3295663, 3265859, 3265891, 3260564, 3253422
2	TRE-SP	3297358	3297358, 3281738, 2414579, 1401002, 2413043, 1401001, 3227534, 2994094, 1165233, 1165604, 1165659, 1148774, 1148414, 1148398, 1148396, 1165386, 1148245, 1148479, 1148612, 1148266	1165233, 1165604, 1165659, 1148774, 1148414
3	TRE-MG	3297577	3297577, 3295892, 3297616, 3288043, 3295928, 3295896, 3278367, 3255267, 3225159, 3297580, 3281817, 3293431, 3281807, 3275973, 3296995, 3296993, 3260701, 3289249, 3225101, 3243299	3297577, 3295892, 3297616, 3288043, 3295928
4	TRE-GO	3297264	3297264, 1423256, 2414608, 3236382, 1422301, 3233711, 1448187, 3275270, 3234892, 1448183, 1423615, 3275274, 2976353, 1419869, 3263799, 2078616, 2640078, 2640079, 1420105, 1420026	2414608, 1422301, 1448187, 3275270, 1448183
5	TRE-RJ	3297002	3297002, 3295571, 3263868, 3285337, 3288697, 3293135, 3297018, 3296793, 3293838, 3292409, 3292418, 3297019, 3285318, 3290418, 3288698, 3297329, 3295568, 3294480, 3297417, 3249800	3297002, 3295571, 3263868, 3285337, 3288697

Tabela 3 – Comparação dos códigos dos documentos retornados pelo Elasticsearch e pelo LLM

Sendo assim, é possível concluir que ambos, Elasticsearch e LLM, apresentaram boas entropias em seus resultados. A diferença é que o LLM tenta buscar as decisões similares, conforme preconiza a tarefa, enquanto o Elasticsearch, busca as decisões mais próximas ao texto informado no experimento. Por isso, ele sempre retorna a própria decisão do experimento como a primeira do *ranking* de resultados, já o LLM, tenta eliminá-la.

4.3.2 BLANC

Uma vantagem significativa em se utilizar um LLM para aprimorar os resultados de uma busca no Elasticsearch é sua capacidade de gerar resumos, algo que o Elasticsearch, sendo apenas um motor de busca, não é capaz de realizar. Assim, diferentemente do Elasticsearch, que se limita à recuperação de documentos, o LLM oferece o valor agregado de gerar textos com informações sintetizadas.

Conforme mostrado na Tabela 4, as pontuações de qualidade dos 25 resumos (5 decisões de cada experimento) gerados pelo LLM, avaliados pela métrica BLANC, variaram de 0,045 a 0,267, com uma média de 0,138. De acordo com Vasilyev *et al.* (2020), modelos

geradores de resumos geralmente obtêm pontuações BLANC entre 0,05 e 0,20, o que indica que os resumos ajudam a descobrir de 5% a 20% dos *tokens* ocultos do texto original. Dessa forma, uma média de 13,8% pode ser considerada um resultado satisfatório, posicionando o modelo dentro do intervalo esperado para a qualidade de resumos gerados por LLM.

Além disso, a Tabela 4 também revela que resumos de documentos mais longos tendem a ter pontuações BLANC menores. O resumo com a menor pontuação, 0,045, foi gerado a partir do documento mais extenso, contendo 4.838 palavras, enquanto o resumo com a maior pontuação, 0,267, foi gerado a partir do documento mais curto, com 445 palavras. Isso indica que textos mais longos dificultam a geração de resumos de qualidade pelo LLM.

Observa-se ainda na Tabela 4 que as decisões mais longas, e, consequentemente, com menores pontuações BLANC, são do TSE e do TRE-GO. Em contraste, a maioria das decisões do TRE-RJ, as quais são mais curtas, apresentam as maiores pontuações.

#	**Tribunal**	**Código do documento**	***Tokens***	**BLANC**
2	TRE-SP	1148774	445	0,267
5	TRE-RJ	3295571	579	0,215
2	TRE-SP	1165604	605	0,207
5	TRE-RJ	3263868	554	0,206
5	TRE-RJ	3285337	491	0,190
5	TRE-RJ	3288697	483	0,188
3	TRE-MG	3297616	640	0,173
1	TSE	3265891	1042	0,158
5	TRE-RJ	3297002	701	0,154
3	TRE-MG	3295928	684	0,151
2	TRE-SP	1148414	667	0,147
3	TRE-MG	3297577	577	0,135
2	TRE-SP	1165233	754	0,125
3	TRE-MG	3295892	1588	0,122
1	TSE	3265859	1298	0,122
4	TRE-GO	1422301	2092	0,115
2	TRE-SP	1165659	824	0,113
3	TRE-MG	3288043	1110	0,108
4	TRE-GO	1448187	2059	0,102
1	TSE	3253422	2236	0,097
4	TRE-GO	2414608	2072	0,082
4	TRE-GO	3275270	3025	0,080
4	TRE-GO	1448183	2059	0,080
1	TSE	3260564	1689	0,078
1	TSE	3295663	4838	0,045

Tabela 4 – Pontuação BLANC dos resumos gerados pelo LLM.

No entanto, ao analisar o resumo de menor qualidade, nota-se que ele é compreensível, informativo, conciso e possui boa fluidez, características que, segundo Vasilyev *et al.* (2020) são necessárias para um bom resumo. Sua qualidade pode ser analisada abaixo comparando a escrita do texto resumido com a do texto original.

- Texto resumido

  - *"O Tribunal Superior Eleitoral (TSE) determinou a* ***intervenção no Partido Renovador Trabalhista Brasileiro*** *(PRTB) devido a* ***disputas internas e irregularidades em convenções partidárias****. O TSE nomeou um interventor para conduzir* ***novas eleições no partido até 29 de fevereiro de 2024****. A decisão foi tomada após a constatação de que as* ***eleições anteriores foram realizadas de forma irregular*** *e que os principais representantes do partido estavam impedidos de conduzir novo processo eleitoral."*

- Texto original com partes suprimidas para uma visão geral mais direta:

  - *"[...]Trata–se inicialmente de ações propostas por membros do* ***Partido Renovador Trabalhista Brasileiro*** *que buscavam, em suma, discutir a gestão e presidência do partido, em virtude de* ***disputas internas*** *[...]. Diante disso, a)* ***foi declarada nula a convenção datada de 15/7/2023****, presidida sem o membro gestor [...]; bem como b) determinado que o representante, então Segundo Vice–Presidente, assumisse a condução dos trabalhos eleitorais, ficando expressamente advertido que "o processo deve ser realizado de maneira presencial e virtual, para fins de acompanhamento pelo Ministério Público Eleitoral, caso assim o queira [...]. Desse modo,* ***DETERMINO a INTERVENÇÃO no PARTIDO RENOVADOR TRABALHISTA BRASILEIRO para fins de convocação de novas eleições no PRTB, até 29 de fevereiro de 2024****, com a escolha de seu Presidente, Diretório Nacional, Comissão Executiva e delegados [...]."*

O mesmo se aplica para o resumo com a maior pontuação BLANC:

- Texto resumido

  - *"****O pedido de antecipação dos efeitos da tutela foi indeferido****, pois* ***não há fundado receio de dano irreparável ou de difícil reparação****. Além disso,* ***não se permite aferir a verossimilhança das alegações da requerente****, considerando-se que a declaração de ausência de justa causa para a desfiliação partidária pressupõe instrução probatória ampla. O Tribunal Superior Eleitoral também se pronunciou sobre a* ***inadmissibilidade da medida liminar****.* ***A celeridade processual já está contemplada nos processos regidos pela resolução em foco****.* ***A Secretaria deve providenciar a retificação da autuação nos termos do quanto requerido pela requerente****."*

- Texto original com partes suprimidas para uma visão geral mais direta:

- *"**Indefiro o pedido de antecipação dos efeitos da tutela**, tendo em vista que, em sede de cognição sumária inerente à atual fase processual [...]. Com efeito, **não há fundado receio de dano irreparável ou de difícil reparação**, especialmente considerando-se a celeridade dos feitos dessa natureza. Além disso, **não se permite aferir a verossimilhança das alegações da requerente**, considerando-se que a declaração de ausência de justa causa para a desfiliação partidária pressupõe instrução probatória ampla, o que ainda não se efetivou nos autos. Nessa esteira, destaco que o c. Tribunal Superior Eleitoral assim se pronunciou relativamente à **inadmissibilidade da medida liminar**, conforme segue: [...] **A celeridade processual, inerente aos feitos eleitorais, já está contemplada nos processos regidos pela resolução em foco**, pois, além da preferência a eles conferida, hão de ser processados e julgados no prazo de 60 dias. [...] Economia e celeridade processual não têm a força de aniquilar a garantia do devido processo legal [...]. **Por fim, providencie a Secretaria a retificação da autuação nos termos do quanto requerido pela requerente** à fl. 24. São Paulo, 13 de dezembro de 2011 [...]."*

Como conclusão dos experimentos realizados, verificou-se que tanto o Elasticsearch quanto o LLM são eficazes na ordenação de documentos similares. No entanto, ao utilizar o LLM para gerar resumos dos documentos retornados pelo Elasticsearch, foi possível alcançar ganhos significativos na qualidade das respostas apresentadas ao usuário final, algo que não poderia ser obtido apenas com o uso do Elasticsearch.

Portanto, devido à capacidade do LLM de gerar respostas mais concisas e com uma ordenação de similaridade adequada, a adoção da RAG proposta neste trabalho para buscar atos judiciais similares se mostra vantajosa em comparação com uma busca realizada puramente no Elasticsearch.

5 CONCLUSÕES

Conforme mostrado no Capítulo 1, o sistema jurídico brasileiro ainda enfrenta uma significativa carência de ferramentas tecnológicas adequadas para otimizar a eficiência das tarefas relacionadas às tramitações processuais. A ausência dessas tecnologias não apenas sobrecarrega os profissionais da área, como também contribui para a lentidão e a ineficiência no manejo dos processos judiciais. Não é por acaso que, nos últimos três anos, observou-se um crescimento de mais de 170% no número de estudos que exploram o potencial da IA para a área jurídica.

Diante desse cenário, o objetivo geral deste trabalho foi propor uma RAG capaz de identificar atos judiciais similares da JE, uma atividade crucial no meio jurídico. Ao otimizá-la, é possível tornar o trabalho dos profissionais, como magistrados, servidores e advogados, mais eficiente, contribuindo para um Judiciário mais ágil e preciso.

Para avaliar o desempenho da RAG proposta, foram realizados cinco experimentos, cada um utilizando um conjunto de dados de tribunais específicos da JE. Nesses experimentos, compararam-se as entropias dos resultados de buscas realizadas com um sistema de busca tradicional, o Elasticsearch, e com um LLM, o Llama 3.1. O objetivo foi aprimorar os resultados obtidos pelo Elasticsearch por meio de tarefas realizadas pelo LLM, entre elas a produção de resumos, os quais foram avaliados utilizando a métrica BLANC.

Os resultados indicaram que o modelo melhorou significativamente a qualidade das respostas fornecidas pelo Elasticsearch, particularmente no que tange ao resumo dos atos judiciais, tarefa que o Elasticsearch é incapaz de realizar. Embora a ordenação das decisões pela RAG não tenha mostrado uma melhoria substancial na entropia em relação à ordenação do Elasticsearch, que atingiu entropia zero em 4 dos 5 experimentos, os atos judiciais reordenados pelo LLM apresentaram entropia zero em todos os experimentos e demonstraram ser realmente similares aos documentos especificados nas buscas.

Além disso, observou-se que os resultados não variaram significativamente entre textos de diferentes tribunais eleitorais. Isso ocorreu apesar das variações na escrita entre tribunais e da falta de padronização nas classes judiciais entre os tribunais da JE. Muitos TREs, inclusive, necessitam de ajustes para adequar suas classes com as definidas pelo CNJ.

A única variação observada foi na pontuação BLANC da qualidade dos resumos, que tende a diminuir à medida que o texto aumenta. Dessa forma, tribunais com atos judiciais mais curtos tendem a gerar resumos melhores. A maior pontuação registrada foi de 0,267 (TRE-SP), enquanto a menor foi de 0,045 (TSE). Mesmo com a menor pontuação observada, o resumo ainda foi considerado bom.

Portanto, em resposta ao objetivo geral deste trabalho, conclui-se que a adoção de um modelo de IA para a busca de atos judiciais similares proporciona ganhos significativos em comparação com os sistemas de recuperação de informação tradicionais. O uso de IA permite a identificação de documentos realmente similares, mesmo que não estejam entre os mais bem ranqueados nos sistemas tradicionais, e oferece respostas resumidas que facilitam a organização e celeridade para o usuário final.

É importante destacar que, apesar da RAG implementada ter utilizado o Elasticsearch como mecanismo de recuperação de informação tradicional e o Llama 3.1 405B Instruct como modelo para aprimorar as respostas ao usuário final, a RAG proposta neste trabalho é agnóstica em relação às ferramentas utilizadas. Assim, poderia ser adotado outro banco de dados como mecanismo de busca, bem como qualquer outro LLM gerador de texto em português.

5.1 Trabalhos futuros

Para trabalhos futuros, sugere-se a expansão dos experimentos para avaliar o desempenho de outros modelos de LLM. Embora este estudo tenha utilizado apenas um modelo, há diversos outros modelos de IA generativa, como o ChatGPT, que podem desempenhar funções semelhantes. Também é válido explorar modelos que exijam menos poder de processamento, um dos principais desafios dos LLMs.

Além disso, como continuação deste estudo, também seriam pertinentes os experimentos com textos de tribunais de outros segmentos da Justiça que não a Eleitoral, como a Justiça do Trabalho, Justiça Militar e Justiça Federal. Da mesma forma, considerando que, em geral, os modelos de LLM generalistas apresentam melhor desempenho na língua inglesa, também seriam válidos experimentos com textos escritos em inglês ou traduzidos do português para o inglês.

Por fim, expandir a RAG para outras tarefas poderia trazer benefícios significativos, como a identificação de precedentes relevantes, a detecção de legislações aplicáveis e a sugestão de atribuição de classes judiciais para os casos dos TREs que necessitam de saneamento em seus atos judiciais. Todas essas tarefas poderiam otimizar o fluxo de trabalho nos tribunais e potencializando a eficácia das decisões judiciais.

REFERÊNCIAS

AGUIAR, A. *et al.* Text classification in legal documents extracted from lawsuits in brazilian courts. *In*: BRITTO, A.; DELGADO, K. V. (ed.). **Intelligent Systems**. [*S.l.: s.n.*]: Springer International Publishing, 2021. p. 586–600. ISBN 978-3-030-91699-2.

AI, M. **MODEL_CARD.md - Meta-Llama-3.1**. 2024. Acessado em 24/08/2024. Disponível em: https://github.com/meta-llama/llama-models/blob/main/models/llama3_1/MODEL_CARD.md.

ALBERTA, I. S. . T. U. of. **COLIEE 2022 - Competition on Legal Information Extraction/Entailment**. 2022. Acessado em 29/03/2024. Disponível em: https://sites.ualberta.ca/~rabelo/COLIEE2022/.

BAR-HILLEL, Y. **Theoretical Aspects of the Mechanization of Literature Searching**. Wiesbaden: Vieweg+Teubner Verlag, 1962. 406–443 p. ISBN 978-3-322-96260-7. Disponível em: https://doi.org/10.1007/978-3-322-96260-7_10.

BENTO, F. M.; TEIVE, R. C. G. Classificação de documentos jurídicos utilizando a arquitetura transformer: uma análise comparativa com algoritmos tradicionais de machine learning e ChatGPT. **Brazilian Journal of Development**, South Florida Publishing LLC, v. 9, n. 6, p. 20208–20224, jun. 2023. ISSN 2525-8761. Disponível em: http://dx.doi.org/10.34117/bjdv9n6-97.

BONAB, H.; SARWAR, S. M.; ALLAN, J. Training effective neural clir by bridging the translation gap. *In*: **Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: Association for Computing Machinery, 2020. p. 9–18. ISBN 9781450380164. Disponível em: https://doi.org/10.1145/3397271.3401035.

BOURNE, C. **Methods of Information Handling**. J. Wiley, 1963. (Information sciences). ISBN 9780471091509. Disponível em: https://books.google.com.br/books?id=IBk7AAAAIAAJ.

BROWN, T. *et al.* Language models are few-shot learners. *In*: LAROCHELLE, H. *et al.* (ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2020. v. 33, p. 1877–1901. Disponível em: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

B.V., E. **Similarity module**. 2024. Acessado em 30/05/2024. Disponível em: https://www.elastic.co/guide/en/elasticsearch/reference/current/index-modules-similarity.html.

CHATTERJEE, N. *et al.* Information retrieval based legal search system. **International Journal of Next-Generation Computing**, Perpetual Innovation Media Pvt. Ltd., fev. 2023. ISSN 2229-4678. Disponível em: http://dx.doi.org/10.47164/ijngc.v14i1.1004.

(CNJ), C. N. de J. **Justiça 4.0: Inteligência Artificial está presente na maioria dos tribunais brasileiros**. 2022. Acessado em 29/03/2024. Disponível em: https://www.cnj.jus.br/justica-4-0-inteligencia-artificial-esta-presente-na-maioria-dos-tribunais-brasileiros/.

(CNJ), C. N. de J. **Resultados Pesquisa IA no Poder Judiciário - 2022**. 2022. Acessado em 29/03/2024. Disponível em: https://paineisanalytics.cnj.jus.br/single/?appid=9e4f18ac-e253-4893-8ca1-b81d8af59ff6&sheet=b8267e5a-1f1f-41a7-90ff-d7a2f4ed34ea&lang=pt-BR&theme=IA_PJ&opt=ctxmenu,currsel&select=language,BR.

(CNJ), C. N. de J. **Justiça em Números 2023**. Brasília - DF: Conselho Nacional de Justiça (CNJ), 2023. ISBN 978-65-5972-116-0. Disponível em: https://www.cnj.jus.br/wp-content/uploads/2023/08/justica-em-numeros-2023.pdf.

CONNEAU, A.; LAMPLE, G. Cross-lingual language model pretraining. *In*: _____. **Proceedings of the 33rd International Conference on Neural Information Processing Systems**. Red Hook, NY, USA: Curran Associates Inc., 2019.

CORPORATION, N. **Access to NVIDIA NIM Now Available Free to Developer Program Members**. 2024. Acessado em 15/08/2024. Disponível em: https://developer.nvidia.com/blog/access-to-nvidia-nim-now-available-free-to-developer-program-members/.

CUI, J. *et al.* Chatlaw: Open-source legal large language model with integrated external knowledge bases. **ArXiv**, abs/2306.16092, 2023. Disponível em: https://api.semanticscholar.org/CorpusID:259274889.

DEVLIN, J. *et al.* BERT: pre-training of deep bidirectional transformers for language understanding. *In*: BURSTEIN, J.; DORAN, C.; SOLORIO, T. (ed.). **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)**. Association for Computational Linguistics, 2019. p. 4171–4186. Disponível em: https://doi.org/10.18653/v1/n19-1423.

DUBEY, A. *et al.* **The Llama 3 Herd of Models**. arXiv, 2024. Disponível em: https://arxiv.org/abs/2407.21783.

FRAENKEL, A. S. Legal information retrieval. *In*: ALT, F. L.; RUBINOFF, M. (ed.). Elsevier, 1969, (Advances in Computers, v. 9). p. 113–178. Disponível em: https://www.sciencedirect.com/science/article/pii/S0065245808603125.

GOMES, T. A. **Avaliação de técnicas de similaridade textual na uniformização de jurisprudência**. 2021. Dissertação (Mestrado) — Universidade de Brasília (UnB), Brasília, Brasil, 2021. Disponível em: http://repositorio2.unb.br/jspui/handle/10482/40798.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing (2Nd Edition)**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2009. ISBN 0131873210.

KANAPALA, A. *et al.* Applying an information retrieval approach to retrieve relevant articles in the legal domain. **Annals of Data Science**, Springer Science and Business Media LLC, set. 2022. ISSN 2198-5812. Disponível em: http://dx.doi.org/10.1007/s40745-022-00442-4.

KIM, M.-Y.; RABELO, J.; GOEBEL, R. Statute law information retrieval and entailment. *In*: **Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law**. ACM, 2019. (ICAIL '19). Disponível em: http://dx.doi.org/10.1145/3322640.3326742.

LEWIS, P. *et al.* Retrieval-augmented generation for knowledge-intensive nlp tasks. *In*: **Proceedings of the 34th International Conference on Neural Information Processing Systems**. Red Hook, NY, USA: Curran Associates Inc., 2020. (NIPS '20). ISBN 9781713829546.

LI, H. *et al.* Sailer: Structure-aware pre-trained language model for legal case retrieval. *In*: **Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: Association for Computing Machinery, 2023. (SIGIR '23), p. 1035–1044. ISBN 9781450394086. Disponível em: https://doi.org/10.1145/3539618.3591761.

LIN, C.-Y. ROUGE: A package for automatic evaluation of summaries. *In*: **Text Summarization Branches Out**. Barcelona, Spain: Association for Computational Linguistics, 2004. p. 74–81. Disponível em: https://aclanthology.org/W04-1013.

MANNING, C. D.; RAGHAVAN, P.; SCHüTZE, H. **Introduction to Information Retrieval**. USA: Cambridge University Press, 2008. ISBN 0521865719.

NOGUTI, M. Y.; VELLASQUES, E.; OLIVEIRA, L. S. Legal document classification: An application to law area prediction of petitions to public prosecution service. *In*: **2020 International Joint Conference on Neural Networks (IJCNN)**. [*S.l.: s.n.*], 2020. p. 1–8.

OLIVEIRA, R. A. N. de; JUNIOR, M. C. Experimental analysis of stemming on jurisprudential documents retrieval. **Information**, v. 9, n. 2, 2018. ISSN 2078-2489. Disponível em: https://www.mdpi.com/2078-2489/9/2/28.

OLIVEIRA, R. S. de; NASCIMENTO, E. G. S. **Analysing similarities between legal court documents using natural language processing approaches based on Transformers**. arXiv, 2022. Disponível em: https://arxiv.org/abs/2204.07182.

OPENAI *et al.* **GPT-4 Technical Report**. arXiv, 2024. Disponível em: https://arxiv.org/abs/2303.08774.

PARASHAR, S.; MITTAL, N.; MEHTA, P. Casrank: A ranking algorithm for legal statute retrieval. **Multimedia Tools and Applications**, Springer Science and Business Media LLC, v. 83, n. 2, p. 5369–5386, jun. 2023. ISSN 1573-7721. Disponível em: http://dx.doi.org/10.1007/s11042-023-15464-0.

PARIKH, V. *et al.* **AILA 2021 - Artificial Intelligence for Legal Assistance**. 2021. Acessado em 29/03/2024. Disponível em: https://sites.google.com/view/aila-2021/track-description?authuser=0.

PRASAD, N.; BOUGHANEM, M.; DKAKI, T. Exploring large language models and hierarchical frameworks for classification of large unstructured legal documents. *In*: GOHARIAN, N. *et al.* (ed.). **Advances in Information Retrieval**. Cham: Springer Nature Switzerland, 2024. p. 221–237. ISBN 978-3-031-56060-6.

RADFORD, A. *et al.* Improving language understanding by generative pre-training. 2018. Disponível em: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

ROCHA, L. V. **Análise da busca, uso e avaliação dos serviços da biblioteca pelos assessores de ministros do Supremo Tribunal Federal em relação as suas necessidades de informação jurídica**. 2011. Dissertação (Mestrado) — Universidade de Brasília (UnB), Brasília, Brasil, 2011. Disponível em: https://bibliotecadigital.stf.jus.br/xmlui/handle/123456789/1087.

SANSONE, C.; SPERLí, G. Legal information retrieval systems: State-of-the-art and open issues. **Information Systems**, v. 106, p. 101967, 2022. ISSN 0306-4379. Disponível em: https://www.sciencedirect.com/science/article/pii/S0306437921001551.

SHAGHAGHIAN, S. *et al.* Customizing contextualized language models for legal document reviews. *In*: **2020 IEEE International Conference on Big Data (Big Data)**. [*S.l.: s.n.*], 2020. p. 2139–2148.

SHANNON, C. E. A mathematical theory of communication. **The Bell System Technical Journal**, v. 27, n. 3, p. 379–423, 1948.

SHU, D. *et al.* Lawllm: Law large language model for the us legal system. **arXiv preprint arXiv:2407.21065**, 2024.

SOUZA, E. *et al.* An information retrieval pipeline for legislative documents from the brazilian chamber of deputies. *In*: _____. **Legal Knowledge and Information Systems**. IOS Press, 2021. Disponível em: http://dx.doi.org/10.3233/FAIA210326.

SURDEN, H. Chatgpt, ai large language models, and law. **Fordham L. Rev.**, HeinOnline, v. 92, p. 1941, 2023.

TOSTA, M. D. **Segmentação de Documentos Jurídicos usando Supervisão Fraca**. 2022. Dissertação (Mestrado) — Universidade Federal de Mato Grosso do Sul (UFMS), Campo Grande, Brasil, 2022. Disponível em: https://repositorio.ufms.br/handle/123456789/5852.

VASILYEV, O.; DHARNIDHARKA, V.; BOHANNON, J. Fill in the BLANC: Human-free quality estimation of document summaries. *In*: EGER, S. *et al.* (ed.). **Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems**. Online: Association for Computational Linguistics, 2020. p. 11–20. Disponível em: https://aclanthology.org/2020.eval4nlp-1.2.

VASILYEV, O. *et al.* **Sensitivity of BLANC to human-scored qualities of text summaries**. arXiv, 2020. Disponível em: https://arxiv.org/abs/2010.06716.

VASWANI, A. *et al.* Attention is all you need. *In*: GUYON, I. *et al.* (ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2017. v. 30. Disponível em: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

WAGH, R.; ANAND, D. Application of citation network analysis for improved similarity index estimation of legal case documents : A study. *In*: **2017 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)**. [*S.l.: s.n.*], 2017. p. 1–5.

WESTERMANN, H.; SAVELKA, J.; BENYEKHLEF, K. Paragraph similarity scoring and fine-tuned bert for legal information retrieval and entailment. *In*: OKAZAKI, N. *et al.* (ed.). **New Frontiers in Artificial Intelligence**. Cham: Springer International Publishing, 2021. p. 269–285. ISBN 978-3-030-79942-7.

YANG, G. Adaptive retrieval method of legal information based on artificial intelligence. *In*: **2020 IEEE International Conference on Industrial Application of Artificial Intelligence (IAAI)**. [*S.l.: s.n.*], 2020. p. 13–17.

ZHANG, L.; ZHAO, X. An overview of cross-language information retrieval. *In*: _____. **Artificial Intelligence and Security**. Springer International Publishing, 2020. p. 26–37. ISBN 9783030578848. Disponível em: http://dx.doi.org/10.1007/978-3-030-57884-8_3.

ZHANG, R. *et al.* Result diversification for legal case retrieval. *In*: **Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region**. ACM, 2023. (SIGIR-AP '23). Disponível em: http://dx.doi.org/10.1145/3624918.3625319.

ZHEBEL, V.; ZUBAREV, D.; SOCHENKOV, I. Different approaches in cross-language similar documents retrieval in the legal domain. *In*: KARPOV, A.; POTAPOVA, R. (ed.). **Speech and Computer**. Cham: Springer International Publishing, 2020. p. 679–686. ISBN 978-3-030-60276-5.

ŠAVELKA, J.; ASHLEY, K. D. Legal information retrieval for understanding statutory terms. **Artificial Intelligence and Law**, Springer Science and Business Media LLC, v. 30, n. 2, p. 245–289, jul. 2021. ISSN 1572-8382. Disponível em: http://dx.doi.org/10.1007/s10506-021-09293-5.