UNIVERSIDADE FEDERAL DE GOIÁS (UFG)

INSTITUTO DE INFORMÁTICA (INF)

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO (PPGCC)

WILLGNNER FERREIRA SANTOS

Avaliação de Grandes Modelos de Linguagem para Classificação de Documentos Jurídicos em Português

GOIÂNIA
2024

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES

E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico

[X] Dissertação [ ] Tese [ ] Outro*:_____

*No caso de mestrado/doutorado profissional, indique o formato do Trabalho de Conclusão de Curso, permitido no documento de área, correspondente ao programa de pós-graduação, orientado pela legislação vigente da CAPES.

Exemplos: Estudo de caso ou Revisão sistemática ou outros formatos.

2. Nome completo do autor

Willgnner Ferreira Santos

3. Título do trabalho

Avaliação de Grandes Modelos de Linguagem para Classificação de Documentos Jurídicos em Português

4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:
a) consulta ao(à) autor(a) e ao(à) orientador(a);
b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.
O documento não será disponibilizado durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Arlindo Rodrigues Galvao Filho**, **Professor do Magistério Superior**, em 13/12/2024, às 19:20, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Willgnner Ferreira Santos**, **Usuário Externo**, em 13/12/2024, às 21:47, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5039720** e o código CRC **4B2E42FA**.

WILLGNNER FERREIRA SANTOS

Avaliação de Grandes Modelos de Linguagem para Classificação de Documentos Jurídicos em Português

Dissertação apresentada ao Programa de Pós–Graduação em Ciência da Computação (PPGCC), do Instituto de Informática (INF), da Universidade Federal de Goiás (UFG), como requisito para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação.

Linha de Pesquisa: Sistemas Inteligentes e Aplicações.

Orientador: Prof. Dr. Arlindo Rodrigues Galvão Filho.

Co-Orientador: Prof. Dr. Sávio Salvarino Teles de Oliveira.

GOIÂNIA
2024

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

SANTOS, WILLGNNER FERREIRA
Avaliação de Grandes Modelos de Linguagem para Classificação de Documentos Jurídicos em Português [manuscrito] / WILLGNNER FERREIRA SANTOS. - 2024.
CII, 102 f.

Orientador: Prof. Dr. Arlindo Rodrigues Galvão Filho; co-orientador Dr. Sávio Salvarino Teles de Oliveira.
Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2024.
Bibliografia. Apêndice.
Inclui siglas, abreviaturas, lista de figuras, lista de tabelas.

1. Grandes Modelos de Linguagem. 2. Classificação de Documentos Jurídicos. 3. Processamento de Linguagem Natural. I. Rodrigues Galvão Filho, Arlindo , orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE DISSERTAÇÃO

Ata nº **36** da sessão de Defesa de Dissertação de **Willgnner Ferreira Santos**, que confere o título de Mestre em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos vinte e seis dias do mês de novembro de dois mil e vinte e quatro, a partir das nove horas, via webconferência, realizou-se a sessão pública de Defesa de Dissertação intitulada **"Avaliação de Grandes Modelos de Linguagem para Classificação de Documentos Jurídicos em Português"**. Os trabalhos foram instalados pelo Orientador, Professor Doutor Arlindo Rodrigues Galvão Filho (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professor Doutor Sávio Salvarino Teles de Oliveira (INF/UFG), coorientador; Professor Doutor Rodrigo Zempulski Fanucchi (COPEL Energia), membro titular externo; e Professor Doutor Anderson da Silva Soares (INF/UFG), membro titular interno. A realização da banca ocorreu por meio de videoconferência. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Arlindo Rodrigues Galvão Filho, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos vinte e seis dias do mês de novembro de dois mil e vinte e quatro.

TÍTULO SUGERIDO PELA BANCA

Documento assinado eletronicamente por **Savio Salvarino Teles De Oliveira**, **Professor do Magistério Superior**, em 26/11/2024, às 10:50, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Arlindo Rodrigues Galvao Filho**, **Professor do Magistério Superior**, em 26/11/2024, às 10:51, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **RODRIGO ZEMPULSKI FANUCCHI**, **Usuário Externo**, em 26/11/2024, às 10:51, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Anderson Da Silva Soares**, **Professor do Magistério Superior**, em 26/11/2024, às 10:51, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Willgnner Ferreira Santos**, **Usuário Externo**, em 26/11/2024, às 11:15, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Referência: Processo nº 23070.053494/2024-82 SEI nº 4930064

Willgnner Ferreira Santos

Graduado em Engenharia de Computação pela Pontifícia Universidade Católica de Goiás (PUC Goiás), onde atuou como monitor em disciplinas da Escola de Ciências Exatas e da Computação (ECEC). Durante a graduação, integrou os grupos de pesquisa em Ciência Computacional e Planejamento Urbano, desenvolvendo dois projetos de iniciação científica no departamento de Computação. Trabalha no Serviço Nacional de Aprendizagem Industrial (SENAI) Goiás, focado em projetos de Inteligência Artificial (IA) para a indústria e educação. Possui experiência em ciência da computação, com ênfase em aprendizado profundo, aprendizado de máquina, visão computacional, processamento digital de imagens, ciência de dados, mineração de dados e processamento de linguagem natural.

Dedico este trabalho aos meus pais, Nirto Ferreira dos Santos e Hilda das Dores Gomes dos Santos, pelo amor, apoio incondicional e por sempre acreditarem em meu potencial. À minha amada namorada, Jayni Rodrigues de Sousa Melo, por estar ao meu lado em todos os momentos e por me oferecer carinho, paciência e encorajamento durante esta jornada. E, acima de tudo, dedico a Deus, pela força e sabedoria que me guiou em cada etapa deste percurso.

Agradecimentos

Em primeiro lugar, agradeço a Deus, por me conceder forças e sabedoria para enfrentar os momentos mais desafiadores desta caminhada.

Aos meus pais, Nirto Ferreira dos Santos e Hilda das Dores Gomes dos Santos, por todo o amor, apoio e incentivo ao longo da minha trajetória.

À minha namorada e futura esposa, Jayni Rodrigues de Sousa Melo, por todo o carinho, paciência e compreensão durante essa jornada. Sua presença foi importante para que eu mantivesse o foco e seguisse em frente.

Aos meus amigos Douglas Vieira do Nascimento, Walcy Júnior Rios e João Paulo Cavalcante Presa, pelo apoio prestado.

Um agradecimento ao meu orientador, Prof. Dr. Arlindo Rodrigues Galvão Filho, que me acompanha desde a graduação, onde iniciei no Laboratório de Computação Científica (LCC) da PUC Goiás. Sua orientação, paciência e sabedoria foram fundamentais para minha formação e para o desenvolvimento deste trabalho.

Agradeço muito ao meu coorientador, Prof. Dr. Sávio Salvarino Teles de Oliveira, pela dedicação incansável ao longo deste processo. Em momentos importantes, ele esteve presente, revisando o trabalho até altas horas da noite e começando cedo no dia seguinte, sempre disposto a contribuir sem medir esforços. Sempre disposto a contribuir, sua dedicação foi vital para o progresso deste projeto.

Aos Professores Doutores Anderson da Silva Soares e Rodrigo Zempulski Fanucchi por gentilmente aceitarem o convite para compor a banca de avaliação da minha dissertação. Sou grato pelo tempo e *feedback* que contribuíram para o aprimoramento deste trabalho.

À UFG e ao INF, por me proporcionar o ambiente acadêmico e os recursos necessários para a realização deste trabalho.

Às secretárias do PPGCC, Mariana Alves Rodrigues Santana e Mirian Castro Portilho Dias Amorim, pelo suporte e pela ajuda em momentos fundamentais do curso.

Agradeço à Defensoria Pública do Estado de Goiás (DPE-GO) por fornecer os dados que possibilitaram este estudo e pelo apoio dos Defensores Públicos Tiago Ordones Rêgo Bicalho e Guilherme Vaz, essenciais para o desenvolvimento deste trabalho. Estendo minha gratidão ao Diretor de Tecnologia da Informação da DPE-GO, Leandro

Silva de Lima, pelo compromisso e pela colaboração que foram indispensáveis para a realização deste trabalho.

Ao meu ex-gestor e parceiro de aulas, Pabllo Borges Cardoso, Chefe do Departamento de Produção de Software na DPE-GO, por todo o aprendizado e pela colaboração durante nossa jornada profissional.

Agradeço à minha ex-gestora, Patrícia Bianca Nascimento e Silva Gomes, Chefe do Departamento de Suporte em Tecnologia e Comunicação na DPE-GO, pela parceria, pela primeira oportunidade na instituição e por sempre incentivar meu mestrado.

Aos colegas da DPE-GO, que de alguma forma contribuíram para este projeto, meu sincero agradecimento.

Agradeço também a todos que, direta ou indiretamente, colaboraram com o desenvolvimento deste trabalho, oferecendo apoio, orientação e incentivo nos momentos mais necessários.

Por fim, agradeço ao Grupo de Pesquisa de Processamento de Linguagem Natural (PLN), liderado pelo Prof. Dr. Sávio Salvarino Teles de Oliveira, por todo o suporte acadêmico durante a condução deste trabalho.

E, por último, agradeço a mim mesmo por não ter desistido, por ter encontrado resiliência para superar as adversidades e seguir adiante. Sei que a jornada não foi fácil, mas cada passo valeu a pena.

Artificial intelligence is the science of making machines do things that would require intelligence if done by humans.

Marvin Minsky,
Semantic Information Processing.

Resumo

SANTOS, W. F.. **Avaliação de Grandes Modelos de Linguagem para Classificação de Documentos Jurídicos em Português**. GOIÂNIA, 2024. 102p. Dissertação de Mestrado. Programa de Pós-Graduação em Ciência da Computação (PPGCC), Instituto de Informática (INF), Universidade Federal de Goiás (UFG).

A crescente demanda processual nas instituições jurídicas tem gerado sobrecarga de trabalho, impactando a eficiência do sistema jurídico. Esse cenário, agravado pela limitação de recursos humanos, ressalta a necessidade de soluções tecnológicas que agilizem o processamento e a análise de documentos. Diante dessa realidade, este trabalho propõe um *pipeline* para a automatização da classificação desses documentos, avaliando quatro métodos de representação de textos jurídicos na entrada do *pipeline*: texto original, resumos, centroides e descrições dos documentos. O *pipeline* foi desenvolvido e testado na Defensoria Pública do Estado de Goiás (DPE-GO). Cada abordagem implementa uma estratégia específica para estruturar os textos de entrada, com o objetivo de aprimorar a capacidade dos modelos de interpretar e classificar documentos jurídicos. Foi introduzido um novo conjunto de dados em português, elaborado para essa aplicação, e o desempenho de Grandes Modelos de Linguagem (LLMs) foi avaliado em tarefas de classificação. Os resultados da análise demonstram que o uso de resumos melhora a acurácia da classificação e maximiza o *F1-score*, otimizando o uso de LLMs ao reduzir a quantidade de *tokens* processados, sem comprometer a precisão. Esses resultados evidenciam o impacto das representações textuais dos documentos e o potencial dos LLMs na classificação automática de documentos jurídicos, como no caso da DPE-GO. As contribuições deste trabalho apontam que a aplicação de LLMs, combinada com representações textuais otimizadas, pode aumentar a produtividade e a qualidade dos serviços prestados pelas instituições jurídicas, promovendo avanços na eficiência do sistema jurídico como um todo.

Palavras–chave

Grandes Modelos de Linguagem, Classificação de Documentos Jurídicos, Processamento de Linguagem Natural.

Abstract

SANTOS, W. F.. **Evaluation of Large Language Models for Legal Document Classification in Portuguese**. GOIÂNIA, 2024. 102p. MSc. Dissertation. Programa de Pós-Graduação em Ciência da Computação (PPGCC), Instituto de Informática (INF), Universidade Federal de Goiás (UFG).

The increasing procedural demand in judicial institutions has caused a workload overload, impacting the efficiency of the legal system. This scenario, exacerbated by limited human resources, highlights the need for technological solutions to streamline the processing and analysis of documents. In light of this reality, this work proposes a pipeline for automating the classification of these documents, evaluating four methods of representing legal texts at the pipeline's input: original text, summaries, centroids, and document descriptions. The pipeline was developed and tested at the Public Defender's Office of the State of Goiás (DPE-GO). Each approach implements a specific strategy to structure the input texts, aiming to enhance the models' ability to interpret and classify legal documents. A new Portuguese dataset was introduced, specifically designed for this application, and the performance of Large Language Models (LLMs) was evaluated in classification tasks. The analysis results demonstrate that the use of summaries improves classification accuracy and maximizes the F1-score, optimizing the use of LLMs by reducing the number of tokens processed without compromising precision. These findings highlight the impact of textual representations of documents and the potential of LLMs for the automatic classification of legal documents, as in the case of DPE-GO. The contributions of this work indicate that the application of LLMs, combined with optimized textual representations, can significantly increase the productivity and quality of services provided by judicial institutions, promoting advancements in the overall efficiency of the legal system.

Keywords

Large Language Models, Legal Document Classification, Natural Language Processing.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

Universidade Federal de Goiás (UFG)
Instituto de Informática (INF)
Pós-Graduação em Ciência da Computação (PPGCC)
Pontifícia Universidade Católica de Goiás (PUC Goiás)
Escola de Ciências Exatas e da Computação (ECEC)
Serviço Nacional de Aprendizagem Industrial (SENAI)
Laboratório de Computação Científica (LCC)
Inteligência Artificial (IA)
Large Language Models (LLMs)
Processamento de Linguagem Natural (PLN)
Defensoria Pública do Estado de Goiás (DPE-GO)
Redes Neurais Profundas (RNPs)
Redes Neurais Convolucionais (RNCs)
Redes Neurais Recorrentes (RNRs)
Generative Pre-Trained Transformer 3 (GPT-3)
Bidirecional Encoder Representations from Transformers (BERT)
Named Entity Recognition (NER)
Term Frequency-Inverse Document Frequency (TF-IDF)
Large Language Model for Meta AI (LLaMA)
Byte Pair Encoding (BPE)
Clue And Reasoning Prompting (CARP)
Editing Intent Classification (EIC)
Internet Movie Database (IMDB)
Machine Learning (ML)
Federated Learning (FL)
Occlusion-based Hierarchical Explanation-extractor (Ob-HEx)
International Language Data Collection Expert (ILDC-Expert)
Multi-stage Encoder-based Supervised with Clustering (MESc)
Cadastro de Pessoa Física (CPF)
OpenDocument (ODF)

Hypertext Markup Language (HTML)
Portable Document Format (PDF)
Uniform Resource Locato (URLs)
Lei Geral de Proteção de Dados Pessoais (LGPD)
Registro Geral (RG)
Comma-Separated Values (CSV)
Graphics Processing Unit (GPUs)
Application Programming Interface (API)
Tensor processing Unit (TPUs)
Memória de Acesso Aleatório (RAM)
Gigabyte (GB)
True Positives (TP)
True Negatives (TN)
False Positives FP)
False Negatives (FN)
Receiver Operating Characteristic (ROC)
True Positive Rate (TPR)
False Positive Rate (FPR)
Area Under the Precision-Recall Curve (AUC-PR)
Coeficiente de Correlação de Matthews (MCC)
Diretoria de Tecnologia da Informação (DTI)
Departamento de Ciência de Dados (DCD)
Support Vector Machine (SVM)

CAPÍTULO 1

Introdução

Nos últimos anos, instituições do setor jurídico têm enfrentado desafios relacionados ao aumento de demandas processuais e à limitação de recursos humanos [58]. Esses problemas afetam diretamente a eficiência dos fluxos de trabalho, principalmente diante do volume crescente de documentos e petições jurídicas, que demandam soluções tecnológicas eficazes para organização e análise [1]. A correta classificação e organização desses documentos é fundamental para otimizar os processos e garantir que os serviços jurídicos sejam prestados com maior agilidade, qualidade e consistência.

A Inteligência Artificial (IA) tem se mostrado uma aliada poderosa nesse contexto, oferecendo alternativas viáveis para a automatização de tarefas repetitivas e intensivas em tempo [77]. Ferramentas baseadas em IA já são utilizadas em diversos países, como nos Estados Unidos e na Europa, onde escritórios de advocacia e tribunais têm explorado tecnologias para análise automatizada de contratos, classificação de documentos e até previsão de decisões judiciais [46, 33, 56]. Grandes Modelos de Linguagem, do inglês, Large Language Models (LLMs) [52], em particular, destacam-se por sua capacidade de lidar com grandes volumes de dados textuais e realizar tarefas complexas de Processamento de Linguagem Natural (PLN), como a classificação de documentos jurídicos, a extração de informações e a geração de textos [43].

No entanto, embora os LLMs apresentem alto desempenho, sua eficácia pode ser ampliada por meio de metodologias estruturadas de pré-processamento e pós-processamento de dados, que ajudam a refinar os resultados e a otimizar o desempenho dos modelos [83, 53]. Essas metodologias incluem o uso de resumos, que fornecem representações textuais mais concisas e eficientes para os modelos de classificação [19]. Nesse sentido, o desenvolvimento de *pipelines* bem estruturados, que organizam o fluxo de processamento de dados em etapas como limpeza, normalização e preparação textual, é fundamental para extrair o máximo potencial dos LLMs [18].

Este trabalho propõe um *pipeline* para automação da classificação de documentos jurídicos, combinando diferentes métodos de representação textual com o objetivo de otimizar o desempenho em tarefas de classificação automática. Quatro abordagens principais foram investigadas, sendo o uso do texto original completo, resumos, centroides e

descrições. A eficácia dessas abordagens foi avaliada com base em métricas como acurácia e *F1-score*, considerando diferentes famílias de LLMs.

Como estudo de caso, o *pipeline* foi aplicado na Defensoria Pública do Estado de Goiás (DPE-GO) [16], que enfrenta desafios como o elevado déficit de defensores públicos [57] e o aumento constante da demanda por processos jurídicos. A DPE-GO, assim como outras instituições do setor jurídico, lida com grandes volumes de documentos complexos e enfrenta dificuldades relacionadas à limitação de recursos humanos. Essa aplicação prática permitiu validar a abordagem proposta e explorar o impacto da automação em cenários reais, destacando a relevância do uso de IA para otimizar processos jurídicos e reduzir a sobrecarga de trabalho.

Os resultados deste estudo têm implicações para o setor jurídico como um todo, oferecendo *insights* sobre como combinar o potencial dos LLMs com técnicas estruturadas de representação textual para alcançar maior eficiência e precisão em tarefas fundamentais. Além disso, o trabalho busca demonstrar que a adoção de soluções baseadas em IA pode transformar a rotina de instituições como a DPE-GO e servir como modelo para outras organizações jurídicas no Brasil e no mundo.

1.1 Problema de Pesquisa

Os documentos jurídicos normalmente são grandes, complexos e frequentemente contêm terminologia difícil de entender, estruturas de frases complicadas e referências cruzadas extensivas. Além disso, o grande volume de documentos que necessitam de classificação, sobrecarrega os fluxos de trabalho e pode resultar em atrasos, podendo ser um cenário preocupante nas defensorias públicas, onde a limitação de recursos humanos é uma preocupação constante.

As bases de dados jurídicas são caracterizadas por um grande volume de documentos longos [19]. Devido ao tamanho desses documentos, um desafio relevante na literatura é determinar a melhor forma de representar o conteúdo textual dos documentos jurídicos como entrada para algoritmos de classificação [84]. Assim, o problema de pesquisa que este estudo busca resolver é identificar: **qual é o impacto das diferentes representações textuais no desempenho de um *pipeline* de classificação automática de documentos jurídicos, com foco na avaliação de LLMs, utilizando a DPE-GO como estudo de caso?**

1.2 Objetivos

Este trabalho explora quatro métodos de representação distintos para otimizar a entrada no *pipeline* de classificação: classificação baseada no texto original, resumos,

centroides e a descrição do documento. Cada abordagem emprega uma estratégia para estruturar o texto de entrada, com o objetivo de aprimorar a capacidade do modelo de interpretar e classificar com precisão documentos jurídicos.

Este estudo apresenta a proposta de um *pipeline* para classificação automática de documentos jurídicos, utilizando diferentes LLMs e métodos de representação textual. O *pipeline* foi desenvolvido para realizar etapas de pré-processamento e pós-processamento, otimizando as representações textuais com o objetivo de maximizar o desempenho dos LLMs na tarefa de classificação jurídica. A DPE-GO é utilizada como estudo de caso para validar a abordagem proposta.

Os objetivos específicos do estudo são:

1. Desenvolver um *pipeline* para a classificação de documentos jurídicos, com aplicação no contexto da DPE-GO.
2. Comparar diferentes métodos de representação de entrada com o objetivo de otimizar o processo de classificação de documentos jurídicos.
3. Avaliar o desempenho de diferentes LLMs na tarefa de classificação jurídica.
4. Construir um conjunto de dados específico, baseado em documentos jurídicos da DPE-GO, para validar a abordagem.

1.3 Contribuições

Este estudo traz as seguintes contribuições para o campo da classificação de textos jurídicos:

1. Propõe e avalia métodos de representação textual com o objetivo de aprimorar métricas como acurácia e *F1-score* em *pipelines* de classificação jurídica.
2. Desenvolve um conjunto de dados voltado para tarefas de classificação jurídica, utilizando a DPE-GO como estudo de caso, e o torna aplicável em contextos mais amplos.
3. Analisa o desempenho de diferentes LLMs na tarefa de classificação automática de documentos jurídicos.
4. Contribui para a eficiência no tratamento de documentos jurídicos, aprimorando etapas como análise, classificação e organização de dados, com maior rapidez e precisão, além de reduzir atrasos e sobrecargas nos fluxos de trabalho.

1.4 Organização do Trabalho

Esta dissertação está organizada em seis capítulos. O capítulo 2 apresenta uma visão geral dos fundamentos conceituais do trabalho, juntamente com uma revisão da

literatura e dos trabalhos relacionados que fornecem a base para a compreensão do tema. O capítulo 3 apresenta a proposta detalhada do *Pipeline* para a Classificação Automatizada de Documentos Jurídicos utilizando LLMs, comparando diferentes formas de representação dos documentos jurídicos de entrada.

No capítulo 4 é apresentada a metodologia, detalhando os procedimentos adotados para a condução da pesquisa, incluindo a descrição do conjunto de dados utilizado, os métodos de representação aplicados e as técnicas empregadas para a análise e validação dos resultados obtidos. O capítulo 5 apresenta os resultados das avaliações realizadas no *pipeline* com diferentes modelos de linguagem, avaliando o desempenho de métodos de representação na classificação de documentos jurídicos. O capítulo 6 apresenta a conclusão do trabalho, destacando as principais contribuições da pesquisa, suas limitações e possíveis direções para estudos futuros.

CAPÍTULO 2

Conceitos e Trabalhos Relacionados

Este capítulo apresenta os conceitos fundamentais e os trabalhos relacionados que embasam o desenvolvimento deste estudo. Inicialmente, são abordados os conceitos teóricos que suportam as metodologias aplicadas, como representações textuais, classificação de texto e os LLMs. Em seguida, são discutidos estudos relevantes na literatura, destacando avanços, desafios e lacunas em problemas similares, com foco em tarefas de PLN aplicadas a textos jurídicos.

2.1 Aprendizado Profundo

O aprendizado profundo, uma subárea do aprendizado de máquina, tem se destacado em aplicações de PLN devido ao uso de Redes Neurais Profundas (RNPs) [91]. Estas redes, compostas por múltiplas camadas de neurônios artificiais, são capazes de extrair características complexas e representar dados hierarquicamente.

O aprendizado profundo em PLN utiliza Redes Neurais Convolucionais (RNCs) para extração de características textuais e Redes Neurais Recorrentes (RNRs) para modelar dependências temporais em análise e geração de texto. No entanto, a arquitetura de *Transformers*, exemplificada pelo Generative Pre-Trained Transformer 3 (GPT-3), revolucionou tarefas como tradução automática e resumo de texto [79]. Apesar de sua inovação, o GPT-3 pode não ser a melhor escolha em certos contextos devido a limitações em eficiência computacional, como o alto consumo de recursos e memória, além de custos elevados para implementação e treinamento em larga escala.

Nesses casos, modelos mais recentes ou ajustados, podem ser mais adequados para atender às necessidades específicas, oferecendo maior eficiência e capacidade de processar contextos mais extensos. As aplicações em PLN dependem de vários fatores, incluindo um pré-processamento de texto eficaz, um grande conjunto de *corpus* de treinamento, a escolha da arquitetura de modelo adequada e a otimização de hiperparâmetros [80, 2].

Exemplos de sucesso no uso do aprendizado profundo em PLN incluem tradução automática com modelos como Bidirecional Encoder Representations from Transformers

(BERT) e GPT-3, análise de sentimento, sumarização de texto, geração de texto, classificação de texto e desenvolvimento de *chatbots*. Essas aplicações demonstram como o aprendizado profundo está avançando as capacidades de PLN, permitindo interpretações e gerações de texto mais precisas e contextuais.

2.2 Processamento de Linguagem Natural

O PLN é um campo interdisciplinar situado na interseção da linguística, ciência da computação e IA, visando capacitar computadores para entender e manipular a linguagem humana [26]. Surgido na década de 1950 com experimentos em tradução automática [25], o PLN evoluiu de abordagens baseadas em regras linguísticas para métodos estatísticos e, mais recentemente, para o aprendizado de máquina e redes neurais.

Os conceitos centrais no PLN incluem morfologia, sintaxe, semântica e pragmática, essenciais para a análise de textos. Técnicas como análise sintática e Reconhecimento de Entidades Nomeadas (NER, do inglês Named Entity Recognition) [74] são importantes para compreender contextos linguísticos. O NER é uma técnica que identifica e classifica automaticamente entidades específicas em textos, como nomes de pessoas, organizações, datas e locais, sendo utilizado em aplicações como extração de informações e análise de sentimentos. O aprendizado de máquina desempenha um papel vital na modelagem desses aspectos [26].

O PLN possui aplicações variadas [51], como tradução automática, assistentes virtuais, e ferramentas de acessibilidade, todas contribuindo para facilitar a interação humano-computador e o processamento de grandes volumes de *corpus*. No entanto, desafios como ambiguidade linguística, *viés* algorítmico e privacidade de *corpus* ainda persistem. A Figura 2.1 demonstra como o PLN atua como uma ponte entre IA e linguística, beneficiando-se dos avanços em ambas as áreas [62].

Algumas tarefas de PLN incluem extração de informação, reconhecimento de fala, detecção e correção de erros gramaticais, busca inteligente e sistemas de recomendação de conteúdo. Estas aplicações destacam a importância do PLN em diversos campos, desde a medicina até a inteligência de negócios.

2.3 Classificação de Texto

A classificação de texto evoluiu com o avanço dos LLMs. Diferentemente das abordagens tradicionais, que utilizavam métodos como *bag-of-words* [67], Termo Frequência-Frequência Inversa do Documento, do inglês Term Frequency-Inverse Document Frequency (TF-IDF) [92] ou *embeddings fixos* [38], as técnicas permitem a classificação com maior riqueza contextual e eficiência. O método *bag-of-words* representa

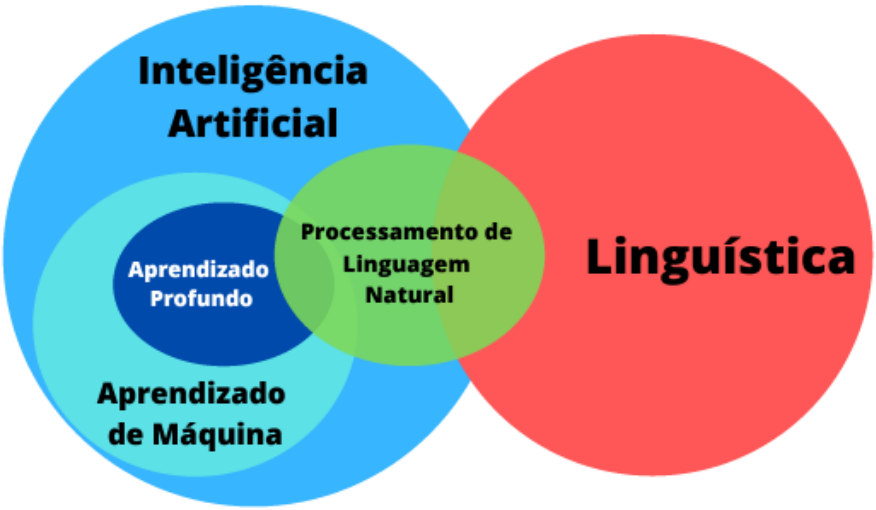

Figura 2.1: *Relacionamento do PLN com as áreas de IA e linguística (Fonte [62])*

o texto como um conjunto de palavras, ignorando sua ordem ou contexto, enquanto o TF-IDF atribui pesos às palavras com base em sua frequência no texto e em todo o *corpus*, destacando termos mais relevantes. Já os *embeddings* fixos, como Word2Vec [11] ou GloVe [23], geram representações vetoriais estáticas para as palavras, sem considerar variações de significado em diferentes contextos. Modelos como GPT, Large Language Model for Meta AI (LLaMA), Qwen e Mixtral superam essas limitações ao integrar uma série de melhorias que revolucionaram o processo, permitindo categorizar textos com maior precisão e adaptabilidade.

O processo de tokenização em modelos modernos representa uma ruptura com os métodos convencionais. Enquanto os sistemas antigos geralmente dividiam o texto em palavras ou caracteres, os LLMs utilizam tokenizadores baseados em subpalavras, como o Byte Pair Encoding (BPE) [6] ou SentencePiece [36]. Essa técnica segmenta o texto em unidades menores que podem ser palavras completas, subpalavras ou mesmo caracteres, dependendo do contexto. Por exemplo, uma palavra como "classificação" pode ser dividida em sub*tokens* como "class", "ifica" e "ção", o que ajuda o modelo a capturar padrões morfológicos e semânticos de forma eficiente. Esse tipo de tokenização também reduz o vocabulário necessário, tornando o processo computacionalmente mais viável e eficiente.

As técnicas contemporâneas de classificação de texto se beneficiam de estratégias avançadas como *fine-tuning* supervisionado [66] e aprendizado por *prompt* [10]. O *fine-tuning* supervisionado ajusta os pesos do modelo para tarefas específicas, utilizando conjuntos de dados específicos para melhorar o desempenho em um domínio. Já o aprendizado por *prompt* explora o modelo pré-treinado sem a necessidade de alterar seus pesos, otimizando as entradas para que o modelo produza respostas alinhadas à tarefa de classifi-

cação desejada. Além disso, métodos de *zero-shot* [34] e *few-shot learning* [75] permitem que os modelos realizem classificações com poucos ou nenhum exemplo prévio, aproveitando o amplo conhecimento pré-treinado em seus dados.

A utilização de *embeddings* contextuais [55] é outro ponto-chave nas abordagens modernas. Ao contrário dos *embeddings* estáticos como Word2Vec ou GloVe, os *embeddings* contextuais gerados por LLMs são dinâmicos e dependem do contexto da frase. Isso significa que uma palavra como "banco" pode ter diferentes representações dependendo se está sendo usada no contexto de uma instituição financeira ou de um assento em um parque. Essa capacidade de adaptação ao contexto aumenta a precisão das classificações.

2.4 Família dos LLMs

Os LLMs representam uma das tecnologias mais avançadas em PLN. Desenvolvidos com bilhões de parâmetros e treinados em extensas bases de dados textuais, esses modelos são projetados para lidar com tarefas complexas, como compreensão de linguagem, geração textual, tradução, entre outras. Cada família de LLMs, como LLaMA, Mixtral e Qwen, adota abordagens arquitetônicas específicas para otimizar desempenho e eficiência em diferentes cenários de aplicação.

Essas famílias variam em tamanho, capacidade e estratégias de treinamento, incluindo técnicas como *Instruct Tuning*, que ajustam os modelos para interpretar e seguir instruções humanas com precisão. A seguir, serão detalhadas as principais características das famílias de modelos exploradas neste trabalho, com ênfase em suas particularidades e nas aplicações específicas no campo do PLN.

2.4.1 LLaMA

Os LLMs Llama-3.1-8B-Instruct-Turbo [45] e Llama-3.1-70B-Instruct-Turbo [44] fazem parte da família LLaMA, desenvolvida pela Meta AI. Esses modelos compartilham a mesma arquitetura e princípios fundamentais, porém diferem em seus tamanhos e capacidades. A versão de 8 bilhões de parâmetros oferece um equilíbrio entre performance e eficiência computacional, enquanto a versão de 70 bilhões de parâmetros é projetada para tarefas mais complexas e de maior escala.

O Llama-3.2-3B-Instruct-Turbo [40], com 3 bilhões de parâmetros, foi projetado para maximizar a eficiência computacional ao lidar com longas sequências de texto, como diálogos multilíngues e tarefas de geração textual. Posicionado entre os modelos de 8B e 70B em termos de tamanho, ele apresenta um desempenho adequado para aplicações que exigem recursos computacionais mais modestos, mas que ainda demandam precisão em cenários complexos, como a análise e a representação de documentos jurídicos.

Ambos os modelos utilizam a arquitetura de transformador, conhecida por sua eficácia em modelar relações sequenciais em textos. Eles foram treinados com uma grande quantidade de dados textuais, abrangendo múltiplos domínios e idiomas, o que os torna adequados para uma ampla gama de aplicações de PLN, incluindo a classificação de textos e compreensão de instruções [81].

2.4.2 Mixtral

Os modelos Mixtral-7B-Instruct-v0.3 [47], Mixtral-8x7B-Instruct-v0.1 [49] e Mixtral-8x22B-Instruct-v0.1 [48] fazem parte da família Mixtral, desenvolvida para lidar com tarefas de PLN em grande escala. Esses modelos compartilham a mesma base arquitetônica, mas diferem em termos de parâmetros e capacidades de inferência. O Mixtral-7B é uma versão compacta, com foco em eficiência e desempenho em tarefas menores, enquanto os modelos Mixtral-8x7B e Mixtral-8x22B foram projetados para cenários de alta demanda, com capacidade de processar grandes volumes de dados e tarefas mais complexas.

O modelo Mixtral-7B-Instruct-v0.3 é otimizado para eficiência, proporcionando um equilíbrio entre precisão e uso de recursos computacionais. Ele é indicado para cenários em que o custo de processamento é uma preocupação e onde tarefas de tamanho médio precisam ser resolvidas com agilidade. Por outro lado, o Mixtral-8x7B-Instruct-v0.1 e o Mixtral-8x22B-Instruct-v0.1 oferecem capacidades mais amplas, com o Mixtral-8x22B sendo adequado para tarefas de alta complexidade e para grandes volumes de dados, permitindo a representação simultânea de textos extensos com precisão elevada. O modelo Mixtral-8x22B, por exemplo, é capaz de escalar suas operações para lidar com milhões de *tokens* em um ambiente otimizado, permitindo maior cobertura de dados.

2.4.3 Qwen

O Qwen2 [68] é uma evolução da família dos LLMs Qwen. Esses modelos variam de 0,5 a 72 bilhões de parâmetros, incluindo uma versão com *Mixture-of-Experts*, sendo uma arquitetura em aprendizado de máquina que combina o uso de múltiplos "especialistas" (modelos ou partes de um modelo) para resolver tarefas complexas de forma eficiente, o que lhes permite se destacar em uma ampla variedade de tarefas de PLN. Comparado a modelos anteriores, como o Qwen1.5, o Qwen2 apresentou desempenho superior em diversos *benchmarks* que avaliam a compreensão de linguagem, geração de texto, capacidade multilíngue, raciocínio matemático, entre outras habilidades. Além disso, o Qwen2-72B demonstrou competitividade em relação a modelos proprietários de ponta, oferecendo uma capacidade de representação em tarefas que exigem um entendimento profundo de grandes volumes de dados textuais. Este modelo é uma escolha

robusta para cenários que demandam a representação de entradas extensas, como análises jurídicas ou documentos técnicos, mantendo eficiência e precisão mesmo com textos de grande complexidade.

2.4.4 *Instruct-Tuning*

Os modelos de grandes famílias de LLMs utilizam a técnica de *Instruct Tuning*, que ajusta os modelos para seguirem instruções humanas com maior precisão. Essa abordagem permite que os LLMs sejam treinados com exemplos específicos de instruções anotadas, o que aprimora a capacidade dos modelos de interpretar e gerar respostas de forma alinhada às expectativas dos usuários [70]. O treinamento adicional para compreender e executar comandos torna esses modelos ideais para aplicações interativas, como assistentes virtuais e sistemas de resposta automática.

Com essa técnica, os LLMs são otimizados para interpretar e responder a perguntas baseadas em exemplos de instruções previamente definidas, garantindo que consigam seguir comandos textuais complexos de maneira eficiente. Isso os torna apropriados para muitas aplicações, que vão desde sistemas de suporte ao cliente até tarefas de representação de linguagem em escala industrial.

2.5 Trabalhos Correlatos

Nesta seção, os trabalhos correlatos foram organizados em três grupos para facilitar a compreensão e análise dos avanços na classificação de textos com o uso de LLMs. Primeiro, foram analisados os trabalhos relacionados à classificação de documentos textuais utilizando LLMs, que incluem abordagens gerais e multidomínio para classificação de textos, com destaque para técnicas e métodos amplamente aplicáveis. Em seguida, foram explorados os estudos de classificação de documentos jurídicos, que abordam a classificação de textos específicos dessa área, como petições, decisões judiciais e processos, sem necessariamente fazer uso direto de LLMs. Por fim, foram apresentados os trabalhos de classificação de documentos jurídicos utilizando LLMs, que destacam soluções para lidar com as particularidades dos textos jurídicos, demonstrando a capacidade de LLMs em automatizar e otimizar tarefas nesse setor.

2.5.1 Classificação de Documentos Textuais utilizando Grandes Modelos de Linguagem

A classificação de documentos textuais utilizando LLMs representa um avanço no campo do PLN. Esses modelos, como GPT, BERT e LLaMA, são capazes de capturar

nuances semânticas e contextuais de textos, permitindo categorizar documentos de maneira eficiente e precisa. Aplicações incluem a organização de grandes volumes de dados textuais em categorias predefinidas, como temas, sentimentos ou tópicos, com impacto direto em áreas como negócios, saúde, educação e jurídico. A utilização de LLMs potencializa o desempenho em tarefas complexas de classificação, superando técnicas tradicionais ao lidar com contextos variados e textos mais longos.

O método Clue And Reasoning Prompting (CARP) [76], foi introduzido para lidar com fenômenos linguísticos complexos em tarefas de classificação de textos. Técnicas como o ajuste fino [39] em múltiplas etapas do LLaMA mostraram-se eficazes em *pipelines* de recuperação de texto, destacando a flexibilidade e o poder desses modelos em tarefas de classificação complexas.

O artigo [20] examina a evolução da classificação de texto no PLN, com foco em modelos baseados em transformadores, incluindo LLMs. Ele aborda a ampliação das aplicações, indo além de entradas textuais para dados multimodais, e analisa o desempenho de modelos em diferentes *benchmarks*. Destaca limitações como custos e acessibilidade, questionando a ideia de superioridade universal dos LLMs. O estudo também explora implicações éticas e sociais, reforçando a necessidade de uma abordagem consciente e criteriosa no uso dessas tecnologias.

A abordagem discutida [73] trata da aplicação de LLMs em tarefas de classificação, destacando a lacuna existente em estudos focados nesse potencial. Para isso, propõe um *framework* para investigar o *fine-tuning* de LLMs em abordagens baseadas em geração e codificação, aplicando-o na tarefa de Editing Intent Classification (EIC). O estudo realiza comparações extensivas entre modelos e métodos de treinamento, revelando novos *insights*. Além disso, utiliza o modelo EIC de melhor desempenho para criar o Re3-Sci2.0, um conjunto de dados com 94 mil edições anotadas em revisões científicas, permitindo uma análise aprofundada do comportamento humano na escrita acadêmica. O *framework*, modelos e dados foram disponibilizados publicamente.

A análise apresentada [88] demonstra a disponibilidade de conjuntos de dados textuais clínicos públicos, fundamentais para o desenvolvimento de LLMs clínicos. Apesar da existência de 192 conjuntos identificados, menos da metade é acessível livremente, destacando limitações éticas e de privacidade. A maioria dos dados disponíveis concentra-se nas Américas, Europa e Ásia, deixando regiões como África e Oceania subrepresentadas. Os dados atendem principalmente tarefas como reconhecimento de entidades, classificação de texto e extração de eventos, mas enfrentam desafios de diversidade e acessibilidade. O estudo reforça a necessidade de compartilhamento responsável de dados clínicos diversificados para reduzir disparidades e avançar a pesquisa em saúde globalmente.

A pesquisa de [42], explora uma nova metodologia para reduzir os custos de

anotação em tarefas de classificação de texto, integrando anotadores humanos e LLMs em uma estrutura de Aprendizagem Ativa. Ao utilizar técnicas de amostragem de incerteza, a abordagem identifica amostras que precisam de anotação manual e aproveita a saída dos LLMs para automatizar parte do processo. Foram conduzidas avaliações com três conjuntos de dados, sendo o Internet Movie Database (IMDB) para análise de sentimento, *Fake News* para verificação de autenticidade e *Movie Genres* para classificação *multilabel*. Os resultados demonstram uma significativa economia nos custos de anotação, sem comprometer a precisão dos modelos.

O trabalho apresentado em [90], examina a eficácia do *fine- tuning* de LLMs na classificação de textos jurídicos. Foram realizadas comparações entre o DistilBERT pré-treinado e o DistilBERT ajustado com dados do domínio jurídico. O *fine-tuning* mostrou-se vantajoso, aumentando a precisão dos classificadores de texto jurídico. Além disso, dois métodos de avaliação foram aplicados, um que considera o texto completo e outro que utiliza fragmentos de sentenças.

O estudo realizado por [37], propõe uma técnica de pré-processamento de textos utilizando o reconhecimento de NER para preservar a privacidade e melhorar a precisão dos classificadores de texto. Em vez de remover ou ignorar entidades sensíveis, como nomes e endereços, a abordagem substitui essas entidades por suas categorias correspondentes, como “localização” ou “pessoa”. Os resultados experimentais mostram que essa abordagem reduz a dimensionalidade dos dados e aumenta a precisão dos classificadores, enquanto protege a privacidade.

De acordo com [21], foi realizada uma pesquisa abrangente sobre a classificação de texto utilizando modelos baseados em transformadores, incluindo LLMs, destacando sua adequação para uma ampla gama de aplicações, desde análise de sentimentos até *chatbots* de perguntas e respostas. A pesquisa revisa 358 *datasets* de 20 aplicações, propondo uma nova taxonomia que inclui dados multimodais. Embora os LLMs apresentem avanços, o estudo ressalta que sua utilização precisa ser criteriosa, uma vez que eles nem sempre são superiores em termos de precisão. Além disso, são levantadas questões sobre custo, segurança e implicações éticas, como *viés* e direitos autorais. O artigo reforça que, dado o potencial dos transformadores, sua implementação em aplicações do mundo real exige uma abordagem holística e cuidadosa.

2.5.2 Classificação de Documentos Jurídicos

A classificação de documentos jurídicos é uma área de pesquisa relevante, devido ao volume e à complexidade dos textos legais. Métodos, como o BERT e suas variações, têm sido aplicados para lidar com a especificidade da linguagem jurídica e as particularidades dos dados longos e não estruturados encontrados nesses documentos.

Esta subseção descreve trabalhos na área de classificação de documentos jurídicos, utilizando modelos de aprendizado profundo, como o BERT e técnicas tradicionais de PLN.

O trabalho apresentado em [59], propõe uma abordagem de auto-supervisão para a classificação de textos jurídicos utilizando o BERT. Devido à dificuldade de rotular dados jurídicos de forma precisa e às questões de privacidade, o estudo gera textos sintéticos por meio da maximização de ativação. Essa técnica melhora a qualidade e diversidade dos dados, permitindo que o modelo Legal-BERT, uma variação do BERT treinada especificamente em textos jurídicos para capturar melhor o vocabulário e as *nuances* desse domínio, alcance bons resultados com menor quantidade de dados reais. O método é testado em dois conjuntos de dados jurídicos, mostrando eficácia na classificação com menor variabilidade nos resultados.

Um modelo baseado no BERT foi desenvolvido para a classificação de casos jurídicos na Índia em categorias criminais ou civis [78]. Utilizando um conjunto de dados de documentos de processos judiciais, o modelo ajusta parâmetros do BERT-Base para melhorar a compreensão contextual e a distinção entre os tipos de casos. O estudo mostra que o modelo proposto oferece alta precisão, *recall* e *F1-score*, com uma precisão geral de 92.9% e um *F1* de 0.93, indicando o potencial da abordagem para automação na gestão de casos jurídicos.

Segundo [29], foi desenvolvido um motor de recuperação de informações baseado em IA para serviços jurídicos, utilizando técnicas de PLN e Aprendizado de Máquina, do inglês, Machine Learning (ML). A pesquisa foca na extração de informações essenciais, como nomes de juízes e datas de julgamento, a partir de textos jurídicos. O modelo proposto utiliza um sistema de reconhecimento de NER personalizado, combinado com um mecanismo de *fallback*. Esse mecanismo atua como uma estratégia de apoio, utilizando expressões regulares para extrair informações sempre que o método principal NER não conseguir identificar corretamente os dados. Essa abordagem garante maior robustez ao sistema, atingindo uma acurácia agregada de 95%.

Um *benchmark* de Federated Learning (FL) aplicado ao domínio jurídico, denominado FEDLEGAL [94], permite que múltiplos participantes colaborem no treinamento de um modelo sem compartilhar diretamente seus dados sensíveis. O estudo avalia cinco tarefas de PLN jurídico e uma tarefa de privacidade com base em dados de tribunais chineses, mostrando que o FL enfrenta desafios devido à não homogeneidade dos dados reais. A pesquisa também sugere que o FL é uma abordagem promissora para preservar a privacidade em cenários jurídicos reais.

Os autores [30], propõem um método para a classificação de sentenças em documentos jurídicos, organizando-os em 13 segmentos, conhecidos como papéis retóricos. O método combina a classificação sequencial de sentenças com a técnica `SetFit` para

melhorar a precisão da extração de informações em documentos complexos. O modelo atinge uma pontuação *F1* de 0.83, superando o *baseline* de 0.79, mostrando-se eficaz na análise de casos jurídicos em países com sistemas sobrecarregados de processos judiciais, como a Índia.

A proposta apresentada [89] aborda a classificação automatizada de textos jurídicos, utilizando RNCs para capturar hierarquias em dados sequenciais. Com técnicas como *max-pooling* e ativação softmax, o modelo oferece uma abordagem eficiente para categorizar citações jurídicas em múltiplas classes. A pesquisa avança o estado da arte na análise de textos jurídicos, contribuindo para maior precisão e adaptabilidade em aplicações de PLN no domínio legal.

O estudo [65] apresenta o Occlusion-based Hierarchical Explanation-extractor (Ob-HEx), um algoritmo de explicação para modelos hierárquicos usados em textos longos, como documentos jurídicos. Baseado em perturbações na entrada, o Ob-HEx torna os modelos mais interpretáveis, abordando a falta de transparência comum em métodos de caixa-preta. Aplicado a modelos hierárquicos de transformadores treinados em textos jurídicos indianos, o Ob-HEx melhora a confiabilidade e alcança um ganho mínimo de 1 ponto sobre os *benchmarks* anteriores no conjunto de dados International Language Data Collection Expert (ILDC-Expert).

2.5.3 Classificação de Documentos Jurídicos utilizando Grandes Modelos de Linguagem

A classificação de documentos jurídicos utilizando LLMs tem se destacado como uma para lidar com o grande volume e complexidade dos textos legais. Esses modelos são capazes de compreender a linguagem técnica e formal dos documentos jurídicos, como petições, sentenças e contratos, permitindo sua organização e categorização de forma automática e eficiente. Com a capacidade de capturar contextos e padrões específicos do domínio jurídico, os LLMs podem ser ajustados para atender às demandas dessa área, oferecendo suporte para análise, extração de informações relevantes e automação de processos. Essa abordagem tem o potencial de otimizar o trabalho jurídico, reduzindo o tempo gasto em tarefas manuais e aumentando a precisão na gestão de informações legais.

Os LLMs têm mostrado grande potencial na classificação de textos jurídicos, facilitando a automação e a análise de documentos legais complexos; um exemplo notável é o SaulLM-7B [12], um modelo desenvolvido especificamente para o domínio jurídico.

O artigo [4] apresenta o LegalLens, um estudo focado na identificação de violações legais em dados textuais não estruturados e na associação dessas violações com indivíduos potencialmente afetados. Foram criados dois conjuntos de dados validados por

especialistas no domínio. A pesquisa utilizou modelos BERT ajustados e experimentos *few-shot* com LLMs, obtendo um *F1-score* de 62.69% na identificação de violações e 81.02% na associação com vítimas. Os autores disponibilizaram publicamente os *datasets* e o código, incentivando avanços em PLN no contexto jurídico.

Para simplificar operações com LLMs, um *toolkit* denominado Sketch [28], inclui esquemas de descrição de tarefas e *templates* de *prompts* para tarefas de PLN, além de um processo interativo para estruturar saídas de LLMs. O *toolkit* também disponibiliza um *dataset* de código aberto para controle de formato de saída e um modelo baseado no LLaMA3-8B-Instruct. A iniciativa visa facilitar o uso de LLMs em várias aplicações, permitindo uma abordagem "*plug-and-play*". Todos os componentes serão gradualmente disponibilizados em código aberto.

Um *benchmark* foi proposto em [50] para avaliar vários modelos de linguagem em quatro dimensões, o processamento de documentos longos (até 50K *tokens*), uso de conhecimento específico do domínio jurídico, compreensão multilíngue (em cinco idiomas) e multitarefa (como recuperação de informações, geração de visões de tribunal, e tarefas de classificação de texto). Os *datasets* abrangem o sistema jurídico suíço e demonstram que os modelos multilíngues existentes têm dificuldades em lidar com essas tarefas, mesmo após extenso pré-treinamento e ajuste. Todos os recursos do *benchmark* são disponibilizados sob licença CC BY-SA.

O trabalho descrito em [87], os autores analisaram o impacto do ajuste fino de LLMs no contexto jurídico, comparando dois métodos de predição, um baseado em fragmentos de textos e outro em documentos completos. O trabalho de [37], avalia o uso de NER como técnica de pré-processamento para preservar a privacidade e melhorar a precisão da classificação.

Os resultados encontrados [64] propõem o *framework* Multi-stage Encoder-based Supervised with Clustering (MESc) para a previsão de julgamentos em documentos jurídicos extensos e sem estrutura definida. Utilizando LLMs como GPT-Neo e GPT-J, o modelo combina *embeddings* extraídos das últimas camadas com agrupamento não supervisionado e representações inter-partes em camadas de transformadores adicionais. Os resultados, testados em documentos jurídicos de diferentes jurisdições, mostram um ganho mínimo de 2 pontos em desempenho em relação a métodos anteriores, destacando a eficácia do *framework* para classificação de textos legais complexos.

Em [59], propõe uma nova técnica de auto-supervisão para a geração de dados sintéticos em modelos BERT aplicados ao domínio jurídico. O BERT também foi utilizado para classificar casos jurídicos em categorias criminais e civis, ajustando parâmetros para melhorar a acurácia do modelo [78].

2.6 Considerações Finais

Muitos estudos recentes investigam o uso de LLMs para a classificação de textos jurídicos, com diferentes abordagens para lidar com a complexidade e variabilidade desses documentos. Entretanto, nenhum destes trabalhos propõem um *pipeline* para classificação automática de documentos jurídicos, com características dos documentos textuais da DPE-GO.

Embora [87] trate de uma abordagem semelhante ao nosso foco no uso de LLMs para classificação, nosso trabalho se diferencia por explorar múltiplas estratégias de representação de documentos, como resumos, centroides, descrições e abordagem direta, ampliando o leque de técnicas avaliadas.

O trabalho de [59] possui similaridade com o nosso, pois utiliza modelos pré-treinados para tarefas de classificação jurídica, mas a diferença está na ênfase do nosso estudo nos métodos de representação do texto original (por meio de resumos e centroides), ao invés de gerar novos dados sintéticos.

Diferentemente do estudo de [78], nossa pesquisa se estende para além da categorização binária (criminal/civil), abordando um conjunto mais amplo de categorias jurídicas e introduzindo métodos diversos de representação de texto.

Apesar das contribuições de [37] nas técnicas de pré-processamento, nossa abordagem foca em otimizar a estrutura do texto jurídico completo, enquanto o estudo se concentra na anonimização de dados sensíveis por meio de NER.

Os autores [22] também avaliam a eficiência das abordagens de classificação, mas nosso foco principal está na avaliação da precisão e do custo computacional relacionado aos métodos de representação de texto jurídico, ao invés de questões ambientais.

Nosso trabalho propõe uma solução para a classificação de documentos na literatura, com um *pipeline* desenvolvido para documentos jurídicos, que busca avaliar diferentes técnicas de representação de textos jurídicos (como resumo, centroide, classificação baseada no texto original e descrições) aplicadas à tarefa de classificação. Esta solução foi concebida no contexto da DPE-GO, um importante órgão jurídico público de grande relevância no estado de Goiás, em Goiânia. Com ela, destacamos o impacto de cada técnica no desempenho dos LLMs para a classificação de documentos jurídicos longos e complexos, característicos da DPE-GO, ampliando a compreensão e aplicabilidade dessas tecnologias em cenários reais.

CAPÍTULO 3

Pipeline para a Classificação Automatizada de Documentos Jurídicos por Meio de LLMs

A Figura 3.1 ilustra o *pipeline* utilizado neste trabalho, desde a coleta de dados até a classificação dos documentos jurídicos. Após a etapa de pré-processamento, que inclui a remoção de ruídos e a anonimização de dados sensíveis, os documentos são submetidos a uma análise estatística detalhada. O *corpus* é, então, dividido para garantir uma distribuição equilibrada entre os conjuntos de avaliação e classificação.

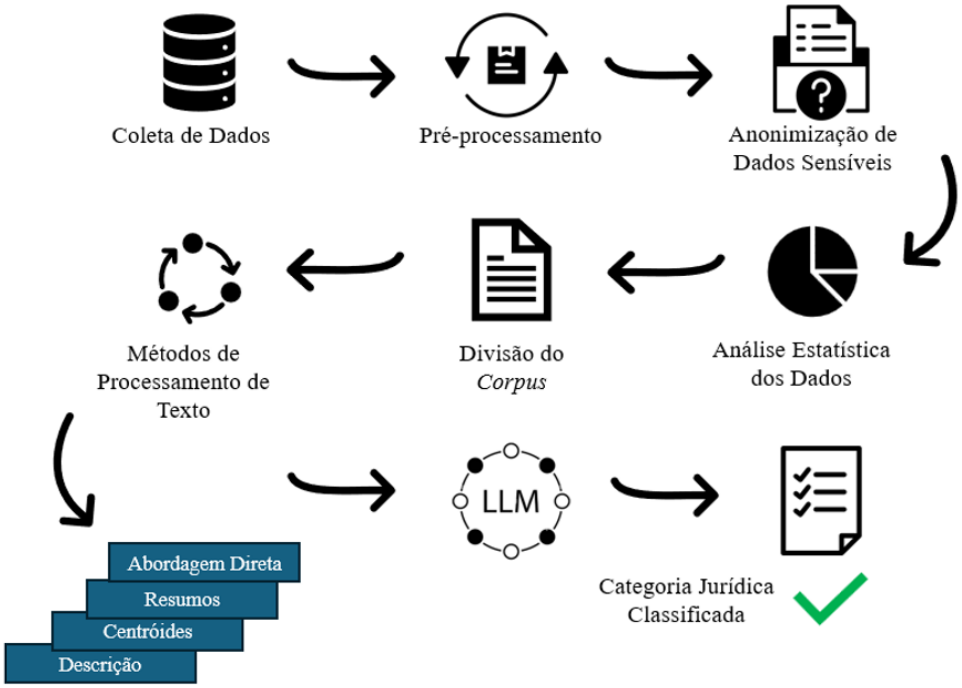

Figura 3.1: *Processo do pipeline utilizado, desde a coleta e anonimização dos dados até a classificação jurídica final, passando por etapas de pré-processamento, análise estatística e aplicação de métodos de representação de texto com os LLM (Fonte: Elaborado pelo Autor)*

A representação de texto é uma etapa importante no *pipeline* que impacta o sucesso das tarefas de classificação de documentos jurídicos. Este estudo explora,

de forma geral, o impacto de diferentes métodos de representação de documentos no desempenho dos LLMs na classificação de textos jurídicos. Para validar a abordagem proposta, foi aplicado um estudo de caso na DPE-GO, onde o *pipeline* desenvolvido foi utilizado para avaliar e aprimorar a classificação de documentos jurídicos dessa instituição. Reconhecendo que a extensão dos documentos jurídicos pode representar um desafio para os LLMs, exploramos quatro métodos de representação diferentes para otimizar a entrada, incluindo a classificação baseada no texto original, resumos, centroides e descrição. Cada abordagem emprega uma estratégia para estruturar o texto de entrada, com o objetivo de aprimorar a capacidade do modelo de interpretar e classificar com precisão documentos jurídicos.

Os documentos são processados utilizando os quatro métodos na otimização da representação textual. Esses métodos permitem criar representações de texto mais adequadas ao entendimento dos LLMs que foram utilizados neste estudo. Cada um desses métodos de representação textual será detalhado nas seções deste capítulo, destacando suas contribuições para a melhoria dos resultados da classificação dos documentos jurídicos.

A Equação 3-1 resume os diferentes métodos de representação de texto aplicados no *pipeline* de classificação de documentos jurídicos. Nela, a função $f_{\text{classificacao}}$ representa o algoritmo de classificação responsável por mapear o texto de entrada T para uma das categorias jurídicas disponíveis. A escolha do método M de representação de texto é importante para a eficácia dessa classificação, uma vez que cada abordagem modifica a representação do texto original de maneira diferente.

$$\text{Categoria} = f_{\text{classificacao}}(T, M) \tag{3-1}$$

Onde:

- $f_{\text{classificacao}}$ é a função que realiza a classificação do texto.
- T é o texto completo da petição.
- M é o método aplicado para a classificação, que varia conforme a abordagem:
  - **Classificação do texto original**: $M = \emptyset$, onde o texto completo T é utilizado diretamente na classificação.
  - **Resumo**: $M = S(T)$, onde $S(T)$ é o resumo gerado a partir do texto T.
  - **Centroides**: $M = C_i$, onde C_i é o centroide da categoria i.
  - **Descrições**: $M = D_i$, onde D_i é a descrição da categoria i.

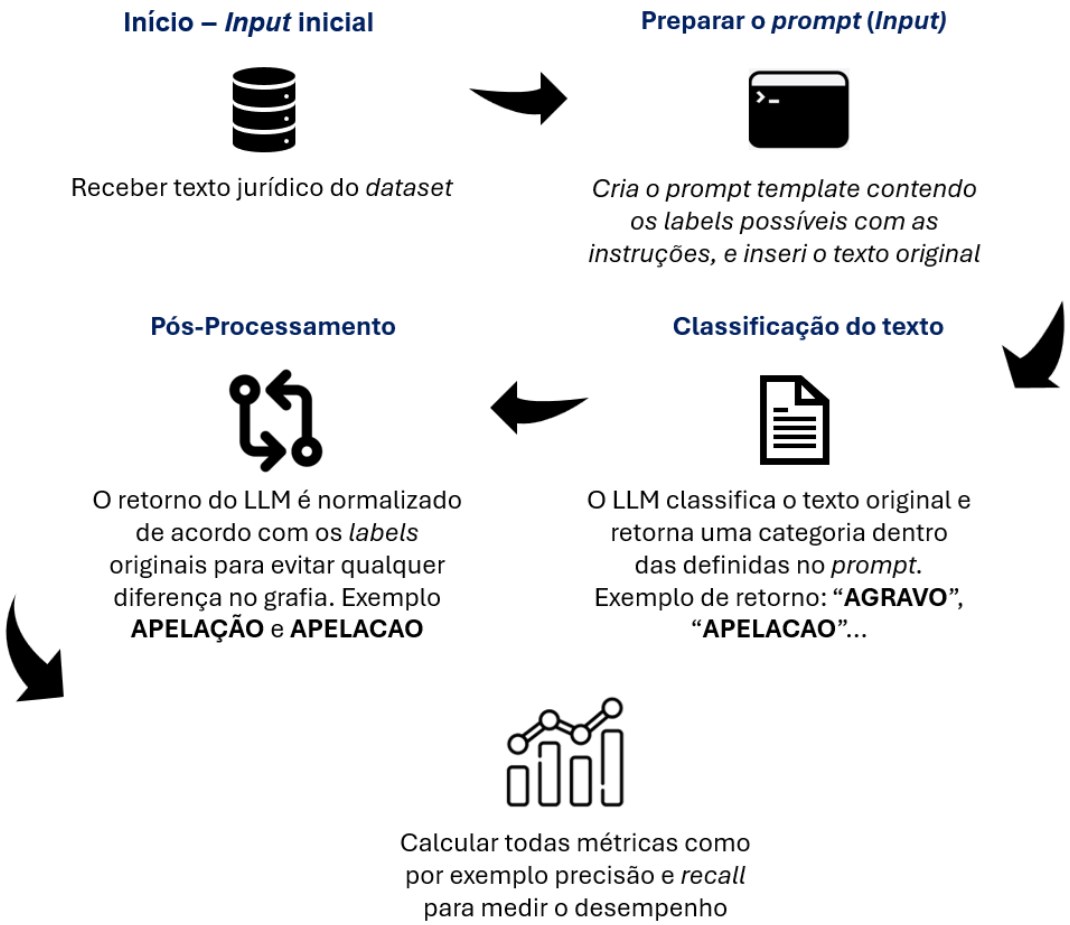

Figura 3.2: *Ilustração do método de classificação baseado no texto original, com etapas desde a entrada do texto completo até a classificação jurídica, passando pela elaboração do prompt, classificação pelo modelo e o cálculo das métricas de avaliação (Fonte: elaborado pelo autor)*

3.1 Método de Classificação Baseado no Texto Original

A Figura 3.2 apresenta o fluxo do **Método de Classificação Baseado no Texto Original**. O processo começa com a entrada do texto completo, sem qualquer simplificação ou condensação prévia. Um *prompt* A é elaborado para instruir o modelo de linguagem a analisar o documento jurídico em sua totalidade e categorizar suas características em uma das categorias jurídicas predefinidas. O modelo processa o texto integralmente, levando em consideração todos os detalhes e *nuances* presentes no conteúdo. Durante o processo, é realizada uma análise do impacto da quantidade de *tokens*, já que textos longos podem aumentar o custo computacional e o tempo de processamento. Os detalhes desse método, incluindo suas vantagens e limitações, serão discutidos a seguir.

A classificação baseada no texto original [85] se diferencia de métodos que utilizam resumos, centroides ou descrições. A integridade dos textos originais é preservada, o que significa que cada documento jurídico é apresentado ao modelo de linguagem em sua

totalidade, sem qualquer simplificação ou resumo do conteúdo. A premissa fundamental dessa abordagem é que, ao fornecer ao modelo o máximo de informações possível, aumenta-se a probabilidade de uma classificação correta, especificamente em casos onde detalhes específicos podem fazer a diferença. O modelo tem, portanto, a oportunidade de considerar todas as *nuances* e características intrínsecas dos textos completos, incluindo informações contextuais e específicas.

No *pipeline* desenvolvido, o processo começa com o carregamento do *dataset* de textos jurídicos já pré-processados, conforme indicado na Figura 4.4. Para realizar a classificação, utiliza-se um *prompt* A desenvolvido, que orienta o modelo a categorizar o texto em uma das categorias jurídicas predefinidas, como "AGRAVO", "APELACAO", ou "PROGRESSAO-DE-REGIME", entre outras categorias. O *prompt* é uma ferramenta importante de "*prompt engineering*", projetada para guiar o modelo a focar nas características mais relevantes do texto.

Um exemplo típico de texto jurídico seria: "O réu solicita agravo devido à decisão desfavorável em primeira instância, alegando ausência de provas conclusivas sobre a autoria do crime" (*input*). O *prompt* seria uma solicitação para classificar o texto em uma das seguintes categorias: "AGRAVO", "APELACAO", e assim por diante. O modelo retorna apenas a categoria, que neste caso seria "AGRAVO" (*output*).

Após a classificação, ocorre o pós-processamento, que normaliza os resultados retornados pelo modelo e os compara com os rótulos reais do *dataset*. Essa normalização é importante para evitar discrepâncias devido a variações de grafia ou formatação, como no exemplo "APELAÇÃO" *versus* "APELACAO". Essa etapa inclui também a identificação da melhor correspondência entre a saída do modelo e os rótulos esperados, utilizando métricas como Similaridade de Jaccard para capturar variações sutis.

Um ponto relevante observado foi o impacto do número de *tokens* processados. A média de *tokens* processados por texto foi de 1.613,79. Isso implica em maior custo computacional e tempo de processamento mais longo em comparação com abordagens que utilizam resumos. Embora o modelo tenha acesso a uma riqueza de informações, nem sempre isso resulta em um melhor desempenho. Em alguns casos, a grande quantidade de informações pode sobrecarregar o modelo, dificultando a identificação dos traços mais importantes para a classificação e comprometendo a precisão.

A principal vantagem da classificação baseada no texto original é o uso completo dos textos originais, permitindo ao modelo acessar todos os detalhes e *nuances* do documento. Esse método é útil em situações onde detalhes específicos, que podem ser omitidos em resumos ou centroides, são importantes para determinar a categoria jurídica correta. Entretanto, a desvantagem dessa abordagem está no aumento do custo computacional e no tempo de processamento, devido ao maior número de *tokens* que o modelo precisa processar. Além disso, a grande quantidade de informações pode, em

algumas situações, sobrecarregar o modelo, dificultando a identificação dos traços mais importantes para a classificação e, assim, comprometendo a precisão dos resultados.

Ademais, outro aspecto a ser considerado é a eficiência em termos de acurácia *versus* custo computacional. Embora a abordagem direta forneça ao modelo uma riqueza de informações, nem sempre isso significa uma classificação mais precisa. Em algumas categorias jurídicas, os textos são muito extensos e detalhados, o que pode desviar o foco do modelo, levando a classificações incorretas. Esse fenômeno reforça a ideia de que, em certos casos, menos informações mais focadas podem ser mais eficazes para o modelo, como ocorre nas abordagens baseadas em resumos ou centroides.

3.2 Método de Classificação Baseado em Resumos

A Figura 3.3 ilustra o fluxo do **Método de Classificação Baseado em Resumos**. O processo inicia com a vetorização dos textos jurídicos utilizando a técnica TF-IDF, que converte os textos em vetores numéricos, destacando as palavras mais relevantes de cada documento. Em seguida, são calculados os centroides para cada categoria jurídica, permitindo a identificação dos documentos mais próximos a esses centroides, considerados os mais representativos de suas categorias. Esses documentos servem como base para a geração de resumos, utilizando um *prompt* A.2 para orientar o modelo de linguagem a criar textos compactos, limitados a 200 palavras, que preservem as características centrais de cada categoria jurídica. Esses resumos são então utilizados como referência no processo de classificação jurídica. Por fim, uma análise avalia a eficácia do método, equilibrando a eficiência computacional proporcionada pelos resumos com o risco potencial de perda de informações relevantes. Os detalhes desse método e suas implicações serão discutidos a seguir.

O método de classificação baseado em resumos [13] utiliza técnicas de vetorização, *clustering* e PLN para otimizar a análise de textos jurídicos. O processo começa com a vetorização dos textos jurídicos utilizando a técnica de TF-IDF. Essa técnica transforma os textos em vetores numéricos, atribuindo pesos às palavras com base em sua relevância dentro de cada documento e em relação ao restante do *corpus*. A escolha do TF-IDF foi estratégica, pois permite identificar palavras que são características de cada documento, destacando os padrões centrais necessários para a classificação.

Após a vetorização, são formados *clusters* para cada categoria jurídica. Para isso, calcula-se o centroide de cada *cluster*, que representa a média dos vetores TF-IDF dos textos associados a uma categoria específica. O centroide funciona como um ponto matemático que captura as características centrais da categoria. Em seguida, a distância cosseno, uma métrica que avalia a similaridade entre dois vetores ao medir o ângulo entre eles, é usada para identificar o texto mais próximo ao centróide, considerado o mais

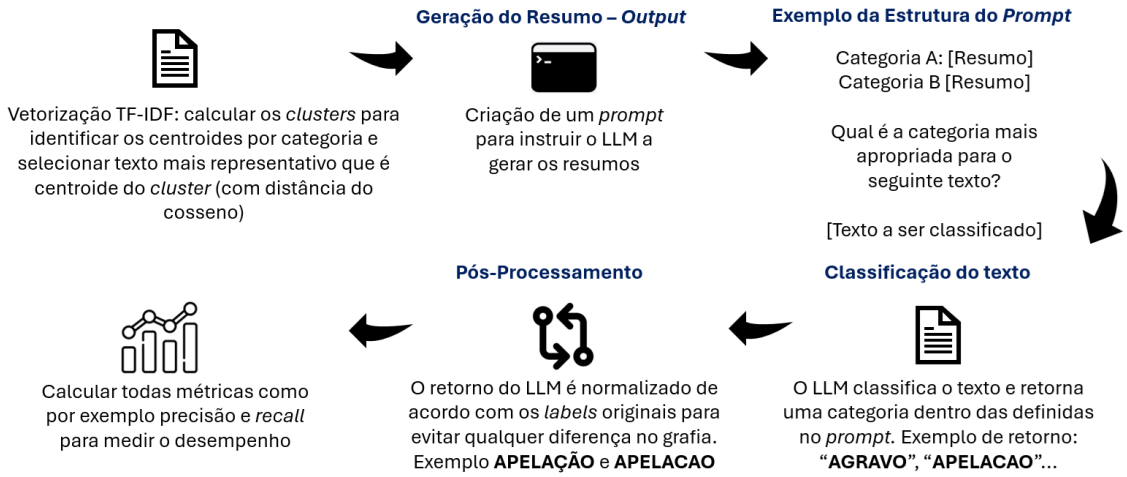

Figura 3.3: *Ilustração do método de classificação baseado em resumos, mostrando etapas desde a vetorização dos textos jurídicos até a análise de eficiência e nuances (Fonte: elaborado pelo autor)*

representativo da categoria. Este texto é então utilizado como base para a geração de resumos, pois encapsula as informações mais relevantes da classe.

Com os textos representativos selecionados, são gerados resumos utilizando um LLM. Essa etapa é guiada por um *prompt* A.2, projetado para extrair a essência do documento em até 200 palavras [5]. Os resumos têm o objetivo de preservar os conceitos centrais do texto original, ignorando detalhes como nomes e números específicos, de forma a criar entradas compactas e informativas para o modelo. O limite de palavras foi escolhido para equilibrar a representatividade dos resumos e a eficiência computacional, reduzindo o número de *tokens* processados.

O próximo passo no *pipeline* envolve a classificação de textos novos. O texto novo não é diretamente comparado com os resumos previamente gerados. Em vez disso, o resumo representativo de cada categoria jurídica é usado como referência para a classificação. No *prompt* classificação A, os resumos são incluídos como parte das instruções ao modelo, e o texto novo é analisado para determinar a categoria mais apropriada. Assim, o modelo utiliza os resumos como guias indiretos para compreender as características principais de cada categoria e fazer a predição.

Após o modelo de linguagem prever a categoria jurídica de um texto novo, os resultados passam por uma etapa de pós-processamento. Essa etapa inclui a normalização dos rótulos retornados pelo modelo para padronizar formatos, garantindo que as previsões sejam comparáveis aos rótulos reais. Por exemplo, categorias como "APELAÇÃO" e "APELACAO" são tratadas como equivalentes.

Embora a abordagem baseada em resumos seja eficiente, ela apresenta uma limitação inerente, como a possível perda de detalhes importantes durante a geração

dos resumos. Esses detalhes podem ser importantes em textos jurídicos mais complexos, impactando negativamente a precisão em certos casos. No entanto, o compromisso entre eficiência computacional e representatividade textual torna o método promissor em cenários onde o custo e o tempo são fatores determinantes.

Por fim, a escolha entre trabalhar com textos completos ou resumos depende do caso de uso e dos objetivos do sistema. Textos completos fornecem uma visão mais detalhada, mas com maior custo computacional. Resumos, por outro lado, oferecem uma alternativa eficiente que mantém o desempenho robusto em um formato otimizado. O *pipeline* descrito representa uma integração inteligente dessas técnicas, destacando-se pela sua eficiência e adaptabilidade em classificações jurídicas.

3.3 Método de Classificação Baseado em Centroides

A Figura 3.4 apresenta o fluxo do **Método de Classificação Baseado em Centroides**. O processo começa com a vetorização dos textos jurídicos utilizando a técnica TF-IDF, seguida pelo cálculo de um centroide para cada categoria jurídica. O documento mais próximo desse centroide é identificado como o mais representativo da categoria, atuando como uma referência central para a análise de novos textos. Através de um *prompt* A, o modelo de linguagem utiliza o documento centroide como referência semântica para interpretar os novos documentos e determinar a categoria jurídica mais apropriada. Os detalhes desse método e suas implicações serão discutidos a seguir.

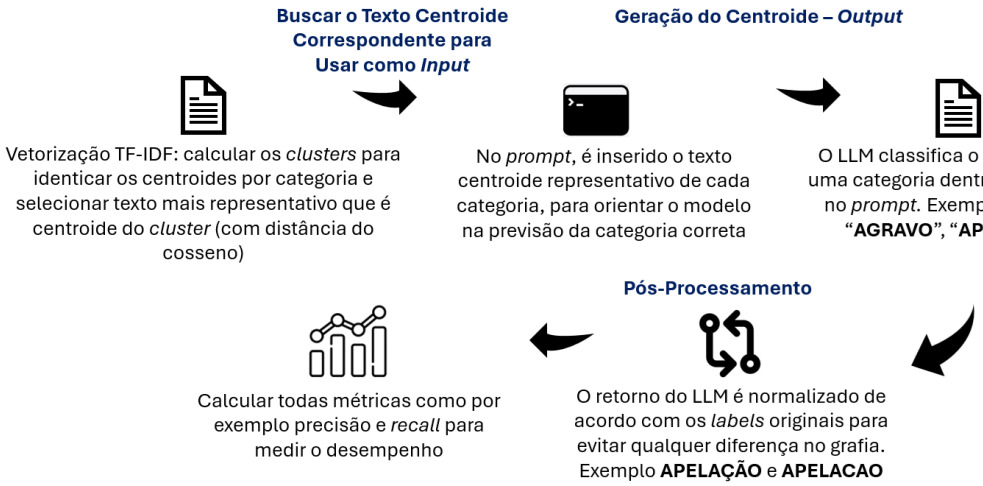

Figura 3.4: *Ilustração do método de classificação baseado em centroides, mostrando as etapas desde a vetorização dos textos jurídicos até a análise de precisão e limitações (Fonte: elaborado pelo autor)*

O método de classificação baseado em centroides [9] utiliza a vetorização dos textos jurídicos com a técnica TF-IDF, semelhante à abordagem de resumos. Após a

vetorização, calcula-se o vetor centroide para cada categoria jurídica no *dataset*. O centroide representa a média dos vetores TF-IDF de todos os textos associados a uma categoria, funcionando como uma "representação central" que captura as características mais comuns entre os documentos daquela classe.

Com base nos centroides, identifica-se o texto mais representativo da categoria, denominado documento centroide. Este é determinado como o texto cujo vetor TF-IDF apresenta a menor distância cosseno em relação ao vetor centroide A.4. Esse texto é utilizado diretamente como referência para a classificação, eliminando a necessidade de geração de resumos adicionais, como ocorre na abordagem anterior. A utilização do documento centroide garante que o modelo de linguagem trabalhe com uma representação clara e concisa de cada categoria jurídica.

Na etapa de classificação, um novo texto é analisado utilizando o documento centroide correspondente à sua categoria real, previamente identificado. O modelo de linguagem é instruído por meio de um *prompt* A, que inclui diretamente o texto do centroide como subsídio principal e solicita a classificação em uma das categorias predefinidas. Essa abordagem reduz a variabilidade textual que pode surgir ao trabalhar diretamente com textos completos, permitindo que o modelo se concentre nos padrões linguísticos centrais da categoria.

Embora eficiente, a abordagem baseada em centroides possui limitações. Em categorias jurídicas com alta diversidade textual, o documento centroide pode não capturar todas as *nuances* importantes para uma classificação precisa. Isso ocorre porque o centroide é uma média estatística dos textos da categoria, o que pode levar a uma subrepresentação de variações entre os textos individuais. Assim, em casos de categorias heterogêneas, a precisão do método pode ser impactada negativamente.

Apesar dessas limitações, o método de centroides apresenta vantagens claras em relação a custos computacionais e simplicidade operacional. Ao utilizar diretamente os textos mais representativos, ele elimina etapas como a geração de resumos e permite ao modelo de linguagem trabalhar com informações já otimizadas. Essa abordagem é útil quando se busca eficiência em contextos onde as categorias jurídicas apresentam padrões textuais bem definidos e consistentes.

3.3.1 Diferença para o Método de Resumos

Nesta subseção, exploramos as diferenças entre o método de resumos e o método de centroides, destacando como esses processos se distinguem em termos de complexidade e tempo de processamento ao lidar com diferentes volumes de textos.

Se considerarmos uma categoria com 50 textos e outra com 300 textos, os processos apresentam diferenças em termos de tempo e complexidade. No método de

resumos, o processo começa com a vetorização dos textos utilizando TF-IDF. Em seguida, calcula-se o centroide, que é a média dos vetores TF-IDF de todos os textos da categoria. Após isso, identifica-se o texto mais próximo ao centroide usando a distância cosseno. Esse texto, chamado de texto representativo, é então utilizado como base para a geração de resumos.

A geração de resumos utiliza um LLM para criar uma versão compacta e informativa do texto representativo. O LLM é guiado por um *prompt* projetado A.2, que instrui o modelo a sintetizar o texto em até 200 palavras, mantendo os conceitos centrais e descartando detalhes irrelevantes, como nomes próprios e números específicos. O processo de gerar resumos é mais demorado quanto maior for o número de textos na categoria, pois o modelo precisa processar o texto representativo para garantir que os resumos reflitam adequadamente as características da categoria jurídica.

Já no método de centroides, o centroide também é calculado como a média dos vetores TF-IDF, e o texto mais próximo a ele é identificado. No entanto, ao contrário do método de resumos, esse texto é utilizado em sua versão completa, sem passar pela etapa de geração de resumos. Isso elimina a necessidade de processamento adicional pelo LLM, reduzindo o tempo total para preparar o sistema. Em categorias menores, como aquelas com 50 textos, o impacto no tempo é menor, mas em categorias maiores, com 300 textos, o aumento no volume de dados impacta a etapa de cálculo do centroide e a identificação do texto mais próximo.

3.4 Método de Classificação Baseado em Descrição

A Figura 3.5 apresenta o fluxo do **Método de Classificação Baseado em Descrição**. O processo inicia-se com a criação de descrições detalhadas para cada categoria jurídica, as quais são incluídas em um *prompt* A.5 que guia o modelo de linguagem. O modelo processa o texto completo do documento jurídico e utiliza as descrições como referência semântica, permitindo uma classificação mais precisa em categorias complexas. O fluxo também ressalta o impacto dessa abordagem no custo computacional, que é analisado em termos de precisão *versus* eficiência. Os detalhes desse método e suas implicações serão discutidos a seguir.

O método de classificação baseada em descrição [90] envolve a utilização de explicações detalhadas que definem os critérios e características que um documento jurídico deve possuir para ser classificado em uma determinada categoria. Cada categoria jurídica é acompanhada de uma descrição específica, incluída no *prompt* A.5 fornecido ao LLM, para orientar a classificação. O objetivo principal dessa abordagem é fornecer ao modelo uma base de comparação mais informativa, o que pode ser útil em categorias que

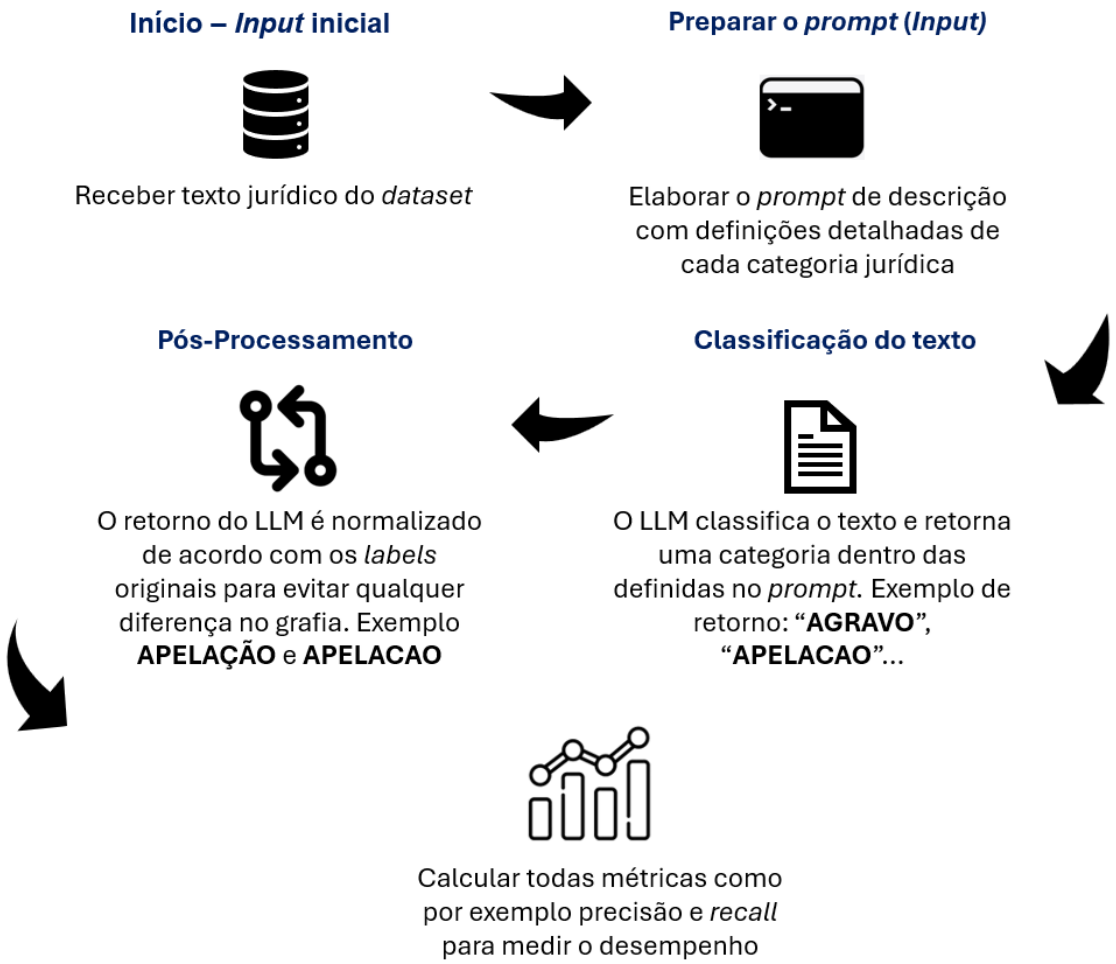

Figura 3.5: *Ilustração do método de classificação baseado em descrições, com etapas que vão da definição de descrições para cada categoria jurídica até a análise de precisão e custo computacional (Fonte: elaborado pelo autor)*

apresentam características sobrepostas ou são difíceis de distinguir sem um entendimento mais profundo dos detalhes jurídicos.

Ao usar a abordagem descritiva, o modelo de linguagem processa o texto completo do documento jurídico e utiliza as descrições das categorias fornecidas como referências para a sua classificação. As descrições contêm explicações concisas das características principais e dos critérios de inclusão de cada categoria, orientando o modelo na identificação dos elementos relevantes presentes no documento. Inicialmente, essas descrições foram elaboradas com o apoio de defensores públicos, que utilizaram seu conhecimento jurídico para definir os componentes centrais de cada categoria. Posteriormente, com a atualização do conjunto de dados, o modelo GPT-4 foi utilizado para expandir e ajustar as descrições às novas exigências e particularidades do *dataset*.

Esse processo de classificação é guiado por um *prompt* específico A.5, que instrui o modelo a focar nas características mais relevantes do texto do documento e associá-las

às definições das categorias jurídicas predefinidas. Diferente de uma "comparação" literal, o modelo analisa o contexto do texto e utiliza as descrições como um guia semântico para melhorar sua acurácia na classificação. Dessa forma, o modelo não realiza uma comparação no sentido tradicional, mas sim utiliza as descrições como uma camada adicional de informação para interpretar e categorizar os documentos.

Uma das principais vantagens dessa abordagem é o fato de que ela permite que o modelo de linguagem tome decisões com base em informações mais detalhadas. Isso pode levar a uma maior precisão de classificação, sobretudo em casos onde as categorias jurídicas são complexas e possuem definições que exigem um entendimento mais profundo. Além disso, essa abordagem é eficaz em contextos onde há categorias cujas características são similares ou ambíguas, dificultando a distinção sem uma compreensão detalhada de seus critérios específicos. Ao fornecer descrições adicionais, o modelo é capaz de considerar *nuances* que, de outra forma, poderiam passar despercebidas.

Entretanto, essa abordagem apresenta também desafios e desvantagens. Um dos maiores desafios está relacionado ao custo computacional. O uso de descrições detalhadas aumenta o número de *tokens* que o modelo precisa processar, o que pode resultar em custos computacionais mais altos e em tempos de processamento mais longos. Além disso, a complexidade adicional das descrições pode, em alguns casos, sobrecarregar o modelo, tornando mais difícil a identificação das características mais relevantes para a classificação.

CAPÍTULO 4

Metodologia

4.1 Coleta de Dados

Os dados fornecidos pela DPE-GO consistem em um *corpus* contendo 3.458 textos de diversos tamanhos, com até 8.000 *tokens*. Esses textos abrangem petições iniciais, casos criminais e processuais, distribuídos em 24 categorias distintas. A geração do *corpus* contou com o apoio dos LLMs. A Tabela 4.1 apresenta a quantidade de textos por categoria no conjunto de dados original. Os dados não foram disponibilizados para testes ou uso por outros autores devido a restrições impostas e políticas internas da DPE-GO.

Categoria	**Número de Textos**
EXTINCAO-DE-PUNIBILIDADE	500
AGRAVO	441
IMPUGNACAO	425
EMBARGOS	304
APELACAO	296
REGISTRO-CIVIL	236
LIVRAMENTO-CONDICIONAL	218
INDULTO-COMUTACAO	170
CUMPRIMENTO-DE-SETENCA	150
OFICIOS	149
INDENIZATORIAS	143
EXCECAO-DE-PRE-EXECUTIVIDADE	98
TRANSFERENCIA-DE-EXECUCAO	85
USUCAPIAO	42
UNIFICACAO-DE-PENAS	39
HABEAS-CORPUS	36
INTIMACAO-NEGATIVA	26
DISSOLUCAO-DE-CONDOMINIO	26
REMICAO-DE-PENA	25
PROGRESSAO-DE-REGIME	19
CONSIGNACAO-EM-PAGAMENTO	18
CONTRARRAZOES-AO-AGRAVO	15
SAIDA-TEMPORARIA	13
ALVARA-JUDICIAL-LIBERACAO-DE-CORPO	12

Tabela 4.1: *Número de textos disponíveis em cada categoria jurídica (Fonte: elaborado pelo autor)*

4.1.1 Análise Estatística

A Figura 4.1 apresenta um histograma da quantidade de *tokens* nos textos originais, gerado a partir do processamento realizado pelo modelo Llama-3.1-8B-Instruct-Turbo. Através dessa visualização, é possível observar uma distribuição assimétrica, concentrando-se em textos com até 2.000 *tokens*, enquanto alguns textos chegam a ultrapassar 8.000 *tokens*. O valor médio de *tokens* por texto é de aproximadamente 1.613,79, indicado pela linha vermelha tracejada, e a mediana, representada pela linha verde tracejada, está em 1.333,50 *tokens*. Isso indica que a maioria dos textos tem menos *tokens* que a média, sugerindo uma distribuição positiva, onde existem alguns textos longos que elevam a média.

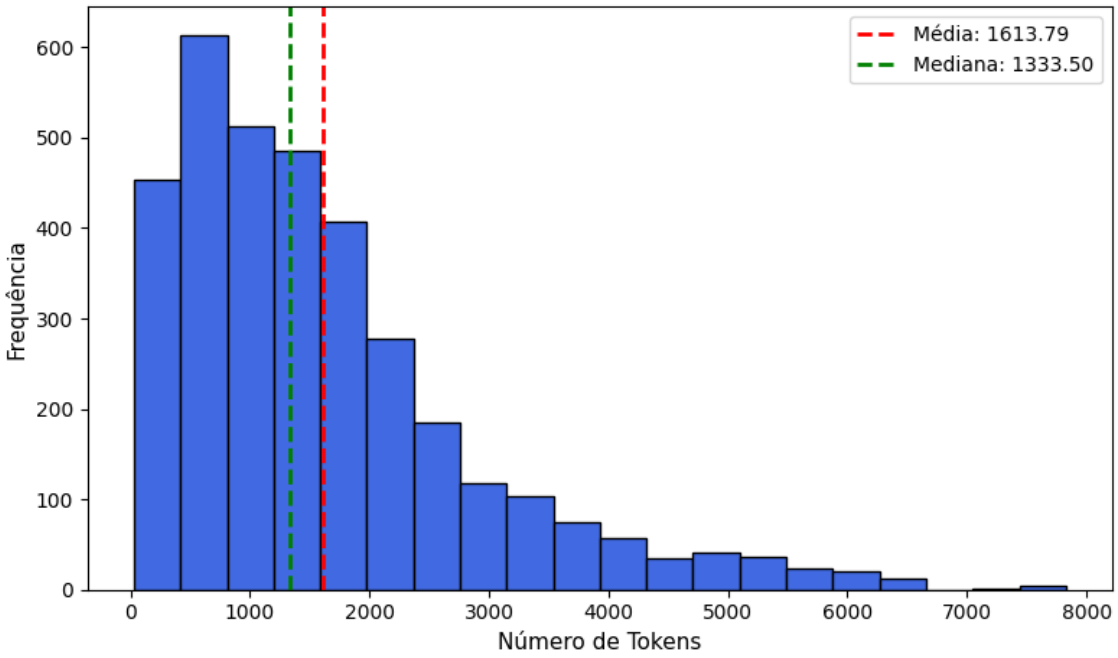

Figura 4.1: *Histograma da quantidade de tokens nos textos originais, com média e mediana indicadas (Fonte: elaborado pelo autor)*

Na Figura 4.2, temos a média de *tokens* por texto, segmentada por cada uma das 24 categorias presentes no *corpus*. As categorias de "INDENIZATORIAS" e "*HABEAS-CORPUS*" destacam-se como as que possuem a maior quantidade média de *tokens*, com mais de 3.000 *tokens* em média. Por outro lado, categorias como "OFICIOS" e "INTIMACAO-NEGATIVA" apresentam uma média inferior, ficando abaixo de 1.000 *tokens*. Essas diferenças podem ser explicadas pela natureza do conteúdo de cada categoria. Por exemplo, petições de indenizações ou *habeas corpus* tendem a conter argumentações mais elaboradas e longas, enquanto ofícios e intimações são documentos mais diretos e sucintos.

Embora o método atual utilize modelos de linguagem para a classificação, a análise do número de *tokens* em cada categoria pode sugerir uma abordagem complementar. Por exemplo, textos mais curtos (menos de 1.000 *tokens*) poderiam ser priorizados para categorias como "OFÍCIOS" e "INTIMAÇÃO-NEGATIVA", enquanto textos mais longos (acima de 3.000 *tokens*) poderiam indicar maior probabilidade de pertencer a categorias como "INDENIZATÓRIAS" ou "HABEAS CORPUS". Essa análise pode ser explorada como um critério adicional ou uma etapa preliminar para otimizar o processo de classificação.

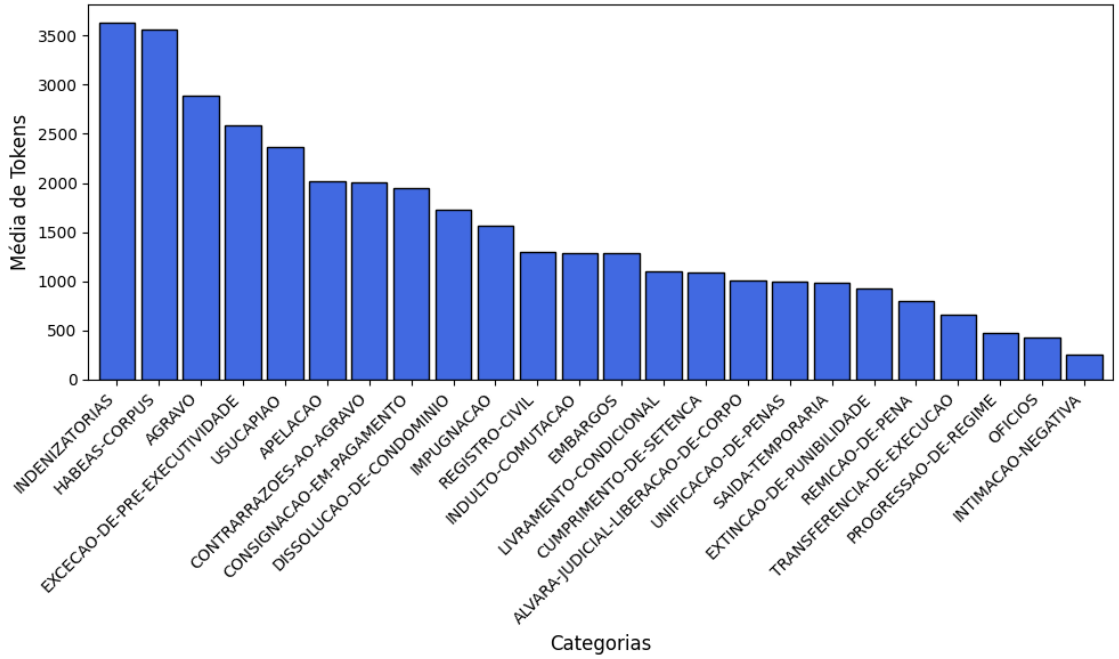

Figura 4.2: *Média de tokens por texto em cada categoria jurídica*
(Fonte: elaborado pelo autor)

A Figura 4.3 oferece uma visão complementar ao mostrar a média de palavras por texto em cada categoria. Observa-se que as categorias com maior quantidade de *tokens* também possuem maior número de palavras, como é o caso das "INDENIZATORIAS" e "*HABEAS-CORPUS*". No entanto, o número médio de palavras por categoria é visivelmente inferior ao número médio de *tokens*, o que reflete a presença de múltiplos *tokens* para palavras compostas, sinais de pontuação e outras formas de construção linguística. Categorias como "OFICIOS" e "INTIMACAO-NEGATIVA" mantêm-se entre aquelas com menor média de palavras, corroborando com as observações de que esses documentos são mais concisos.

Essa análise quantitativa das características dos textos, em termos de *tokens* e palavras, é importante para a escolha de abordagens de pré-processamento e para a construção de modelos de linguagem. A variação no tamanho dos textos, tanto em termos de *tokens* quanto de palavras, reflete a diversidade das categorias e impõe desafios

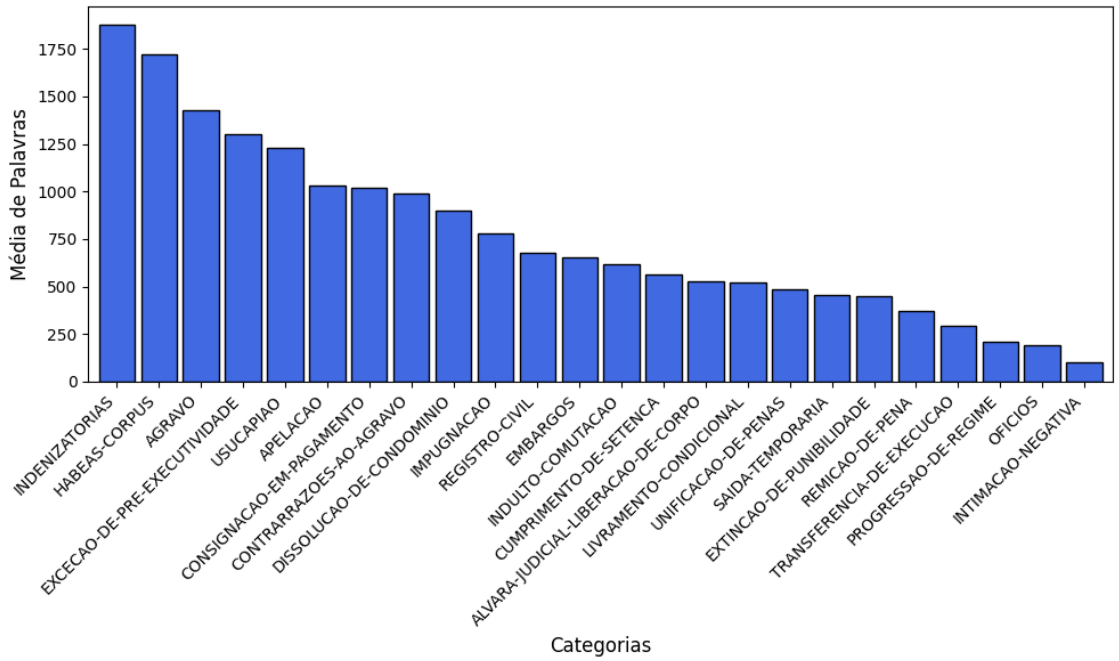

Figura 4.3: *Média de comprimento das palavras por categoria (Fonte: elaborado pelo autor)*

ao treinamento de modelos de aprendizado de máquina, de modo específico ao se tratar de textos mais extensos, como os de "INDENIZATORIAS". Consequentemente, abordagens personalizadas, como a utilização de LLMs com capacidade de representação de grandes volumes de texto, tornam-se necessárias para garantir a precisão e eficiência da classificação automática.

4.1.2 Divisão do *Corpus*

O *corpus* foi dividido [32] em duas partes, o Cojunto de Avaliação, Tabela 4.2 e Conjunto de Classificação, Tabela 4.3. Essa divisão foi feita para garantir que diferentes aspectos dos LLMs pudessem ser avaliados de forma eficiente. Inicialmente, 24,15% do *corpus* foi alocado para o conjunto de avaliação, enquanto os 75,85% restantes foram reservados para o conjunto de classificação. A distribuição das porcentagens foi equilibrada entre as classes, levando em consideração textos com maiores quantidades de amostras até aqueles com menos amostras. Ao final desse processo, observaram-se os seguintes tamanhos de amostras em cada conjunto, o conjunto de avaliação continha 835 amostras, enquanto o conjunto de classificação recebeu 2.623 amostras. Apenas as abordagens de classificação baseadas no texto original e na descrição conseguiram classificar o *corpus* completo, pois não exigiam a geração de resumos ou centroides para a inferência subsequente.

O conjunto de avaliação, sendo utilizado exclusivamente para a geração de centroides e resumos, não foi usado diretamente para a classificação, mas teve um papel fun-

damental na criação de representações compactas e generalizadas dos dados. As categorias com maior número de textos neste conjunto são "EXTINCAO-DE-PUNIBILIDADE" (100 textos), "AGRAVO" (90 textos) e "IMPUGNACAO" (75 textos). Embora o balanceamento seja importante, algumas categorias como "SAIDA-TEMPORARIA" e "ALVARA-JUDICIAL-LIBERACAO-DE-CORPO" possuem apenas 10 textos cada, o que pode afetar a representatividade dessas categorias na criação dos centroides. Além disso, características como o tamanho médio das petições podem ser exploradas como uma abordagem complementar para aprimorar a representatividade em categorias com poucos textos. Textos mais longos, frequentemente associados a maior densidade e complexidade informativa, poderiam receber maior peso na criação de centroides e resumos, equilibrando o impacto das categorias menos representadas no conjunto de dados.

Categoria	**Número de Textos**
EXTINCAO-DE-PUNIBILIDADE	100
AGRAVO	90
IMPUGNACAO	75
EMBARGOS	70
APELACAO	65
REGISTRO-CIVIL	50
LIVRAMENTO-CONDICIONAL	45
INDULTO-COMUTACAO	40
CUMPRIMENTO-DE-SETENCA	35
OFICIOS	30
INDENIZATORIAS	30
TRANSFERENCIA-DE-EXECUCAO	20
EXCECAO-DE-PRE-EXECUTIVIDADE	20
USUCAPIAO	20
UNIFICACAO-DE-PENAS	15
HABEAS-CORPUS	15
INTIMACAO-NEGATIVA	15
DISSOLUCAO-DE-CONDOMINIO	15
REMICAO-DE-PENA	15
PROGRESSAO-DE-REGIME	15
CONSIGNACAO-EM-PAGAMENTO	10
CONTRARRAZOES-AO-AGRAVO	10
SAIDA-TEMPORARIA	10
ALVARA-JUDICIAL-LIBERACAO-DE-CORPO	10

Tabela 4.2: *Quantidade de textos em cada categoria jurídica no conjunto de avaliação (Fonte: elaborado pelo autor)*

Por outro lado, o conjunto de classificação recebeu 2.623 amostras, utilizadas em todas as abordagens de classificação. Nas abordagens baseadas no texto original e na descrição, o conjunto de classificação foi utilizado integralmente, uma vez que não houve a necessidade de resumir ou gerar centroides. As categorias com maior quantidade de textos no conjunto de classificação são novamente "EXTINCAO-DE-PUNIBILIDADE" (400 textos), "AGRAVO" (351 textos) e "IMPUGNACAO" (350 textos).

A discrepância no número de textos entre o conjunto de avaliação e o conjunto

Categoria	Número de Textos
EXTINCAO-DE-PUNIBILIDADE	400
AGRAVO	351
IMPUGNACAO	350
EMBARGOS	234
APELACAO	231
REGISTRO-CIVIL	186
LIVRAMENTO-CONDICIONAL	173
INDULTO-COMUTACAO	130
OFICIOS	117
CUMPRIMENTO-DE-SETENCA	113
INDENIZATORIAS	113
EXCECAO-DE-PRE-EXECUTIVIDADE	70
TRANSFERENCIA-DE-EXECUCAO	60
USUCAPIAO	22
UNIFICACAO-DE-PENAS	15
HABEAS-CORPUS	13
INTIMACAO-NEGATIVA	11
DISSOLUCAO-DE-CONDOMINIO	10
REMICAO-DE-PENA	10
CONSIGNACAO-EM-PAGAMENTO	6
CONTRARRAZOES-AO-AGRAVO	6
PROGRESSAO-DE-REGIME	4
SAIDA-TEMPORARIA	2
ALVARA-JUDICIAL-LIBERACAO-DE-CORPO	2

Tabela 4.3: *Quantidade de textos em cada categoria jurídica no conjunto de classificação (Fonte: elaborado pelo autor)*

de classificação é intencional, dado que o primeiro foi utilizado apenas para a geração de resumos e centroides, enquanto o segundo foi utilizado para avaliar o desempenho do modelo nas abordagens de classificação. Além disso, algumas categorias, como "UNIFICACAO-DE-PENAS" (15 textos) e "SAIDA-TEMPORARIA" (2 textos), possuem poucos exemplos, o que pode dificultar o desempenho do modelo nessas classes. Para mitigar esse problema, técnicas como ajuste de pesos ou balanceamento de dados podem ser aplicadas para garantir uma avaliação justa em todas as categorias.

4.2 Pré-processamento

A Figura 4.4 ilustra o processo de pré-processamento dos dados textuais, iniciado com a coleta de documentos em vários formatos e seguido pela conversão para um formato unificado (.docx). Após essa conversão, os dados passam por etapas de limpeza, normalização, convertendo caracteres para minúsculas e eliminando acentuações e lematização, que consiste em identificar o "lema" de uma palavra flexionada, ou seja, a sua forma originanas quais elementos irrelevantes são removidos e o texto é preparado para a análise. Além disso, há a remoção de *stopwords* e a anonimização de dados sensíveis,

assegurando a conformidade com as exigências de privacidade. Ao final, os documentos processados são armazenados em um formato tabular, facilitando a etapa posterior de classificação. Esse processo é descrito de forma mais detalhada nas seções seguintes, onde cada uma dessas etapas é explorada.

A Figura 4.4 ilustra o processo de pré-processamento dos dados textuais, o processo iniciou com a coleta de documentos em diversos formatos, como .odt, .doc, .html e .pdf, seguido da conversão automatizada para o formato unificado (.docx). Em seguida, houve a limpeza dos dados, removendo elementos indesejados, como caracteres não alfanuméricos, e a normalização do texto, convertendo caracteres para minúsculas e eliminando acentuações. Posteriormente, o texto passou pela lematização e remoção de *stopwords* para simplificar o conteúdo e reduzir ruídos. Antes do armazenamento final, foi realizada a anonimização de dados sensíveis, substituindo informações pessoais e padrões como Cadastro de Pessoa Física (CPF) e números de processos por termos neutros. Por fim, os dados tratados foram armazenados em formato tabular, organizados para facilitar a classificação e análise subsequente.

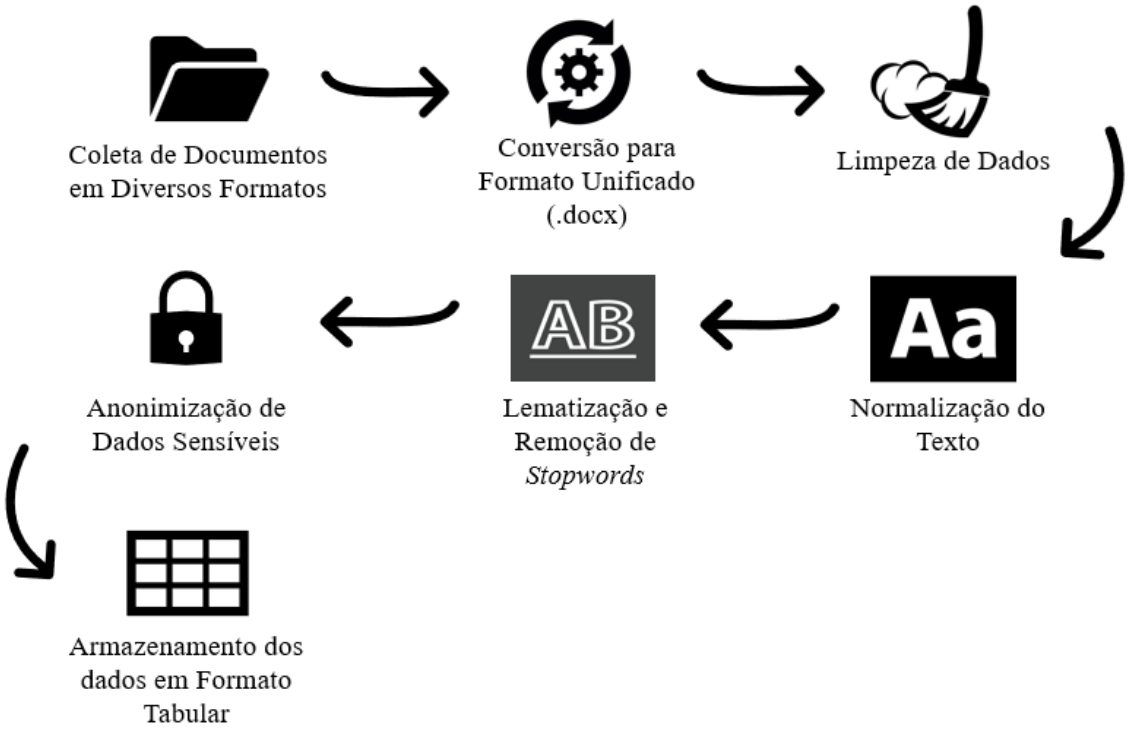

Figura 4.4: *Processo de pré-processamento, incluindo coleta, conversão, limpeza, normalização, lematização, anonimização e armazenamento dos dados em formato tabular (Fonte: elaborado pelo autor)*

Para o processo de pré-processamento dos dados textuais, foi necessário lidar com documentos em formatos, como .odt, .doc, .docm, .html e .pdf, além de arquivos já no formato .docx. A primeira etapa consistiu na **conversão dos arquivos** para um

formato unificado (.docx). A conversão de arquivos para formatos unificados é uma prática importante para garantir a consistência dos dados antes da aplicação dos métodos de representação textual, conforme sugerido por [60] em sua revisão sobre métodos de representação de texto utilizando aprendizado de máquina. Essa conversão foi realizada de forma automatizada por meio de dois *scripts*.

O primeiro *script* utilizou a biblioteca `win32com` para manipular arquivos do Microsoft Word, a automação na manipulação de arquivos utilizando bibliotecas específicas, foi destacada por [50] como uma forma eficaz de facilitar a padronização e processamento de documentos em grande escala. Ele automatizou a abertura e conversão de documentos como .doc, .docm, .dot e .dotm para o formato .docx. Para arquivos em outros formatos, como .odt, a conversão foi realizada usando o comando `unoconv`, que permite a transformação de arquivos OpenDocument (ODF) em formatos de texto do Microsoft Word. Arquivos em Hypertext Markup Language (HTML) e Portable Document Format (PDF) foram convertidos para .docx utilizando a biblioteca `pypandoc`, que oferece uma interface simplificada para o `Pandoc`, um conversor universal de documentos. Após a conversão bem-sucedida, os arquivos originais eram removidos para evitar redundâncias. Essa padronização dos documentos garantiu que todos estivessem no mesmo formato para a fase seguinte de processamento, evitando possíveis problemas relacionados à leitura ou extração de conteúdo de arquivos em formatos variados.

Após a conversão, os documentos passaram por uma **etapa de limpeza e normalização de texto**, que foi executada pelo segundo *script*. O uso de bibliotecas como `BeautifulSoup` tem sido fundamental em tarefas de extração e limpeza de dados, sobretudo na remoção de elementos indesejados de HTML, conforme observado em estudos recentes sobre processamento textual [63]. Inicialmente, utilizou-se a biblioteca `BeautifulSoup` para remover quaisquer elementos de HTML, como *tags* e atributos, que poderiam estar presentes em documentos provenientes de fontes da *web* ou arquivos HTML convertidos. Em seguida, foi realizada a remoção de Uniform Resource Locato (URLs), endereços de *e-mail* e quaisquer outros elementos que não fossem relevantes para a análise textual, como caracteres não alfanuméricos e números.

Uma vez que o texto foi "limpo", todos os caracteres foram convertidos para **minúsculas**, e foram removidas as acentuações usando a biblioteca `unidecode`. Esse processo garantiu que variações ortográficas ou acentuais não interferissem na análise posterior, assegurando a consistência entre os textos. Em seguida, foi aplicada a **lematização** [8], que converteu as palavras para suas formas básicas ou raízes, facilitando a categorização e análise do conteúdo. Para tanto, o lematizador da biblioteca `nltk` foi utilizado. Durante esse processo, as ***stopwords*** (palavras comuns e sem significado relevante para a análise, como artigos e preposições) também foram removidas. Isso reduziu o "ruído" dos textos e ajudou a focar nas palavras mais relevantes para o contexto jurídico

dos documentos.

Os textos, após serem limpos e normalizados, foram armazenados em um formato tabular, com três colunas principais: `COLUNA_TEXTO`, contendo o texto processado; `CATEGORIA`, representando a categoria ou tipo do documento jurídico, obtido a partir do nome da pasta onde o arquivo original estava armazenado; e `ID_CATEGORIA`, que atribuía um identificador numérico único para cada categoria, facilitando a classificação posterior.

Essas etapas de **conversão, padronização e limpeza** foram fundamentais para garantir que os documentos jurídicos pudessem ser processados de maneira eficiente por modelos de PLN. Com os textos padronizados e categorizados, a próxima fase do processo envolveu a tokenização e preparação para a classificação, onde o próprio modelo de linguagem responsável por realizar a classificação lidaria automaticamente com as tarefas de tokenização e segmentação do texto. Essa abordagem permitiu um fluxo contínuo e automatizado desde a conversão até a categorização dos documentos, criando uma base de dados robusta para a análise e extração de informações em grande escala.

Assim, o processo como um todo, que incluiu tanto a padronização dos arquivos quanto a normalização dos textos, foi uma etapa fundamental para garantir a **qualidade dos dados de entrada** e maximizar o desempenho dos modelos de classificação subsequentes, promovendo uma análise precisa e eficaz dos textos jurídicos.

4.2.1 Processo de Anonimização de Dados Sensíveis

O processo de anonimização de dados sensíveis foi estruturado com o objetivo de assegurar a privacidade das informações contidas nos documentos jurídicos analisados, em conformidade com legislações como a Lei Geral de Proteção de Dados Pessoais (LGPD) [71]. A implementação do fluxo de anonimização foi realizada por meio de um *script* em `Python`, utilizando técnicas de processamento de texto e expressões regulares para a identificação e substituição de padrões sensíveis.

O procedimento começa na substituição de termos específicos encontrados nos documentos. Para isso, um conjunto de expressões regulares foi elaborado, abrangendo padrões comuns em textos jurídicos, como números de processos, nomes próprios e informações institucionais. Por exemplo, o padrão `\bPROCESSO\s+(\S+)` é utilizado para substituir números de processos pela palavra "PROCESSO". Essa operação é realizada pela função *anonimizar_texto()*, que itera sobre um dicionário de padrões e substitui os termos encontrados por seus equivalentes neutros. A contagem de substituições realizadas é mantida por meio da função *re.subn()*, que permite rastrear a frequência de cada padrão anonimizado.

O processo realizado incluiu a detecção e anonimização de dados sensíveis, garantindo a proteção das informações e o cumprimento das normas de privacidade. Infor-

mações como números de CPF, Registro Geral (RG), endereços, *e-mails*, e dados bancários são identificados e substituídos utilizando um segundo conjunto de expressões regulares. Este conjunto é mais abrangente, visando cobrir variações na representação dos dados sensíveis, como diferenças no formato ou uso de abreviações. A função *anonimizar_dados_sensiveis()* aplica essas regras e registra o número de ocorrências anonimizadas, assegurando que nenhuma informação pessoal permaneça nos documentos.

Durante todo o processo, o *script* mantém um contador que armazena o número de anonimizações realizadas para cada padrão. Os resultados obtidos são analisados e apresentados em formato visual, por meio de gráficos de barras horizontais gerados com a biblioteca `Matplotlib`, e em formato tabular, utilizando a biblioteca `Pandas`. Esses resultados indicam a frequência de termos sensíveis encontrados nos textos processados, permitindo uma avaliação detalhada da incidência de informações protegidas.

Além disso, os documentos anonimizados foram salvos em novos arquivos nos formatos Comma-Separated Values (CSV) e XLSX, preservando a integridade das informações originais, mas assegurando a anonimização de todos os dados sensíveis. Essa abordagem não apenas garante a conformidade com a LGPD, mas também facilita a reutilização dos dados anonimizados em análises futuras. A implementação modular do *script* permite sua adaptação para outros contextos, como a inclusão de novos padrões ou o processamento de diferentes tipos de documentos, demonstrando a robustez e escalabilidade do método proposto.

O uso de expressões regulares e ferramentas programáticas foi importante para a eficiência do processo, proporcionando uma solução automatizada, confiável e adaptável. A aplicação dessa metodologia reforça a importância de práticas de anonimização no contexto da proteção de dados e evidencia como abordagens técnicas podem ser integradas para garantir a privacidade de informações sensíveis em cenários jurídicos e institucionais.

4.3 Implementação

Nas implementações, foram utilizadas ferramentas e tecnologias descritas a seguir. A linguagem `Python` foi utilizada no desenvolvimento do código. A `Pandas` foi empregada para a manipulação e representação de dados estruturados, em arquivos no formato CSV, permitindo operações eficientes de filtragem e agregação.

O ambiente de execução utilizado para desenvolver e executar o código foi o **`Google Colab`**. O `Colab` facilita a execução de código `Python` com suporte a Graphics Processing Unit (GPUs), ideal para o processamento de modelos de linguagem. Também proporciona fácil integração com o `Google Drive`. O **`Google Drive`** foi utilizado para armazenamento dos *datasets* de entrada (arquivos CSV) e dos resultados gerados

(arquivos CSV e Excel contendo as classificações, resumos, relatórios de métricas e matrizes de confusão). Os códigos usados para construir o *pipeline* e outras etapas da pesquisa está disponível no repositório do `GitHub`.[1]

A biblioteca `Scikit-learn` foi utilizada em muitas tarefas, e a funcionalidade `TfidfVectorizer` foi utilizada para transformar os textos em vetores numéricos, permitindo que as palavras fossem manipuladas de maneira quantitativa. Para calcular a similaridade entre os vetores dos textos e os centroides, foi utilizado a função `cosine_distances`. Além disso, o `LabelEncoder` foi empregado para converter os rótulos das categorias de *string* para valores numéricos.

A biblioteca `Matplotlib` foi utilizada para gerar histogramas da distribuição de *tokens* e gráficos de barras, exibindo as métricas de desempenho dos modelos. Com a `Seaborn`, foi possível criar visualizações, como matrizes de confusão, tanto em formatos numéricos quanto normalizados.

Também foi utilizada a biblioteca `Tiktoken`, importante para contagem precisa de *tokens* nos textos processados, fator importante ao lidar com LLMs. A integração dos modelos de PLN foi facilitada pelo uso da `Langchain`, que permitiu a construção de cadeias de PLN e a lógica necessária para a criação e gestão de *prompts*. Para lidar com conjuntos de dados desbalanceados, utilizamos a `Imbalanced-learn`, que oferece técnicas de balanceamento de dados, fundamentais para garantir a robustez do modelo durante o treinamento.

Os modelos foram acessados via Application Programming Interface (API) e configurado para processar *prompts* e realizar inferências. A **`Langchain_OpenAI`** foi usada usada para interagir com o modelo e construir as cadeias de *prompts* para tarefas de classificação e resumo de textos jurídicos.

O desempenho dos modelos foi avaliado utilizando um conjunto de dados de classificação que passou pelo processo de classificação automática. Para garantir a qualidade das classificações, foram aplicados métodos padronizados de avaliação que incluem a análise de previsões corretas, erros e a capacidade discriminatória do modelo. As ferramentas usadas para gerar e visualizar os resultados incluem a matriz de confusão, histogramas de contagem de *tokens*, e gráficos de barras para comparação das classes.

As avaliações foram realizadas com o objetivo de verificar a eficiência do modelo ao lidar com textos jurídicos, bem como sua capacidade de generalizar a partir dos dados de classificação para novos casos.

[1] https://github.com/Willgnner-Santos/DPE-Legal-Doc-Classification-Pipeline.git

4.4 Ambiente de Avaliação

O ambiente de avaliação foi configurado e executado no **Google Colab**, uma plataforma baseada em nuvem que permite a execução de *scripts* Python com suporte a *hardware* especializado, como **GPUs** e **Tensor processing Unit (TPUs)**.

4.4.1 Configurações de *Hardware* e *Software*

Google Colab versão Pro, com suporte a GPU, fornecendo aceleração para o processamento dos modelos de linguagem e vetorização dos textos. **Sistema operacional** Windows 11, baseado no ambiente Colab. **Memória de Acesso Aleatório (RAM)** aproximadamente 25 Gigabyte (GB), disponíveis no Colab Pro. **Armazenamento** Google Drive, utilizado como sistema de armazenamento de dados e resultados gerados, com integração direta ao Colab. **API Together** utilizada para acessar modelos disponíveis na plataforma Together.

Além disso, as seguintes bibliotecas de *software* foram instaladas para a execução do método. **Python 3.9** foi a versão utilizada no ambiente Colab. As **bibliotecas Python** principais incluíram Pandas, Matplotlib, Seaborn, Tiktoken, Scikit-learn, Imbalanced-learn, Langchain e Langchain_OpenAI.

4.4.2 Conjuntos de Dados

Para a avaliação do método, foram utilizados dois conjuntos de dados fornecidos pela DPE-GO. Esses conjuntos apresentam informações organizadas em colunas específicas que descrevem o conteúdo dos dados.

O primeiro conjunto de dados contém duas colunas principais, sendo facts, que representa o texto da petição, e issue_area, que indica a categoria jurídica associada. Esse conjunto foi utilizado para o cálculo de centroides e para a geração de resumos de cada categoria, além de servir para testar o desempenho do modelo com textos já categorizados.

O segundo conjunto de dados contém textos que ainda não foram classificados. Esse conjunto foi utilizado para avaliar a capacidade do modelo em realizar a classificação automática desses textos em suas respectivas categorias jurídicas.

4.4.3 Parâmetros de Algoritmos e Ferramentas

Os parâmetros e configurações utilizados para a replicação dos resultados incluem a **API base** https://api.together.xyz/v1, com os **parâmetros de inferência** configurados em uma **temperatura** de 0.1, **max_*tokens*** de 100 e o parâmetro Verbose definido como *true*.

Vetorização de Textos

Para a vetorização dos textos das petições, utilizou-se a técnica TF-IDF com o **vectorizer** `TfidfVectorizer()` da biblioteca `Scikit-learn`, configurado de forma padrão, sem remoção de *stopwords* ou limites de n-gramas.

Classificação de Textos

O processo de classificação foi realizado utilizando a classe **`PromptTemplate`** da biblioteca Langchain, onde um *template* foi utilizado para gerar *prompts* específicos com o texto centroide e o texto de interesse, e **`Langchain`**, que integrou os modelos e o processo de geração de *prompts* e inferências.

4.4.4 Ferramentas de Avaliação

A avaliação dos resultados foi realizada por meio de ferramentas de visualização e análise. A **matriz de confusão** foi utilizada para avaliar a precisão das classificações feitas pelo modelo, gerada com a biblioteca `Seaborn`. Um **histograma de *tokens*** foi criado para visualizar a distribuição da contagem de *tokens* nos textos originais e resumidos, utilizando a biblioteca `Matplotlib`. Os **gráficos de barras** foram gerados gráficos para comparar as métricas de *precision*, *recall*, e *F1-score* por classe.

Todos os dados e resultados foram salvos no `Google Drive`, em formatos `CSV` e `XLSX`, facilitando a replicação e a análise posterior.

4.5 Métricas de Avaliação

Para avaliar o desempenho do modelo de classificação, a escolha das métricas foi realizada com base em critérios científicos, considerando tanto a natureza dos dados quanto o contexto de aplicação. Embora a acurácia seja uma métrica intuitiva, ela pode ser enganosa em cenários com dados desbalanceados, razão pela qual foi complementada por métricas como precisão, *recall* e *F1-score*, que fornecem uma visão mais granular sobre os erros do modelo. O *F1-score*, em particular, foi selecionado como métrica a ser maximizada, pois equilibra precisão e *recall*, sendo relevante em situações onde falsos positivos e falsos negativos possuem impacto considerável. Além disso, métricas como AUC-ROC e MCC foram consideradas para avaliar a capacidade discriminativa do modelo e sua robustez em cenários desbalanceados. A exclusão de outras métricas foi motivada pela necessidade de selecionar aquelas que capturam com maior precisão os erros significativos e refletem o desempenho do modelo em condições reais, garantindo uma análise alinhada aos objetivos práticos do sistema.

4.5.1 Acurácia

A acurácia mede a proporção de previsões corretas em relação ao número total de previsões realizadas. Embora seja uma métrica simples e intuitiva, pode ser enganosa em conjuntos de dados desbalanceados, onde uma classe é muito mais frequente que outra.

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4-1}$$

Onde:

- Verdadeiros Positivos, do inglês, True Positives (TP).
- Verdadeiros Negativos, do inglês, True Negatives (TN).
- Falsos Positivos, do inglês, False Positives (FP).
- Falsos Negativos, do inglês, False Negatives (FN).

4.5.2 Precisão

A precisão mede a proporção de previsões positivas corretas em relação ao número total de previsões positivas. Essa métrica é útil quando o custo de um falso positivo é elevado.

$$\text{Precisão} = \frac{TP}{TP + FP} \tag{4-2}$$

4.5.3 *Recall*

O *recall*, também conhecida como sensibilidade, mede a capacidade do modelo de identificar corretamente todas as instâncias positivas. Esta métrica é importante quando o custo de um falso negativo é alto.

$$Recall = \frac{TP}{TP + FN} \tag{4-3}$$

4.5.4 *F1-Score*

O *F1-score* é a média harmônica entre precisão e revocação, oferecendo um equilíbrio entre ambas. É útil quando há uma necessidade de compromisso entre as duas métricas, como em problemas com classes desbalanceadas.

$$F1\text{-}score = 2 \times \frac{\text{Precisão} \times Recall}{\text{Precisão} + Recall} \tag{4-4}$$

4.5.5 AUC-ROC

A curva ROC (Receiver Operating Characteristic) avalia o desempenho do modelo ao variar o limiar de decisão, plotando a Taxa de Verdadeiros Positivos, do inglês, True Positive Rate (TPR) contra a Taxa de Falsos Positivos, do inglês, False Positive Rate (FPR).

$$\text{TPR} = \frac{TP}{TP+FN}, \quad \text{FPR} = \frac{FP}{FP+TN} \tag{4-5}$$

4.5.6 AUC-PR

$$\text{AUC-PR} = \int_0^1 \text{Precision}(\text{Recall})\, d(\text{Recall}) \tag{4-6}$$

A Area Under the Precision-Recall Curve (AUC-PR) é uma métrica relevante para conjuntos de dados desbalanceados. Ela foca na qualidade das previsões positivas e na capacidade do modelo de identificar corretamente as instâncias positivas. A fórmula apresentada calcula a AUC-PR como a área sob a curva de Precisão em função do *Recall*, representada pela integral. Isso significa que somamos os valores de Precisão ao longo de todos os valores possíveis de *Recall* (de 0 a 1), avaliando o desempenho do modelo em diferentes limiares de classificação. Quanto maior a AUC-PR, melhor o modelo.

4.5.7 MCC

O Coeficiente de Correlação de Matthews (MCC) é uma métrica que considera todas as quatro categorias de previsões (verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos) e é uma métrica robusta mesmo em cenários de dados desbalanceados. O MCC varia de -1 (previsão perfeitamente incorreta) a 1 (previsão perfeitamente correta), sendo 0 equivalente a uma previsão aleatória.

$$\text{MCC} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{4-7}$$

CAPÍTULO 5

Resultados

5.1 Anonimizações

Os resultados do processo de anonimização demonstraram uma aplicação eficaz da metodologia proposta, com destaque para a alta frequência de anonimizações realizadas em padrões sensíveis como números de processos e nomes próprios, conforme ilustrado na Figura 5.1. Os números de processos foram os dados mais anonimizados, totalizando mais de 8.000 ocorrências, seguidos por nomes próprios, que ultrapassaram 4.000 anonimizações. Esses resultados refletem a relevância dos padrões escolhidos e a robustez do *script* na identificação e substituição de informações relevantes.

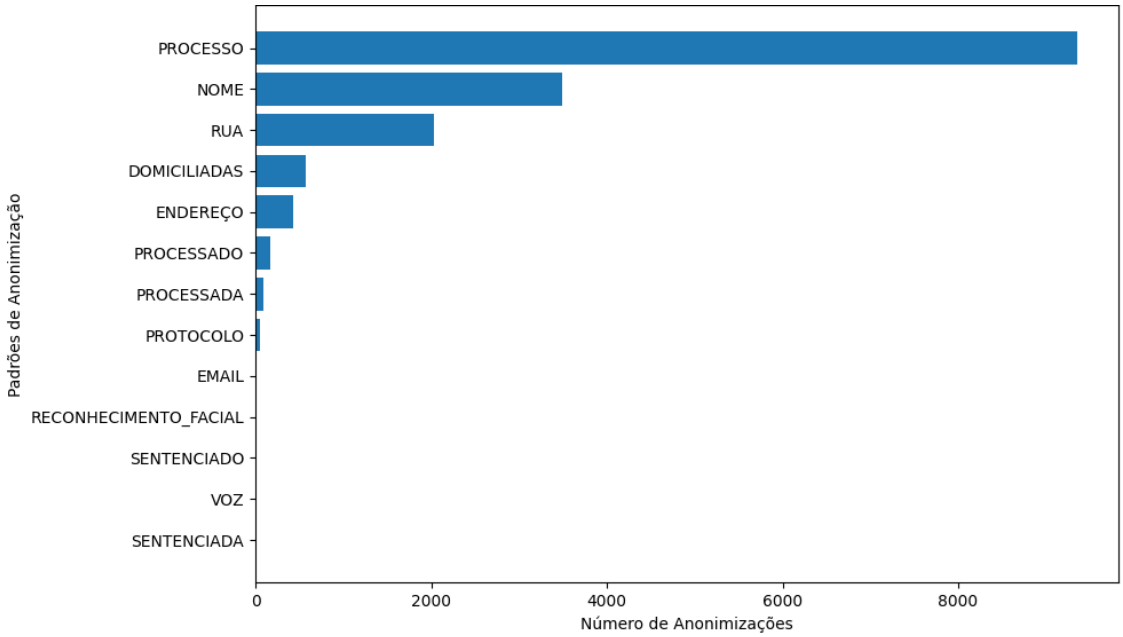

Figura 5.1: *Anonimizações realizadas, indicando a frequência de cada tipo de dado anonimizado (Fonte: elaborado pelo autor)*

Além disso, o nível de assertividade do processo foi validado pela Diretoria de Tecnologia da Informação (DTI) da DPE-GO, com ciência e aprovação do Departamento de Ciência de Dados (DCD). Essa validação atesta a eficiência da técnica empregada e

assegura que os dados anonimizados atendem aos requisitos de privacidade estabelecidos pela LGPD.

A distribuição das anonimizações evidenciada no gráfico também sugere uma predominância de certos padrões sensíveis nos documentos jurídicos analisados, como endereços e informações institucionais, reforçando a importância de uma abordagem automatizada e direcionada. A anonimização de informações como *e-mails*, dados bancários e identificadores pessoais (CPF, RG) foi igualmente eficaz, com impacto direto na proteção de informações sensíveis e na conformidade com as legislações aplicáveis.

Esses resultados não apenas destacam a eficácia técnica do método utilizado, mas também demonstram como práticas de anonimização automatizada podem ser integradas em fluxos de trabalho institucionais para fortalecer a segurança e a privacidade de dados em grande escala.

5.2 Análise de Correlação e Seleção de Métricas Estratégicas

Os resultados apresentados na Tabela 5.1 indicam uma correlação entre todas as métricas de desempenho e a pontuação *F1*. Essa convergência positiva indica que, à medida que métricas como acurácia, precisão e *recall* aumentam, a pontuação *F1* também tende a aumentar. Como a *F1-score* é calculada como a média harmônica entre precisão e *recall*, ela se mostrou uma métrica importante nos experimentos, equilibrando a capacidade do modelo de identificar corretamente os verdadeiros positivos, enquanto reduz tanto os falsos positivos quanto os falsos negativos.

Essa correlação também reforça o papel central da pontuação *F1* na avaliação de modelos de classificação de documentos jurídicos, já que ela oferece uma visão mais equilibrada do desempenho geral. Ao convergirem para uma *F1-score* mais alta, as demais métricas indicam que o modelo não apenas realiza previsões precisas, mas também é capaz de manter uma consistência em termos de *recall* e precisão, o que é fundamental em tarefas de classificação de alta complexidade, como as que envolvem textos jurídicos.

5.3 Avaliação de desempenho de LLMs

Os resultados na Tabela 5.1 indicam variações no desempenho dos LLMs ao aplicar diferentes métodos de representação dos documentos de entrada. Essas abordagens foram importantes para avaliar como cada modelo lida com distintas estratégias de otimização e processamento de texto, visando melhorar a eficiência computacional sem comprometer a precisão.

Modelo	Método	Acurácia	Precisão	*Recall*	*F1-Score*	AUC-ROC	AUC-PR	MCC
Llama-3.1-70B-Instruct-Turbo	Resumos	**99.92%**	99.92%	99.92%	**99.92%**	1	0.99	1
	Centroides	99.43%	98.88%	99.43%	99.15%	0.98	0.96	0.99
	Descrições	88.64%	92.94%	88.64%	89.98%	0.93	0.76	0.88
	Class. Textual	90.05%	93.52%	90.05%	91.13%	0.95	0.78	0.89
Qwen2-72B-Instruct	Resumos	99.81%	99.62%	99.81%	99.71%	0.98	0.96	1
	Centroides	99.01%	98.07%	99.01%	98.53%	0.96	0.91	0.99
	Descrições	90.02%	93.30%	90.02%	91.10%	0.94	0.77	0.89
	Class. Textual	87.77%	90.87%	87.77%	88.50%	0.92	0.70	0.87
Llama-3.1-8B-Instruct-Turbo	Resumos	91.31%	90.75%	91.31%	88.52%	0.95	0.87	0.91
	Centroides	59.86%	70.12%	59.86%	54.54%	0.71	0.38	0.61
	Descrições	88.64%	92.94%	88.64%	89.98%	0.93	0.76	0.58
	Class. Textual	49.48%	75.90%	49.48%	51.92%	0.69	0.29	0.49
Mixtral-8x22B-Instruct-v0.1	Resumos	86.54%	83.19%	86.54%	83.88%	0.98	0.93	0.87
	Centroides	89.90%	85.49%	89.90%	86.98%	0.89	0.77	0.89
	Descrições	52.49%	72.07%	52.49%	54.11%	0.73	0.34	0.50
	Class. Textual	79.38%	84.06%	79.38%	80.31%	0.88	0.59	0.78
Mistral-7B-Instruct-v0.3	Resumos	82.16%	72.45%	82.16%	75.90%	0.95	0.89	0.82
	Centroides	31.45%	24.45%	31.45%	22.52%	0.59	0.21	0.28
	Descrições	29.53%	47.15%	29.53%	26.34%	0.55	0.10	0.24
	Class. Textual	45.14%	56.46%	45.14%	41.86%	0.66	0.22	0.41
Mixtral-8x7B-Instruct-v0.1	Resumos	79.26%	68.88%	79.26%	72.46%	0.94	0.85	0.79
	Centroides	64.39%	54.78%	64.39%	57.02%	0.79	0.55	0.64
	Descrições	52.49%	72.07%	52.49%	54.11%	0.73	0.34	0.50
	Class. Textual	48.79%	71.73%	48.79%	47.53%	0.69	0.27	0.46
Llama-3.2-3B-Instruct-Turbo	Resumos	77.24%	72.38%	77.24%	71.19%	0.88	0.73	0.77
	Centroides	21.31%	29.05%	21.31%	13.39%	0.57	0.14	0.19
	Descrições	30.62%	51.34%	30.62%	28.42%	0.64	0.18	0.27
	Class. Textual	21.92%	41.75%	21.92%	16.65%	0.58	0.13	0.19

Tabela 5.1: *Comparação de desempenho de diferentes LLMs, com métricas de acurácia, precisão, recall, F1-score, AUC-ROC, AUC-PR e MCC para cada método de classificação (resumos, centroides, classificação textual e descrições (Fonte: elaborado pelo autor)*

Entre os modelos avaliados, o **Llama-3.1-70B-Instruct-Turbo**, que conta com um comprimento de contexto de 131.072 *tokens* e 70 bilhões de parâmetros, demonstrou um desempenho alto na abordagem de classificação baseada em resumos, atingindo uma acurácia e *F1-score* de 99.92%. A abordagem de resumos foi capaz de reduzir o número de *tokens* processados, com uma média de 97.96 *tokens* por resumo, em comparação aos 1.613,79 *tokens* da média dos textos completos. Essa redução trouxe uma melhora na eficiência computacional, tornando o processamento mais rápido e menos custoso. Além disso, os resultados indicam que a condensação do texto através de resumos não comprometeu a precisão, sugerindo que essa técnica foi eficaz em capturar os elementos relevantes necessários para a classificação.

Por outro lado, a **classificação baseada em centroides** se mostrou menos eficiente, com uma média de 1.602,29 *tokens* por centroide. O Llama-3.1-70B também obteve um bom desempenho com essa técnica (acurácia de 99.43% e *F1-score* de 99.15%), embora o aumento no número de *tokens* tenha resultado em um custo computacional mais

elevado. Mesmo assim, a utilização de centroides proporcionou ao modelo uma visão geral das características centrais de cada categoria, o que, apesar de sua menor eficiência em relação aos resumos, mostrou-se benéfico para manter a precisão da classificação.

O **Qwen2-72B-Instruct**, com 72 bilhões de parâmetros e capacidade de processar até 32.768 *tokens*, também apresentou um desempenho elevado, alcançando uma acurácia de 99.81% e um *F1-score* de 99.71% com a abordagem de resumos. No entanto, notou-se que este modelo teve uma leve queda de desempenho na abordagem de classificação baseado no texto original (Class. Textual) e centroides, sugerindo que a eficiência desse modelo é melhor aproveitada com dados processados em formato resumido.

Modelos menores, como o **Llama-3.1-8B-Instruct-Turbo**, com 8 bilhões de parâmetros, mostraram uma redução no desempenho ao utilizar a abordagem de centroides, com uma acurácia de 59.86% e *F1-score* de 54.54%. Este resultado indica que modelos com menos parâmetros podem sofrer mais com a perda de detalhes essenciais ao utilizar técnicas de simplificação como os centroides. No entanto, esse modelo apresentou bons resultados com a abordagem de resumos, atingindo 91.31% de acurácia e 88.52% de *F1-score*, o que sugere que o resumo do texto é uma técnica mais apropriada para modelos com menor capacidade de processamento.

O **Mistral-7B-Instruct-v0.3**, com um comprimento de contexto de 32.768 *tokens* e 7 bilhões de parâmetros, teve um desempenho contrastante entre as abordagens. Ele obteve uma acurácia de 82.16% na abordagem de resumos, mas um desempenho inferior com centroides (31.45%). Essa discrepância destaca a dificuldade dos modelos menores em processar de forma eficaz representações mais condensadas, como os centroides, onde a variabilidade textual pode impactar negativamente a precisão. O desempenho desse modelo também foi prejudicado na classificação baseado em descrição, com apenas 29.53% de acurácia, reforçando a ideia de que ele não lida bem com textos mais longos ou detalhados.

O **Mixtral-8x22B-Instruct-v0.1**, com 22 bilhões de parâmetros e capacidade de processar até 65.536 *tokens*, apresentou um bom equilíbrio entre as abordagens de resumos e centroides, atingindo acurácia de 86.54% e 89.90%, respectivamente. Esse modelo demonstrou ser capaz de lidar bem com ambas as técnicas de representação dos documentos de entrada, mantendo um desempenho consistente sem comprometer a eficiência.

A inclusão do **Mixtral-8x7B-Instruct-v0.1** mostra a relevância de considerar o equilíbrio entre o tamanho do modelo e o custo computacional. Embora não tenha atingido a mesma precisão dos maiores LLMs, ele ainda foi capaz de fornecer resultados satisfatórios, em tarefas que envolvem resumos. Contudo, quando se utilizam técnicas como centroides ou descrições, seu desempenho cai, o que reforça a necessidade de ajuste e otimização dependendo do tipo de dado e da técnica de pré-processamento aplicada.

Quando analisamos a **Class. Textual**, observa-se que o **Llama-3.1-70B-Instruct-Turbo** também se destaca, atingindo uma acurácia de 90.05% e *F1-score* de 91.13%. Isso sugere que o modelo possui a robustez necessária para processar textos completos, mantendo alta precisão mesmo sem o uso de técnicas adicionais de representação. Por outro lado, a abordagem baseada em **descrição** apresentou uma leve redução de desempenho (88.64%), o que indica que, embora as descrições forneçam um contexto detalhado para o modelo, essa abordagem pode aumentar os custos computacionais devido ao maior número de *tokens* processados.

O **Llama-3.2-3B-Instruct-Turbo**, com 3 bilhões de parâmetros, apresentou os resultados mais baixos entre os modelos avaliados, com uma acurácia de 77.24% na abordagem de resumos e apenas 21.31% com centroides. Isso indica que modelos com menor capacidade de processamento de *tokens* enfrentam dificuldades para manter a precisão quando técnicas mais complexas, como centroides ou descrições, são utilizadas.

Os resultados indicam que os modelos maiores, como o **Llama-3.1-70B-Instruct-Turbo** e o **Qwen2-72B-Instruct**, se destacam em termos de acurácia e *F1-score*, na abordagem de resumos, que equilibra eficiência e precisão. Modelos menores, como o **Mistral-7B-Instruct-v0.3** e o **Llama-3.2-3B-Instruct-Turbo**, mostram limitações quando se aplicam técnicas mais sofisticadas de representação de texto, sugerindo que sua capacidade de contexto e processamento é insuficiente para lidar com a complexidade dos dados jurídicos de forma eficaz.

Durante o estudo, foram avaliadas diferentes arquiteturas dos LLMs, considerando aspectos como capacidade de processamento de *tokens*, número de parâmetros e eficiência em tarefas de classificação de documentos jurídicos. A análise destacou a robustez dos modelos maiores, que obtiveram alta acurácia e *F1-score* com técnicas de representação como resumos, enquanto modelos menores apresentaram limitações em tarefas que exigem maior capacidade de contexto e processamento detalhado. Essa avaliação reforça a importância de selecionar arquiteturas adequadas às demandas computacionais e à complexidade dos dados no uso de LLMs para classificação automática de documentos.

5.3.1 Análise de Acurácias Altas

Os resultados de 99.92% de acurácia e *F1-score* alcançados pelo Llama-3.1-70B-Instruct-Turbo e de 99.81% de acurácia e 99.71% de *F1-score* pelo Qwen2-72B-Instruct na abordagem de resumos devem ser interpretados com cautela. Embora ambas as pontuações sejam altas e demonstrativas da eficácia dos modelos em seus respectivos contextos, é importante avaliar os fatores que podem ter contribuído para esses números e as implicações para sua generalização em cenários mais amplos.

No caso do **Llama-3.1-70B-Instruct-Turbo**, o tamanho robusto do modelo,

com 70 bilhões de parâmetros e capacidade de processar até 131.072 *tokens*, permite que ele lide com textos resumidos sem perder informações relevantes. No entanto, a quantidade relativamente limitada de dados no conjunto de avaliação pode ter favorecido o desempenho, em um ambiente com pouca variabilidade entre as categorias. Esse cenário aumenta a chance de o modelo "memorizar" padrões, o que infla os resultados de forma não representativa para um contexto real.

Da mesma forma, o **Qwen2-72B-Instruct** também apresentou resultados altos com a abordagem de resumos, sugerindo que seu tamanho (72 bilhões de parâmetros) e comprimento de contexto de 32.768 *tokens* foram adequados para lidar com a tarefa. No entanto, é importante destacar que, em situações onde as categorias são bem definidas e apresentam pouca sobreposição textual, o modelo pode estar beneficiado por essa estrutura mais clara. Em um *corpus* mais desafiador e com categorias ambíguas, é possível que o desempenho caia, como frequentemente ocorre em conjuntos de dados mais complexos e variados.

Além disso, tanto o Llama-3.1-70B quanto o Qwen2-72B podem estar se beneficiando de características textuais facilmente distinguíveis entre as classes avaliadas. A presença de categorias bem delimitadas e com pouca variabilidade interna facilita o trabalho de classificação para ambos os modelos. Isso indica que, embora os resultados de acurácia e *F1-score* sejam altos, eles podem não refletir com precisão o desempenho dos modelos em cenários onde as categorias apresentam maior sobreposição de características textuais ou onde os textos são mais ambíguos.

Outro ponto a ser considerado é a importância de um *pipeline* bem estruturado ao utilizar resumos como técnica de representação dos documentos. A redução no número de *tokens* processados contribui para a eficiência computacional sem comprometer a precisão. Contudo, essa abordagem depende de uma divisão estratégica dos dados e de métodos robustos de análise estatística. Em um cenário com maior diversidade e volume de textos jurídicos, o uso de *pipelines* otimizados torna-se ainda mais relevante para garantir que os modelos mantenham um desempenho consistente. É importante reconhecer que, com um *corpus* mais amplo e complexo, os resultados elevados observados neste estudo podem não se repetir, uma vez que a variabilidade e o volume de dados influenciam a capacidade dos modelos de generalizar adequadamente.

Outro aspecto relevante é a relação entre o idioma utilizado no treinamento dos modelos e seu desempenho em tarefas de classificação em português. Modelos como o Llama-3.1-70B-Instruct-Turbo e Qwen2-72B-Instruct possuem treinamentos baseados em corpora multilíngues, mas a presença e a qualidade dos dados em português podem influenciar nos resultados. Essa relação se torna ainda mais evidente em tarefas que exigem um entendimento profundo de *nuances* linguísticas, como no caso dos textos jurídicos. Assim, modelos treinados com maior exposição ao português tendem a apresentar vanta-

gens específicas em relação à precisão e eficiência para essas tarefas.

Essas limitações indicam que, apesar dos resultados promissores, há uma necessidade contínua de supervisão e ajustes manuais no processo de representação e classificação em cenários com alto volume de dados e textos com *nuances* mais sutis.

5.4 Comparação dos Tipos de Representação Textual

A Tabela 5.2 mostra que os diferentes tipos de representação textual, sendo a classificação baseada no texto original, centroides, resumos e descrições, impactam o desempenho dos modelos. Nesta seção, discutimos as diferenças entre essas abordagens, calculando as médias de cada métrica para facilitar a análise comparativa.

A abordagem de **resumos** apresentou o melhor desempenho geral em termos de acurácia, precisão, *recall*, e *F1-score*. Modelos como o Llama-3.1-70B-Instruct-Turbo e o Qwen2-72B-Instruct atingiram quase 100% nessas métricas ao utilizar resumos, indicando que essa técnica captura bem as características essenciais dos textos, mantendo a eficiência computacional devido à redução no número de *tokens*. Essa técnica é eficaz em tarefas que demandam alta precisão e baixa perda de informação. É importante consultar a subseção 5.3.1 para mais detalhes das pontuações de quase 100%.

A técnica de **centroides** também resultados altos, mas de forma ligeiramente inferior aos resumos. Os modelos Llama-3.1-70B-Instruct-Turbo e Qwen2-72B-Instruct, por exemplo, obtiveram altos valores de acurácia e *F1-score*, mas com um custo computacional maior devido ao maior número de *tokens* processados em cada centroide. Apesar disso, essa abordagem ainda é vantajosa para tarefas que requerem uma visão geral das características centrais de cada categoria em contextos onde resumos detalhados não são viáveis.

A abordagem de **descrição** teve um desempenho inferior em termos de *F1-score* e acurácia em comparação aos métodos de resumos e centroides. Isso pode ser atribuído ao fato de que as descrições processam um maior volume de *tokens*, o que pode impactar a eficiência e dificultar o destaque das informações mais relevantes. Portanto, essa metodologia pode não ser a mais eficiente para tarefas de classificação que envolvem categorias diversas e complexas.

A **classificação baseada no texto original** apresentou o pior desempenho em comparação com as outras técnicas. Embora tenha mantido uma precisão relativamente alta, o *F1-score* e o *recall* foram mais baixos, indicando que essa abordagem não consegue capturar adequadamente as *nuances* dos dados textuais. Essa limitação sugere que, embora o método possa ser útil em situações que priorizam simplicidade e rapidez, ele é menos eficaz para tarefas que exigem uma compreensão mais aprofundada do contexto.

5.5 Médias Comparativas

A Tabela 5.2 apresenta as médias das principais métricas para cada tipo de representação textual, calculadas com base nos resultados obtidos pelos modelos. Os valores médios mostram que a técnica de resumos é a mais eficaz entre as abordagens testadas. Em comparação com a abordagem de descrições, a acurácia média dos resumos é 42.51% maior (de 61.77% para 88.03%). Esse padrão de melhoria se repete em outras métricas, como o *F1-score*, onde a abordagem de resumos apresenta um aumento de 41.57% em relação a classificação textual (de 59.70% para 84.51%).

Método	Acurácia	Precisão	*Recall*	*F1-Score*	AUC-ROC	AUC-PR	MCC
Resumos	**88.03%**	83.88%	88.03%	**84.51%**	0.95	0.88	0.88
Centroides	66.47%	65.83%	66.47%	61.73%	0.78	0.56	0.65
Descrições	61.77%	74.54%	61.77%	62.00%	0.77	0.46	0.55
Class. Textual	60.36%	73.47%	60.36%	59.70%	0.76	0.42	0.58

Tabela 5.2: *Comparação das médias das métricas para cada método de representação textual, destacando a eficácia da abordagem de resumos em relação às demais. Os valores indicam o desempenho médio em termos de acurácia, precisão, recall, F1-score, AUC-ROC, AUC-PR e MCC*

Essa análise comparativa ressalta a importância de escolher a representação textual adequada para o contexto de uso. A abordagem de resumos provou ser a mais vantajosa, tanto em termos de eficiência computacional quanto de precisão, superando as outras abordagens. Em particular, a diferença de 21.56% na acurácia entre a abordagem de resumos e a de centroides (88.03% vs. 66.47%) e o aumento de 36.9% no *F1-score* destacam a superioridade dos resumos para cenários de classificação complexa. Por outro lado, métodos como descrições e classificação textual devem ser utilizados com cautela nesses contextos, devido ao desempenho relativamente inferior.

5.6 Avaliação de Desempenho por Categoria

Na análise das Tabelas 5.3 e 5.4, observa-se que os métodos de **resumos** e **descrições** tendem a alcançar melhores médias de precisão, *recall* e *F1-score* na maioria das categorias. Isso sugere que métodos que processam textos de forma mais elaborada, como resumos e descrições, conseguem captar melhor as *nuances* presentes nos documentos jurídicos. Por outro lado, centroides e Class. Textual mostraram desempenho mais modesto, indicando maior dificuldade em lidar com a complexidade dos textos em determinadas categorias, como "CONTRARRAZOES-AO-AGRAVO" e "CUMPRIMENTO-DE-SENTENCA".

Além disso, o modelo Llama-3.1-70B-Instruct-Turbo, conforme apresentado na Tabela 5.1, foi avaliado em todas as categorias e demonstrou resultados consistentes e robustos com o método de resumos, onde alcançou altos índices de precisão, *recall* e *F1-score*. Esse modelo reforça a eficácia dos métodos de representação textual mais elaborados, oferecendo desempenho superior ao lidar com a variedade de categorias e complexidades dos textos jurídicos, sem necessidade de detalhamento adicional na tabela de categorias individuais.

5.6.1 Categorias Mais Desafiadoras

Ao calcular a média das métricas de precisão, *recall* e *F1-score* para todas as categorias, percebe-se que as categorias "PROGRESSAO-DE-REGIME" e "CUMPRIMENTO-DE-SENTENCA" apresentaram maiores desafios para os diferentes métodos de classificação. Essas categorias, que podem possuir terminologias e estruturas jurídicas complexas, podem exigir um entendimento mais refinado dos modelos. Além disso, a quantidade reduzida de textos nessas categorias (com apenas 10 textos para "PROGRESSAO-DE-REGIME" e 35 para "CUMPRIMENTO-DE-SENTENCA" no conjunto de avaliação) pode ter contribuído para a variação dos resultados, uma vez que um volume pequeno de dados tende a dificultar o aprendizado adequado dos modelos, particularmente em métodos como centroides e classificação baseada no texto original. Essa limitação no número de amostras sugere uma possível influência negativa no desempenho geral.

5.7 Principais Resultados, Desempenho Médio e Desafios dos LLMs

O modelo Llama-3.1-70B-Instruct-Turbo (melhor resultado) na Tabela 5.5 apresentou um desempenho notável em praticamente todas as categorias, com valores de precisão, *recall* e F1-score próximos de 1. Esse resultado indica que o modelo tem uma capacidade elevada de prever corretamente as categorias dos textos e, ao mesmo tempo, consegue identificar quase todos os exemplos verdadeiros, minimizando falsos positivos e falsos negativos. A categoria "EXTINCAO-DE-PUNIBILIDADE", atingiu um *F1-score* de 0.92, com um *recall* de 0.85, sugerindo que o modelo é eficiente mesmo em casos mais complexos. No geral, o Llama-3.1-70B demonstrou ser um modelo robusto para a tarefa de classificação de textos jurídicos, lidando bem com a diversidade e complexidade das categorias envolvidas, porém é importante consultar a subseção 5.3.1 para mais detalhes.

O modelo Mistral-8x22B-Instruct-v0.1 (resultado médio) na Tabela 5.6, embora também tenha apresentado um desempenho positivo, ficou abaixo do desempenho do

Llama-3.1-70B. Ele obteve bons resultados em muitas categorias, mas mostrou dificuldades em algumas, como "INDENIZATORIAS", que apresentou um *F1-score* de 0.39, e "IMPUGNACAO", com *F1-score de 0.00*. Esses resultados indicam que o modelo enfrenta desafios específicos em algumas classes, possivelmente devido à baixa representatividade dessas categorias no conjunto de dados. Apesar da precisão alta em muitas categorias, a inconsistência no *recall* sugere que o modelo tem dificuldade em captar todos os exemplos verdadeiros, o que impacta negativamente a performance geral em categorias mais especializadas.

O modelo Llama-3.2-3B-Instruct-Turbo, na Tabela 5.7, apresentou o pior desempenho e com resultados mistos nas diferentes categorias. Embora tenha obtido bom desempenho em categorias como "DISSOLUCAO-DE-CONDOMINIO" (*F1-score* de 0.62) e "USUCAPIAO" (*F1-score* de 0.76), o modelo teve dificuldades notáveis em outras categorias, como "ALVARA-JUDICIAL-LIBERACAO-DE-CORPO" e "UNIFICACAO-DE-PENAS", onde o *F1-score* foi nulo. Isso indica que o modelo apresentou problemas ao identificar corretamente essas classes. Categorias como "AGRAVO" também tiveram baixo *F1-score* (0.22), destacando os desafios de generalização do modelo em cenários com maior variabilidade nos textos jurídicos.

Ao comparar a cobertura das categorias entre os três modelos, o Llama-3.1-70B demonstra uma superioridade, conseguindo manter alta precisão e *recall* na maioria das categorias, o que o torna o modelo mais indicado para a tarefa. O Mixtral-8x22B também tem um desempenho satisfatório, mas enfrenta dificuldades em classes mais complexas ou menos representadas. Por outro lado, o Llama-3.2-3B-Instruct-Turbo falha em várias categorias, evidenciando que, para tarefas que envolvem uma grande variedade de categorias e documentos de alta complexidade, modelos de menor capacidade, não são adequados.

5.8 Discussão dos Resultados

Ao comparar com o desempenho do Llama-3.1-70B e do Qwen2-72B fica evidente que LLMs maiores, com maior capacidade de processamento e contextualização, conseguem capturar *nuances* mais complexas dos textos jurídicos e oferecem resultados superiores em cenários com documentos mais longos e uma maior diversidade de categorias. Os resultados demonstraram um impacto positivo ao utilizar métodos de representação textual com resumos, que ajudam os modelos a capturar as informações mais relevantes e tomar decisões de classificação mais robustas, mesmo em cenários com grande diversidade de categorias jurídicas. Além disso, os resumos reduzem o custo computacional dos LLMs por reduzirem a entrada de contexto dos modelos.

A abordagem de centroides pode ser alternativa viável para modelos como o Qwen2-72B-Instruct e o Llama-3.1-70B-Instruct-Turbo, que conseguiram bons resultados ao usar textos que representam as características centrais das categorias jurídicas. Embora os resultados obtidos com centroides não tenham superado os de resumos, essa técnica demonstrou ser útil em contextos onde o processamento de textos completos seria inviável devido ao custo computacional. Por exemplo, o Qwen2-72B, com um *F1-score* de 98.53% na abordagem de centroides, demonstrou que essa técnica pode ser uma opção eficiente para modelos que precisam lidar com uma grande quantidade de dados de forma mais otimizada.

A análise dos resultados obtidos com diferentes representações textuais, sendo resumos, centroides, classificações baseada no texto original e descrições, revela diferenças no desempenho dos LLMs avaliados no contexto jurídico. Essas variações são importantes para entender a eficácia de cada abordagem no processo de classificação, fornecendo *insights* sobre a melhor técnica de representação de texto para cada tipo de modelo e contexto.

É relevante destacar que LLMs menores, como o Llama-3.1-8B-Instruct-Turbo, também apresentaram um bom desempenho na classificação de documentos jurídicos ao utilizar resumos. Esses modelos podem ser uma excelente alternativa em situações com restrições financeiras ou de recursos computacionais, já que o uso de modelos maiores pode se tornar inviável e comprometer a solução.

A abordagem de resumos apresentou o melhor desempenho geral, em particular para modelos grandes como o Llama-3.1-70B e o Qwen2-72B. O alto *F1-score* e a precisão próximos de 100% para esses modelos indicam que a técnica de resumos consegue captar de maneira eficiente as informações mais relevantes dos documentos jurídicos. Além de manter a precisão, com uma acurácia média de 88.03%, essa técnica reduz o número de *tokens* processados, o que diminui o custo computacional e aumenta a viabilidade para uso contínuo. Essa eficiência é útil em contextos jurídicos que exigem alta precisão, como a classificação de categorias jurídicas específicas, onde erros podem ter impactos consideráveis.

A técnica de centroides, que se baseia na média representativa das características centrais dos documentos, mostrou-se eficaz para modelos de grande porte como o Llama-3.1-70B, com uma acurácia média de 66.47%. Embora essa abordagem não alcance os mesmos níveis de precisão dos resumos, ela apresenta uma boa performance em cenários que exigem uma visão geral das características textuais. No entanto, como processa um número considerável de *tokens*, seu uso pode ser menos eficiente em termos de custo para modelos de menor capacidade. Essa técnica é indicada para cenários onde se precisa de uma compreensão equilibrada do texto, mas sem a necessidade de detalhes minuciosos.

A técnica de descrições, que oferece uma visão detalhada e contextual do

documento, com uma acurácia média de 61.77%. Apesar de fornecer um contexto mais amplo para o modelo, o processamento de descrições implica em um volume elevado de *tokens*, aumentando assim o custo computacional e o tempo de processamento. O modelo Mixtral-8x22B obteve resultados intermediários com descrições, mas para modelos menores, como o Llama-3.2-3B, o desempenho foi reduzido, indicando que essa técnica é menos adequada para tarefas que demandam classificação precisa e eficiente.

A classificação baseada no texto completo, ou seja, utilizando o documento original sem qualquer alteração, apresentou um desempenho inferior em relação aos outros métodos, com uma acurácia média de 60.36%. Embora mantenha uma precisão razoável, modelos menores, como o Llama-3.2-3B, enfrentaram dificuldades em captar as *nuances* dos textos jurídicos quando submetidos a essa abordagem. Essa técnica pode ser indicada para casos onde se prioriza simplicidade e rapidez de implementação, mas ela se mostra limitada para contextos de alta complexidade, como o jurídico, onde é necessário interpretar aspectos semânticos profundos.

Categoria	**Método**	**Precisão Média**	***Recall* Médio**	***F1-Score Médio***
AGRAVO	Resumos	0.71	0.71	0.71
	Centroides	0.56	0.59	0.51
	Class. Textual	0.80	0.54	0.57
	Descrições	0.82	0.61	0.61
ALVARA-JUDICIAL-	Resumos	1.00	1.00	1.00
LIBERACAO-DE-CORPO	Centroides	0.46	0.45	0.45
	Class. Textual	0.47	0.48	0.44
	Descrições	0.50	0.55	0.51
APELACAO	Resumos	0.83	1.00	0.88
	Centroides	0.62	0.96	0.69
	Class. Textual	0.62	0.75	0.60
	Descrições	0.53	0.72	0.55
CONSIGNACAO-	Resumos	1.00	0.88	0.92
EM-PAGAMENTO	Centroides	0.66	0.63	0.64
	Class. Textual	0.75	0.63	0.65
	Descrições	0.55	0.61	0.56
CONTRARRAZOES-	Resumos	0.69	0.71	0.70
AO-AGRAVO	Centroides	0.45	0.55	0.47
	Class. Textual	0.06	0.46	0.11
	Descrições	0.06	0.57	0.11
CUMPRIMENTO-	Resumos	0.84	0.86	0.75
DE-SETENCA	Centroides	0.53	0.46	0.41
	Class. Textual	0.41	0.41	0.37
	Descrições	0.45	0.44	0.41
DISSOLUCAO-	Resumos	1.00	0.94	0.96
DE-CONDOMINIO	Centroides	0.56	0.53	0.54
	Class. Textual	0.81	0.52	0.57
	Descrições	0.90	0.59	0.68
EMBARGOS	Resumos	0.95	1.00	0.97
	Centroides	0.70	0.65	0.67
	Class. Textual	0.86	0.49	0.55
	Descrições	0.81	0.54	0.61
EXCECAO-DE-	Resumos	0.91	1.00	0.93
PRE-EXECUTIVIDADE	Centroides	0.81	0.84	0.82
	Class. Textual	0.90	0.74	0.75
	Descrições	0.84	0.73	0.73
EXTINCAO-DE-	Resumos	0.95	1.00	0.97
PUNIBILIDADE	Centroides	0.53	0.56	0.45
	Class. Textual	0.73	0.77	0.69
	Descrições	0.75	0.70	0.71
HABEAS-CORPUS	Resumos	1.00	0.98	0.99
	Centroides	0.53	0.63	0.53
	Class. Textual	0.57	0.69	0.57
	Descrições	0.61	0.75	0.62
IMPUGNACAO	Resumos	0.61	0.71	0.65
	Centroides	0.70	0.52	0.55
	Class. Textual	0.77	0.61	0.67
	Descrições	0.94	0.63	0.71

Tabela 5.3: *Desempenho médio por categoria para cada método de classificação, incluindo precisão, recall e F1-score - Parte I (Fonte: elaborado pelo autor)*

Categoria	Método	Precisão Média	*Recall* Médio	*F1-Score Médio*
INDENIZATORIAS	Resumos	1.00	0.87	0.88
	Centroides	0.36	0.31	0.32
	Class. Textual	0.57	0.50	0.52
	Descrições	0.53	0.62	0.57
INDULTO-COMUTACAO	Resumos	0.91	1.00	0.93
	Centroides	0.74	0.73	0.65
	Class. Textual	0.70	0.55	0.53
	Descrições	0.74	0.52	0.54
INTIMACAO-NEGATIVA	Resumos	1.00	1.00	1.00
	Centroides	0.52	0.46	0.47
	Class. Dir	0.56	0.25	0.30
	Descrições	0.64	0.26	0.36
LIVRAMENTO-CONDICIONAL	Resumos	0.99	1.00	0.99
	Centroides	0.80	0.57	0.61
	Class. Textual	0.80	0.59	0.64
	Descrições	0.74	0.72	0.69
OFICIOS	Resumos	0.57	0.57	0.57
	Centroides	0.86	0.85	0.85
	Class. Textual	0.99	0.79	0.86
	Descrições	0.90	0.86	0.86
PROGRESSAO-DE-REGIME	Resumos	0.71	0.71	0.71
	Centroides	0.38	0.25	0.28
	Class. Textual	0.46	0.41	0.39
	Descrições	0.51	0.48	0.44
REGISTRO-CIVIL	Resumos	0.86	0.86	0.86
	Centroides	0.56	0.48	0.50
	Class. Textual	0.82	0.51	0.54
	Descrições	0.47	0.45	0.45
REMICAO-DE-PENA	Resumos	1.00	1.00	1.00
	Centroides	0.64	0.50	0.52
	Class. Textual	0.43	0.29	0.32
	Descrições	0.47	0.45	0.45
SAIDA-TEMPORARIA	Resumos	1.00	1.00	1.00
	Centroides	0.45	0.49	0.46
	Class. Textual	0.66	0.49	0.52
	Descrições	0.44	0.58	0.48
TRANSFERENCIA-DE-EXECUCAO	Resumos	1.00	1.00	1.00
	Centroides	0.69	0.67	0.68
	Class. Textual	0.60	0.63	0.50
	Descrições	0.88	0.60	0.62
UNIFICACAO-DE-PENAS	Resumos	0.86	0.86	0.86
	Centroides	0.16	0.15	0.15
	Class. Textual	0.28	0.20	0.22
	Descrições	0.25	0.15	0.17
USUCAPIAO	Resumos	1.00	1.00	1.00
	Centroides	0.97	0.97	0.97
	Class. Textual	0.75	0.94	0.83
	Descrições	0.73	0.82	0.75

Tabela 5.4: *Desempenho médio por categoria para cada método de classificação, incluindo precisão, recall e F1-score - Parte II (Fonte: elaborado pelo autor)*

Categoria	**Precisão**	***Recall***	***F1-Score***
AGRAVO	0.99	1.00	1.00
ALVARA-JUDICIAL-LIBERACAO-DE-CORPO	1.00	1.00	1.00
APELACAO	1.00	1.00	1.00
CONSIGNACAO-EM-PAGAMENTO	1.00	1.00	1.00
CONTRARRAZOES-AO-AGRAVO	1.00	1.00	1.00
CUMPRIMENTO-DE-SENTENCA	1.00	1.00	1.00
DISSOLUCAO-DE-CONDOMINIO	1.00	1.00	1.00
EMBARGOS	1.00	1.00	1.00
EXCECAO-DE-PRE-EXECUTIVIDADE	1.00	1.00	1.00
EXTINCAO-DE-PUNIBILIDADE	1.00	0.85	0.92
HABEAS-CORPUS	1.00	1.00	1.00
IMPUGNACAO	1.00	1.00	1.00
INDENIZATORIAS	1.00	1.00	1.00
INDULTO-COMUTACAO	1.00	1.00	1.00
INTIMACAO-NEGATIVA	1.00	1.00	1.00
LIVRAMENTO-CONDICIONAL	1.00	1.00	1.00
OFICIOS	1.00	1.00	1.00
PROGRESSAO-DE-REGIME	1.00	1.00	1.00
REGISTRO-CIVIL	1.00	1.00	1.00
REMICAO-DE-PENA	1.00	1.00	1.00
SAIDA-TEMPORARIA	1.00	1.00	1.00
TRANSFERENCIA-DE-EXECUCAO	1.00	1.00	1.00
UNIFICACAO-DE-PENAS	1.00	1.00	1.00
USUCAPIAO	1.00	1.00	1.00

Tabela 5.5: *Resultados por categoria para o modelo Llama-3.1-70B-Instruct-Turbo com o método de resumos, mostrando precisão, recall e F1-score para cada categoria jurídica (Fonte: elaborado pelo autor)*

Categoria	**Precisão**	***Recall***	***F1-Score***
AGRAVO	1.00	1.00	1.00
ALVARA-JUDICIAL-LIBERACAO-DE-CORPO	1.00	1.00	1.00
APELACAO	1.00	1.00	1.00
CONSIGNACAO-EM-PAGAMENTO	1.00	1.00	1.00
CONTRARRAZOES-AO-AGRAVO	1.00	1.00	1.00
CUMPRIMENTO-DE-SENTENCA	0.96	1.00	0.98
DISSOLUCAO-DE-CONDOMINIO	1.00	1.00	1.00
EMBARGOS	1.00	1.00	1.00
EXCECAO-DE-PRE-EXECUTIVIDADE	1.00	0.98	0.99
EXTINCAO-DE-PUNIBILIDADE	1.00	1.00	1.00
HABEAS-CORPUS	1.00	1.00	1.00
IMPUGNACAO	0.00	0.00	0.00
INDENIZATORIAS	0.25	1.00	0.39
INDULTO-COMUTACAO	1.00	1.00	1.00
INTIMACAO-NEGATIVA	1.00	1.00	1.00
LIVRAMENTO-CONDICIONAL	1.00	1.00	1.00
OFICIOS	1.00	1.00	1.00
PROGRESSAO-DE-REGIME	1.00	1.00	1.00
REGISTRO-CIVIL	1.00	0.99	1.00
REMICAO-DE-PENA	1.00	1.00	1.00
SAIDA-TEMPORARIA	1.00	1.00	1.00
TRANSFERENCIA-DE-EXECUCAO	1.00	1.00	1.00
UNIFICACAO-DE-PENAS	1.00	1.00	1.00
USUCAPIAO	1.00	1.00	1.00

Tabela 5.6: *Resultados por categoria para o modelo Mixtral-8x22B-Instruct-v0.1 com o método de resumos, mostrando precisão, recall e F1-score para cada categoria jurídica (Fonte: elaborado pelo autor)*

Categoria	**Precisão**	***Recall***	***F1-Score***
AGRAVO	0.77	0.13	0.22
ALVARA-JUDICIAL-LIBERACAO-DE-CORPO	0.00	0.00	0.00
APELACAO	0.28	0.74	0.40
CONSIGNACAO-EM-PAGAMENTO	0.20	0.06	0.10
CONTRARRAZOES-AO-AGRAVO	0.00	0.00	0.00
CUMPRIMENTO-DE-SENTENCA	0.04	0.03	0.04
DISSOLUCAO-DE-CONDOMINIO	0.92	0.46	0.62
EMBARGOS	0.29	0.05	0.08
EXCECAO-DE-PRE-EXECUTIVIDADE	0.94	0.20	0.33
EXTINCAO-DE-PUNIBILIDADE	0.26	0.23	0.24
HABEAS-CORPUS	0.08	0.96	0.14
IMPUGNACAO	0.87	0.11	0.20
INDENIZATORIAS	0.44	0.43	0.43
INDULTO-COMUTACAO	0.17	0.40	0.24
INTIMACAO-NEGATIVA	0.11	0.04	0.06
LIVRAMENTO-CONDICIONAL	0.33	0.75	0.46
OFICIOS	0.83	0.97	0.89
PROGRESSAO-DE-REGIME	0.86	0.32	0.46
REGISTRO-CIVIL	0.95	0.23	0.37
REMICAO-DE-PENA	0.14	0.04	0.06
SAIDA-TEMPORARIA	0.01	0.08	0.01
TRANSFERENCIA-DE-EXECUCAO	1.00	0.07	0.13
UNIFICACAO-DE-PENAS	0.00	0.00	0.00
USUCAPIAO	0.62	1.00	0.76

Tabela 5.7: *Resultados por categoria para o modelo Llama-3.2-3B-Instruct-Turbo com o método de descrições, mostrando precisão, recall e F1-score para cada categoria jurídica*

CAPÍTULO 6

Conclusão

Os resultados experimentais confirmaram a eficácia dos LLMs na tarefa de classificar documentos jurídicos. Entre os modelos avaliados, o Llama-3.1-70B-Instruct-Turbo se destacou, alcançando as melhores pontuações de acurácia e *F1-score* ao utilizar abordagens baseadas em resumos. Isso demonstra a capacidade desse modelo de processar de forma eficiente documentos jurídicos complexos, mantendo um entendimento contextual profundo e detalhado.

No entanto, é importante reconhecer que a ausência de uma linha de base com métodos tradicionais de classificação de texto, como modelos supervisionados ou técnicas clássicas de PLN, limita a abrangência da comparação. Essa decisão justifica-se pela natureza complexa dos textos jurídicos e pela alta capacidade dos LLMs em lidar com dados textuais dessa magnitude. Estudos anteriores [69, 7, 14] demonstram que técnicas tradicionais, como Support Vector Machine (SVM) ou Naive Bayes, apesar de eficientes em cenários mais simples, provavelmente não alcançariam os os níveis de precisão exigidos em problemas de alta complexidade e variabilidade textual, como os apresentados neste estudo. Assim, optou-se por direcionar o foco diretamente ao uso de LLMs para garantir uma solução eficiente e robusta, considerando o custo computacional justificado pelos benefícios esperados.

O conjunto de dados utilizado neste estudo, composto por mais de 3.400 textos jurídicos em português, apresentou desafios únicos devido à sua complexidade e diversidade de categorias. Ao empregar LLMs como o Mixtral-8x22B-Instruct-v0.1 e o Llama-3.1-70B-Instruct-Turbo, o estudo demonstrou que o método de resumo pode melhorar o desempenho de classificação, capturando os aspectos essenciais dos documentos e minimizando detalhes irrelevantes. Comparações com modelos menores, como o Mixtral-7B-Instruct-v0.3 e o Mixtral-8x7B-Instruct-v0.1, também revelaram melhorias em termos de acurácia, precisão, *recall* e eficiência geral proporcionadas pelos LLMs de maior porte utilizando técnicas de representação textual através de resumos.

Além disso, este trabalho destaca a importância de considerar a viabilidade prática dos modelos de médio porte. Embora o Llama-3.1-70B tenha obtido pontuações altas, os custos computacionais elevados limitam sua aplicabilidade em larga escala em

instituições com recursos limitados. Nesse sentido, modelos de médio porte, como o Mixtral-8x22B, oferecem uma alternativa mais viável, mantendo um bom equilíbrio entre desempenho e custo.

Este estudo apresenta um conjunto de dados específicos de textos jurídicos em português, com experimentação realizada utilizando dados da DPE-GO. A pesquisa avalia como diferentes técnicas de processamento de texto, como resumos e centróides, podem ser aplicadas para melhorar os resultados de classificação. Os resultados sugerem que a integração dos LLMs em fluxos de trabalho jurídicos pode reduzir a carga de trabalho dos defensores públicos, automatizando e padronizando os processos de análise e classificação de documentos. Isso é relevante para instituições com recursos limitados, onde a eficiência e a consistência são importantes. As contribuições deste trabalho demonstram o potencial transformador da IA na prática jurídica, abrindo caminho para serviços jurídicos mais eficientes e acessíveis, por meio da integração de tecnologias de PLN e automação.

6.1 Limitações

Uma das principais limitações enfrentadas neste estudo foi o desequilíbrio de classes no conjunto de dados, que dificultou o desempenho adequado dos modelos para certas categorias de documentos jurídicos menos representadas. Esse desequilíbrio possivelmente influenciou os resultados, uma vez que os modelos tendem a apresentar melhor desempenho em classes mais presentes. Outro ponto desafiador foi a questão da anotação e padronização dos dados. Como muitos documentos jurídicos possuem informações sensíveis, como nomes de pessoas, datas e locais, a ausência de um processo de anotação robusto dificultou a identificação de padrões consistentes que poderiam ser úteis para a classificação. A implementação de um NER poderia ter enriquecido o conjunto de dados, agregando informações contextuais importantes e potencializando o desempenho dos modelos, mas sua aplicação prática e confiável em textos jurídicos se mostrou desafiadora e poderá demandar um trabalho futuro mais aprofundado.

Além disso, as restrições financeiras associadas ao uso de APIs para inferência limitaram a quantidade de experimentos possíveis. O alto custo de processamento de grandes volumes de *tokens* em modelos robustos impôs uma necessidade de otimização no número de testes e no tamanho dos lotes de dados processados, o que acabou afetando a profundidade das análises. Por exemplo, a análise dos custos adicionais de gerar resumos para cada documento ainda não foi contabilizada com precisão neste estudo, embora os benefícios observados, como a redução de *tokens* e a manutenção da precisão dos modelos, já indiquem que essa técnica pode trazer vantagens financeiras.

Outro ponto limitante foi o fato de que a alta acurácia alcançada por alguns dos LLMs pode ser um sinal de *overfitting*. Uma hipótese é que o conjunto de dados não foi suficientemente grande e os modelos podem estar se ajustando demais aos dados fornecidos, reduzindo sua capacidade de generalização para novos conjuntos de dados ou contextos distintos.

A variabilidade limitada dos textos jurídicos restringe a aplicabilidade dos resultados a outros contextos e jurisdições. O conjunto de dados utilizado inclui informações da DPE-GO, representando apenas uma pequena parcela do universo jurídico brasileiro, com suas particularidades. Assim, um estudo mais abrangente é necessário para adaptar e validar a eficácia desses modelos em diferentes áreas do direito e em sistemas jurídicos de outros países, que podem diferir substancialmente em terminologia e estrutura.

Outra limitação importante é a relação entre o treinamento prévio dos modelos e o idioma dos dados utilizados no estudo. Como os modelos foram acessados via APIs, o desempenho em português depende diretamente do corpus multilíngue usado no treinamento, o qual pode ter uma representação limitada de textos jurídicos em português. Essa restrição possivelmente influenciou os resultados observados, sobretudo em tarefas que exigem compreensão linguística avançada e terminologias específicas.

Por fim, os dados foram obtidos em formatos variados e desorganizados, com muitos documentos fora de suas categorias corretas, o que exigiu um trabalho para organização e estruturação do *dataset*. As etapas adicionais de aprovação e conformidade com a LGPD aumentaram o tempo necessário para preparação do conjunto de dados, incluindo categorização e organização cuidadosa dos textos, além de medidas para garantir a privacidade e segurança dos dados, conforme requisitos legais.

6.2 Trabalhos Futuros

Dadas as limitações observadas, diversos caminhos podem ser explorados em estudos futuros para expandir a validade e a aplicabilidade dos resultados. Primeiramente, uma expansão do conjunto de dados é fundamental para avaliar e mitigar o potencial *overfitting* em modelos maiores. Ampliar a base de documentos e diversificar as categorias jurídicas permitiria uma avaliação mais precisa da generalização dos LLMs e poderia fornecer informações adicionais sobre a capacidade dos modelos de lidar com dados mais variados.

Outra alternativa promissora é a geração de dados sintéticos para balancear as classes sub-representadas, o que poderia minimizar o *viés* nos resultados e melhorar a robustez do modelo. Além disso, a implementação de técnicas de NER poderia enriquecer os dados, permitindo que o modelo identifique automaticamente informações relevantes,

como nomes, datas e locais, o que agregaria contexto e ajudaria a aumentar a precisão da classificação.

Para validar a eficácia dos LLMs em outros contextos, é recomendável testá-los em diferentes jurisdições e áreas do direito, como direito trabalhista ou empresarial. Esse teste em novas áreas poderia identificar demandas específicas e permitir a adaptação dos modelos a novos cenários. Além disso, adaptar o modelo para outros sistemas jurídicos fora do Brasil, seria útil para explorar sua capacidade de generalização internacional.

Trabalhos futuros também podem investigar o custo-benefício de diferentes técnicas de sumarização automática, comparando seus custos e impactos no desempenho dos modelos para identificar a abordagem mais eficiente.

A exploração de *transfer learning* também merece destaque, já que permite que os modelos aprendam a partir de domínios adjacentes antes de serem ajustados para a classificação jurídica específica. Essa técnica pode ajudar a melhorar o desempenho em classes de dados escassos, facilitando uma melhor adaptação dos modelos. Experimentos com outros modelos de médio porte, também poderiam ser realizados como uma alternativa econômica, explorando ajustes para maximizar a eficiência sem aumentar substancialmente os custos.

Por último, uma aplicação prática seria a avaliação em outras famílias de modelos e, principalmente, com modelos menores, permitindo o uso de suas capacidades de classificação de documentos diretamente em dispositivos móveis. Isto ampliaria o uso prático da IA no dia a dia jurídico, facilitando o acesso a advogados, defensores públicos e outros profissionais do setor.

Referências

[1] AGUIAR, A.; SILVEIRA, R.; PINHEIRO, V.; FURTADO, V.; NETO, J. A. **Text classification in legal documents extracted from lawsuits in brazilian courts**. In: *Anais da X Brazilian Conference on Intelligent Systems*, Porto Alegre, RS, Brasil, 2021. SBC.

[2] AHMAD AGHAEBRAHIMIAN, M. C. **Hyperparameter tuning for deep learning in natural language processing**, 2019.

[3] ANONYMOUS. **One law, many languages: Benchmarking multilingual legal reasoning for judicial support**. In: *ICLR 2024 Workshop on Data-centric Machine Learning Research (DMLR): Harnessing Momentum for Science*, 2024.

[4] BERNSOHN, D.; SEMO, G.; VAZANA, Y.; HAYAT, G.; HAGAG, B.; NIKLAUS, J.; SAHA, R.; TRUSKOVSKYI, K. **Legallens: Leveraging llms for legal violation identification in unstructured text**, 2024.

[5] BHAWSAR, S.; DUBEY, S.; KUSHWAHA, S.; SHARMA, S. **Text classification using deep learning: A survey**. In: *Proceedings of International Conference on Computational Intelligence: ICCI 2021*, p. 205–216. Springer, 2022.

[6] BOSTROM, K.; DURRETT, G. **Byte pair encoding is suboptimal for language model pretraining**. *arXiv preprint arXiv:2004.03720*, 2020.

[7] BROWN, T. B. **Language models are few-shot learners**. *arXiv preprint arXiv:2005.14165*, 2020.

[8] CHAI, C. P. **Comparison of text preprocessing methods**. *Natural Language Engineering*, 29(3):509–553, 2023.

[9] CHAI, Y.; ZHANG, H.; JIN, S. **Neural text classification by jointly learning to cluster and align**, 2020.

[10] CHEN, Y.; LIU, Y.; DONG, L.; WANG, S.; ZHU, C.; ZENG, M.; ZHANG, Y. **Adaprompt: Adaptive model training for prompt-based nlp**. *arXiv preprint arXiv:2202.04824*, 2022.

[11] CHURCH, K. W. **Word2vec**. *Natural Language Engineering*, 23(1):155–162, 2017.

[12] COLOMBO, P.; PIRES, T. P.; BOUDIAF, M.; CULVER, D.; MELO, R.; CORRO, C.; MARTINS, A. F. T.; ESPOSITO, F.; RAPOSO, V. L.; MORGADO, S.; DESA, M. **Saullm-7b: A pioneering large language model for law**, 2024.

[13] DE JESUS FALCÃO, L. C.; OTHERS. **Sumarização de texto em deep learning como etapa inicial para a construção de um modelo de recuperação da informação: análise do setor de mineração no brasil**, 2024.

[14] DEVLIN, J. **Bert: Pre-training of deep bidirectional transformers for language understanding**. *arXiv preprint arXiv:1810.04805*, 2018.

[15] DOGRA, V.; VERMA, S.; KAVITA.; CHATTERJEE, P.; SHAFI, J. **A complete process of text classification system using state-of-the-art nlp models**, 2022.

[16] DPE-GO. http://www2.defensoria.go.def.br/, 2024.

[17] EDWARDS, A.; CAMACHO-COLLADOS, J. **Language models for text classification: Is in-context learning enough?**, 2024.

[18] ELOV, B.; KHAMROEVA, S. M.; XUSAINOVA, Z. **The pipeline processing of nlp**. In: *E3S Web of Conferences*, volume 413, p. 03011. EDP Sciences, 2023.

[19] FENG, Y.; LI, C.; NG, V. **Legal case retrieval: A survey of the state of the art**. In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 6472–6485, 2024.

[20] FIELDS, J.; CHOVANEC, K.; MADIRAJU, P. **A survey of text classification with transformers: How wide? how large? how long? how accurate? how expensive? how safe?** *IEEE Access*, 2024.

[21] FIELDS, J.; CHOVANEC, K.; MADIRAJU, P. **A survey of text classification with transformers: How wide? how large? how long? how accurate? how expensive? how safe?** *IEEE Access*, 12:6518–6531, 2024.

[22] GULTEKIN, S.; GLOBO, A.; ZUGARINI, A.; ERNANDES, M.; RIGUTINI, L. **An energy-based comparative analysis of common approaches to text classification in the legal domain**. In: *AI, Machine Learning and Applications*, AIMLA, p. 31–41. Academy & Industry Research Collaboration Center, Jan. 2024.

[23] HAQ, M. A.; KHAN, M. A. R.; ALSHEHRI, M. **Insider threat detection based on nlp word embedding and machine learning**. *Intell. Autom. Soft Comput*, 33(1):619–635, 2022.

[24] HUI LIU, QINGYU YIN, W. Y. W. **Towards explainable nlp: A generative explanation framework for text classification**, 2018.

[25] HUTCHINS, J. **A prior case study of natural language processing on different domain**, 2014.

[26] J., SHRUTHI, S. S. **A prior case study of natural language processing on different domain**, 2020.

[27] JIANG, C.; YANG, X. **Legal syllogism prompting: Teaching large language models for legal judgment prediction**. In: *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, ICAIL '23, p. 417–421, New York, NY, USA, 2023. Association for Computing Machinery.

[28] JIANG, X.; LI, X.; MA, W.; FANG, X.; YAO, Y.; YU, N.; MENG, X.; HAN, P.; LI, J.; SUN, A.; WANG, Y. **Sketch: A toolkit for streamlining llm operations**, 2024.

[29] KANHAIYA, K.; NAVEEN.; SHARMA, A. K.; GAUTAM, K.; RATHORE, P. S. **Ai enabled-information retrival engine (ai-ire) in legal services: An expert-annotated nlp for legal judgements**. In: *2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, p. 206–210, 2023.

[30] KATARIA, H.; GUPTA, A. **NLP-titan at SemEval-2023 task 6: Identification of rhetorical roles using sequential sentence classification**. In: Ojha, A. K.; Doğruöz, A. S.; Da San Martino, G.; Tayyar Madabushi, H.; Kumar, R.; Sartori, E., editors, *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, p. 1365–1370, Toronto, Canada, July 2023. Association for Computational Linguistics.

[31] KATZ, D. M.; HARTUNG, D.; GERLACH, L.; JANA, A.; AU2, M. J. B. I. **Natural language processing in the legal domain**, 2023.

[32] KHAN, A. A. **Balanced split: A new train-test data splitting strategy for imbalanced datasets**, 2022.

[33] KIESOW CORTEZ, E.; MASLEJ, N. **Adjudication of artificial intelligence and automated decision-making cases in europe and the usa**. *European Journal of Risk Regulation*, 14(3):457–475, 2023.

[34] KOJIMA, T.; GU, S. S.; REID, M.; MATSUO, Y.; IWASAWA, Y. **Large language models are zero-shot reasoners**. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[35] KRUMOV, K.; BOYTCHEVA, S.; KOYTCHEV, I. **SU-FMI at SemEval-2024 task 5: From BERT fine-tuning to LLM prompt engineering - approaches in legal argument reasoning**. In: Ojha, A. K.; Doğruöz, A. S.; Tayyar Madabushi, H.; Da San Martino, G.; Rosenthal, S.; Rosá, A., editors, *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, p. 1652–1658, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[36] KUDO, T. **Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing**. *arXiv preprint arXiv:1808.06226*, 2018.

[37] KUTBI, M. **Named entity recognition utilized to enhance text classification while preserving privacy**. *IEEE Access*, 11:117576–117581, 2023.

[38] LEVIN, K.; HENRY, K.; JANSEN, A.; LIVESCU, K. **Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings**. In: *2013 IEEE workshop on automatic speech recognition and understanding*, p. 410–415. IEEE, 2013.

[39] LIU, W.; PANG, J.; LI, N.; YUE, F.; LIU, G. **Few-shot short-text classification with language representations and centroid similarity**. *Applied Intelligence*, 53(7):8061–8072, 2023.

[40] LLAMA-3.2-3B. **Llama-3.2-3b technical report**. https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct, 2024.

[41] MA, X.; WANG, L.; YANG, N.; WEI, F.; LIN, J. **Fine-tuning llama for multi-stage text retrieval**, 2023.

[42] MAHENDRAN, D.; LUO, C.; MCINNES, B. T. **Review: Privacy-preservation in the context of natural language processing**. *IEEE Access*, 9:147600–147612, 2021.

[43] MARTIN, L.; WHITEHOUSE, N.; YIU, S.; CATTERSON, L.; PERERA, R. **Better call gpt, comparing large language models against lawyers**. *arXiv preprint arXiv:2401.16212*, 2024.

[44] META. **Llama 3 model card**. https://github.com/meta-llama/llama3/tree/main, 2024.

[45] META. **Meta ai introduces llama 3.1: Advanced capabilities in language modeling**. https://ai.meta.com/blog/meta-llama-3-1/?utm_source=twitter&utm_medium=organic_social&utm_content=video&utm_campaign=llama31&s=08, 2024.

[46] MILLS, M.; UEBERGANG, J. **Artificial intelligence in law: An overview**, 2017.

[47] MIXTRAL 7B. **Mistral-7b-instruct-v0.3: Advanced instruction following**. https://build.nvidia.com/mistralai/mistral-7b-instruct-v03, 2024.

[48] MIXTRAL 8X22B. **Mixtral-8x22b technical report**. https://huggingface.co/mistralai/Mixtral-8x22B-Instruct-v0.1, 2024.

[49] MIXTRAL 8X7B. **Mixtral-8x7b technical report**. https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1, 2024.

[50] MODRUŠAN, N.; RABUZIN, K.; MRSIC, L. **Improving public sector efficiency using advanced text mining in the procurement process**, 01 2020.

[51] MONTEJO-RÁEZ, A.; JIMÉNEZ-ZAFRA, S. M. **Current approaches and applications in natural language processing**. *Applied Sciences*, 12(10):4859, 2022.

[52] MORAES, L. D. C.; SILVÉRIO, I. C.; MARQUES, R. A. S.; ANAIA, B. D. C.; DE PAULA, D. F.; DE FARIA, M. C. S.; CLEVESTON, I.; CORREIA, A. D. S.; FREITAG, R. M. K. **Análise de ambiguidade linguística em modelos de linguagem de grande escala (llms)**. *arXiv preprint arXiv:2404.16653*, 2024.

[53] MU, J.; BHAT, S.; VISWANATH, P. **All-but-the-top: Simple and effective postprocessing for word representations**. *arXiv preprint arXiv:1702.01417*, 2017.

[54] NASEEM, U.; RAZZAK, I.; KHAN, S. K.; PRASAD, M. **A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models**, 2021.

[55] NING, L.; LIU, L.; WU, J.; WU, N.; BERLOWITZ, D.; PRAKASH, S.; GREEN, B.; O'BANION, S.; XIE, J. **User-llm: Efficient llm contextualization with user embeddings**. *arXiv preprint arXiv:2402.13598*, 2024.

[56] NONATO, L. G. **O cenário regulatório da inteligência artificial**, 2022.

[57] O POPULAR. **Goiás tem o segundo maior déficit de defensores públicos do país**. https://opopular.com.br/goias-tem-o-segundo-maior-deficit-de-defensores-publicos-do-pais-1.1980613, 2023.

[58] OGLETREE, C. J. **An essay on the new public defender for the 21st century**. *Law and Contemporary Problems*, 58(1):81–93, 1995.

[59] PAL, A.; RAJANALA, S.; PHAN, R. C.-W.; WONG, K. **Self supervised bert for legal text classification**. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 1–5, 2023.

[60] PALANIVINAYAGAM, A.; EL-BAYEH, C. Z.; DAMAŠEVIČIUS, R. **Twenty years of machine-learning-based text classification: A systematic review**. *Algorithms*, 16(5), 2023.

[61] PANGAKIS, N.; WOLKEN, S. **Knowledge distillation in automated annotation: Supervised text classification with llm-generated training labels**, 2024.

[62] PAULUCIO, L. S. **Categorização automática de produtos utilizando apenas o título e aprendizado profundo**, 2022.

[63] PICHIYAN, V.; MUTHULINGAM, S.; G, S.; NALAJALA, S.; CH, A.; DAS, M. N. **Web scraping using natural language processing: Exploiting unstructured text for data extraction and analysis**. *Procedia Computer Science*, 230:193–202, 2023. 3rd International Conference on Evolutionary Computing and Mobile Sustainable Networks (ICECMSN 2023).

[64] PRASAD, N.; BOUGHANEM, M.; DKAKI, T. **Exploring large language models and hierarchical frameworks for classification of large unstructured legal documents**. In: *European Conference on Information Retrieval*, p. 221–237. Springer, 2024.

[65] PRASAD, N.; DKAKI, T.; BOUGHANEM, M. **Explanation extraction from hierarchical classification frameworks for long legal documents**. In: *Findings of the Association for Computational Linguistics: NAACL 2024*, p. 1192–1201, 2024.

[66] PROTTASHA, N. J.; SAMI, A. A.; KOWSHER, M.; MURAD, S. A.; BAIRAGI, A. K.; MASUD, M.; BAZ, M. **Transfer learning for sentiment analysis using bert based supervised fine-tuning**. *Sensors*, 22(11):4157, 2022.

[67] QADER, W. A.; AMEEN, M. M.; AHMED, B. I. **An overview of bag of words; importance, implementation, applications, and challenges**. In: *2019 international engineering conference (IEC)*, p. 200–204. IEEE, 2019.

[68] QWEN2. **Qwen2 technical report**. https://huggingface.co/Qwen/Qwen2-72B-Instruct, 2024.

[69] RADFORD, A.; WU, J.; CHILD, R.; LUAN, D.; AMODEI, D.; SUTSKEVER, I.; OTHERS. **Language models are unsupervised multitask learners**. *OpenAI blog*, 1(8):9, 2019.

[70] RAFFEL, C.; SHAZEER, N.; ROBERTS, A.; LEE, K.; NARANG, S.; MATENA, M.; ZHOU, Y.; LI, W.; LIU, P. J. **Exploring the limits of transfer learning with a unified text-to-text transformer**, 2023.

[71] RAPÔSO, C. F. L.; DE LIMA, H. M.; DE OLIVEIRA JUNIOR, W. F.; SILVA, P. A. F.; DE SOUZA BARROS, E. E. **Lgpd-lei geral de proteção de dados pessoais em tecnologia da informação: Revisão sistemática**. *RACE-Revista de Administração do Cesmac*, 4:58–67, 2019.

[72] ROUZEGAR, H.; MAKREHCHI, M. **Enhancing text classification through LLM-driven active learning and human annotation**. In: Henning, S.; Stede, M., editors, *Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII)*, p. 98–111, St. Julians, Malta, Mar. 2024. Association for Computational Linguistics.

[73] RUAN, Q.; KUZNETSOV, I.; GUREVYCH, I. **Are large language models good classifiers? a study on edit intent classification in scientific document revisions**. *arXiv preprint arXiv:2410.02028*, 2024.

[74] SILVEIRA, M. C. **Named entity recognition**. *Named Entity Recognition-ResearchGate*, 50(5):807–819, 2014.

[75] SONG, C. H.; WU, J.; WASHINGTON, C.; SADLER, B. M.; CHAO, W.-L.; SU, Y. **Llm-planner: Few-shot grounded planning for embodied agents with large language models**. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, p. 2998–3009, 2023.

[76] SUN, X.; LI, X.; LI, J.; WU, F.; GUO, S.; ZHANG, T.; WANG, G. **Text classification via large language models**, 2023.

[77] SUNSTEIN, C. R. **Of artificial intelligence and legal reasoning**. *U. Chi. L. Sch. Roundtable*, 8:29, 2001.

[78] TOMAR, M. S.; GUPTA, V. **Legal case classification using machine learning with nlp**. In: *2023 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, p. 1–6, 2023.

[79] TONG XIAO, J. Z. **Introduction to transformers: an nlp perspective**, 2023.

[80] TORFI, A.; SHIRVANI, R. A.; KENESHLOO, Y.; TAVAF, N.; FOX, E. A. **Natural language processing advancements by deep learning: A survey**, 2020.

[81] TOUVRON, H.; LAVRIL, T.; IZACARD, G.; MARTINET, X.; LACHAUX, M.-A.; LACROIX, T.; ROZIÈRE, B.; GOYAL, N.; HAMBRO, E.; AZHAR, F.; RODRIGUEZ, A.; JOULIN, A.;

GRAVE, E.; LAMPLE, G. **Llama: Open and efficient foundation language models**, 2023.

[82] VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; POLOSUKHIN, I. **Attention is all you need**, 2023.

[83] VIJAYARANI, S.; ILAMATHI, M. J.; NITHYA, M.; OTHERS. **Preprocessing techniques for text mining-an overview**. *International Journal of Computer Science & Communication Networks*, 5(1):7–16, 2015.

[84] WAN, L.; PAPAGEORGIOU, G.; SEDDON, M.; BERNARDONI, M. **Long-length legal document classification**. *arXiv preprint arXiv:1912.06905*, 2019.

[85] WAN, L.; PAPAGEORGIOU, G.; SEDDON, M.; BERNARDONI, M. **Long-length legal document classification**, 2019.

[86] WANG, Z.; PANG, Y.; LIN, Y. **Large language models are zero-shot text classifiers**, 2023.

[87] WEI, F.; KEELING, R.; HUBER-FLIFLET, N.; ZHANG, J.; DABROWSKI, A.; YANG, J.; MAO, Q.; QIN, H. **Empirical study of llm fine-tuning for text classification in legal document review**. In: *2023 IEEE International Conference on Big Data (BigData)*, p. 2786–2792, 2023.

[88] WU, J.; LIU, X.; LI, M.; LI, W.; SU, Z.; LIN, S.; GARAY, L.; ZHANG, Z.; ZHANG, Y.; ZENG, Q.; OTHERS. **Clinical text datasets for medical artificial intelligence and large language models—a systematic review**. *NEJM AI*, 1(6):AIra2400012, 2024.

[89] XIE, Y.; LI, Z.; YIN, Y.; WEI, Z.; XU, G.; LUO, Y. **Advancing legal citation text classification a conv1d-based approach for multi-class classification**. *Journal of Theory and Practice of Engineering Science ISSN*, 2790:1513, 2024.

[90] XU, S.; ZHANG, C.; HONG, D. **Bert-based nlp techniques for classification and severity modeling in basic warranty data study**. *Insurance: Mathematics and Economics*, 107:57–67, 2022.

[91] YANN LECUN, YOSHUA BENGIO, G. H. **Deep learning**, 2015.

[92] YUN-TAO, Z.; LING, G.; YONG-CHENG, W. **An improved tf-idf approach for text classification**. *Journal of Zhejiang University-Science A*, 6(1):49–55, 2005.

[93] ZHANG, Y.; WANG, M.; REN, C.; LI, Q.; TIWARI, P.; WANG, B.; QIN, J. **Pushing the limit of llm capacity for text classification**, 2024.

[94] ZHANG, Z.; HU, X.; ZHANG, J.; ZHANG, Y.; WANG, H.; QU, L.; XU, Z. **FEDLEGAL: The first real-world federated learning benchmark for legal NLP**. In: Rogers, A.; Boyd-Graber, J.; Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 3492–3507, Toronto, Canada, July 2023. Association for Computational Linguistics.

[95] ZHOU, H. **Research of text classification based on tf-idf and cnn-lstm**. *Journal of Physics: Conference Series*, 2171(1):012021, jan 2022.

APÊNDICE A

Apêndice 1

A.1 *Prompt* para Classificação

```
PROMPT_CLASSIFICACAO = (
    "Classifique as peticoes processuais abaixo em uma das
       seguintes categorias: "
    "AGRAVO, CONTRARRAZOES-AO-AGRAVO, EMBARGOS, EXTINCAO-DE-
       PUNIBILIDADE, "
    "HABEAS-CORPUS, IMPUGNACAO, INDULTO-COMUTACAO, INTIMACAO-
       NEGATIVA, "
    "LIVRAMENTO-CONDICIONAL, OFICIOS, PROGRESSAO-DE-REGIME,
       REMICAO-DE-PENA, "
    "TRANSFERENCIA-DE-EXECUCAO, UNIFICACAO-DE-PENAS, APELACAO
       , REGISTRO-CIVIL, "
    "CUMPRIMENTO-DE-SETENCA, INDENIZATORIAS, EXCECAO-DE-PRE-
       EXECUTIVIDADE, "
    "DISSOLUCAO-DE-CONDOMINIO, CONSIGNACAO-EM-PAGAMENTO, "
    "ALVARA-JUDICIAL-LIBERACAO-DE-CORPO, SAIDA-TEMPORARIA,
       USUCAPIAO. "
    "Retorne apenas a categoria e nada mais. "
    "Texto: {texto} "
    "Categoria:"
)
```

A.2 *Prompt* para Resumo

```
PROMPT_RESUMO = (
    "Resuma a peticao abaixo em 200 palavras, focando na
       ideia geral do texto. Ignore detalhes especificos como
        nomes, numeros e locais "
    "que nao definem o conceito central. A classificacao deve
        ser entre as seguintes categorias: AGRAVO,
       CONTRARRAZOES-AO-AGRAVO, EMBARGOS, EXTINCAO-DE-
       PUNIBILIDADE, "
    "HABEAS-CORPUS, IMPUGNACAO, INDULTO-COMUTACAO, INTIMACAO-
       NEGATIVA, LIVRAMENTO-CONDICIONAL, OFICIOS, PROGRESSAO-
       DE-REGIME, REMICAO-DE-PENA, "
    "TRANSFERENCIA-DE-EXECUCAO, UNIFICACAO-DE-PENAS, APELACAO
       , REGISTRO-CIVIL, CUMPRIMENTO-DE-SETENCA,
       INDENIZATORIAS, EXCECAO-DE-PRE-EXECUTIVIDADE, "
    "DISSOLUCAO-DE-CONDOMINIO, CONSIGNACAO-EM-PAGAMENTO,
       ALVARA-JUDICIAL-LIBERACAO-DE-CORPO, SAIDA-TEMPORARIA,
       USUCAPIAO. "
    "Escreva o resumo em portugues e retorne apenas o resumo
       explicativo. "
    "----------------------\n"
    "Peticao de {classe}: {texto}\n"
    "Resumo Explicativo:"
)
```

O *prompt* para geração de resumos instrui o modelo a produzir um texto conciso de até 200 palavras, capturando a essência do documento original e ignorando detalhes irrelevantes, como nomes e números. Esses resumos são armazenados na variável `resumos_df`, que organiza cada resumo em associação à sua categoria jurídica correspondente, coluna (`issue_area`) do *dataset* de avaliação. Esse *dataframe* torna-se uma peça central no *pipeline*, pois serve como referência para a classificação de textos futuros.

Quando textos novos são introduzidos no *pipeline*, eles são carregados no *dataframe* `evaluation_df`. Este *dataframe* contém os textos a serem classificados, junto com suas categorias reais, coluna (`issue_area`) do *dataset* de classificação. O *pipeline* processa cada texto novo iterativamente. Para cada linha de `evaluation_df`, o *pipeline* verifica a categoria jurídica associada ao texto novo e utiliza essa informação para recuperar o resumo correspondente da variável `resumos_df`. Esse resumo é recuperado com o seguinte código:

```
resumo_text = resumos_df[resumos_df['issue_area'] == row['
    issue_area']]['resumo_text'].values[0]
```

O resumo correspondente é então utilizado como entrada no modelo de linguagem para classificação. O *prompt* de classificação (`PROMPT_CLASSIFICACAO`) instrui o modelo a categorizar o texto fornecido, que neste caso é o resumo, em uma das categorias jurídicas predefinidas. O *pipeline* configura o *prompt* e passa o resumo ao modelo com o seguinte trecho de código:

```
result = chain_classificacao.invoke({'texto': resumo_text}).
    strip()
```

Aqui, o resumo recuperado de `resumos_df` substitui o campo texto no *prompt*, e o modelo analisa o conteúdo do resumo para determinar a categoria jurídica correspondente. Importante destacar que o texto novo em si não é diretamente comparado ao resumo. Em vez disso, o resumo é usado como uma entrada simplificada e representativa para orientar a predição do modelo. Isso reduz os custos computacionais, já que o modelo processa apenas os resumos curtos, e não os textos jurídicos completos.

Após a predição, o resultado retornado pelo modelo é normalizado para garantir que diferenças de formatação, como acentos ou uso de maiúsculas e minúsculas, não prejudiquem a avaliação. O rótulo previsto é então comparado ao rótulo real (`issue_area`) do texto novo para calcular métricas de desempenho, como precisão, *recall* e *F1-score*. Esse processo é iterado para cada texto novo no *dataframe* `evaluation_df`, e os resultados são armazenados para análises posteriores com o seguinte código:

```
results.append({
    'Real': row['issue_area'],
    'Previsto': result,
    'Correto': is_correct
})
```

No caso de textos novos sem rótulos reais conhecidos (dados não rotulados), o *pipeline* pode ser ajustado para passar o texto completo do documento ao modelo diretamente, substituindo o uso do resumo. Isso é feito alterando o campo texto no *prompt* para conter o texto novo completo, como mostrado a seguir:

```
result = chain_classificacao.invoke({'texto': row['facts']}).
    strip()
```

Essa abordagem permite classificar textos desconhecidos, mas tem um custo computacional maior, por exemplo para textos longos.

A.3 Geração dos Resumos a Partir dos Textos Originais

```
# Vetorizacao com TF-IDF
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(df['facts'])

# Extraindo os rotulos unicos
labels = df['issue_area'].unique()
selected_examples = []

# Calculando os centroides para cada rotulo
for label in labels:
    indices = df['issue_area'] == label
    cluster_vectors = X[indices]
    cluster_texts = df['facts'][indices]

    # Calculando o vetor centroide
    centroid_vector = cluster_vectors.mean(axis=0)
    centroid_vector = np.asarray(centroid_vector).flatten()

    # Calculando as distancias e encontrando o texto mais
       proximo do centroide
    distances = cosine_distances(cluster_vectors,
       centroid_vector.reshape(1, -1))
    closest_index = np.argmin(distances)
    centroid_text = cluster_texts.iloc[closest_index]

    # Armazenando o exemplo mais proximo ao centroide
    example = df[df['facts'] == centroid_text].iloc[0]
    selected_examples.append(example)

# Convertendo para DataFrame para processar
selected_df = pd.DataFrame(selected_examples)

# Geracao dos resumos a partir dos textos selecionados para
   cada categoria
prompt_template_resumo = PromptTemplate(
    input_variables=["texto", "classe"],
    template=PROMPT_RESUMO,
```

```
)
```

O processo começa com a vetorização dos textos, onde cada documento é transformado em um vetor TF-IDF utilizando a função `TfidfVectorizer()`. Essa técnica converte os textos em uma representação numérica que destaca a importância relativa de cada termo em relação ao *corpus* completo, facilitando a análise de similaridade entre os documentos.

Em seguida, para cada categoria já presente no *dataset* (definida pela coluna `issue_area`), calcula-se o centroide, que representa a média dos vetores TF-IDF dos documentos pertencentes à mesma categoria. Esse vetor sintetiza as características principais dos textos dentro da categoria, funcionando como uma "representação central".

A próxima etapa é identificar os documentos mais representativos de cada categoria. Para isso, calcula-se a distância coseno entre o vetor TF-IDF de cada documento e o centroide da categoria correspondente. A distância coseno mede a similaridade entre vetores, onde valores mais baixos indicam maior proximidade. O documento com a menor distância ao centroide é selecionado como o mais representativo da categoria, pois é o que mais reflete o "padrão médio"dos textos do grupo.

Os documentos selecionados (os mais próximos ao centroide) são armazenados em uma lista e posteriormente convertidos em um *DataFrame*. Esses textos encapsulam as características principais de suas categorias e servem como base para tarefas subsequentes, como a geração de resumos. Assim, o código identifica textos que melhor representam cada categoria, otimizando o processo de análise e geração de informações concisas.

A.4 Cálculo e Seleção dos Documentos Centroides

```
# Vetorizacao com TF-IDF
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(filtered_df['facts'])

# Extraindo os rotulos unicos
labels = filtered_df['issue_area'].unique()
centroids = []

# Calculando os centroides para cada rotulo
for label in labels:
    indices = filtered_df['issue_area'] == label
    cluster_vectors = X[indices]
    cluster_texts = filtered_df['facts'][indices]

    # Calculando o vetor centroide
    centroid_vector = cluster_vectors.mean(axis=0)
    centroid_vector = np.asarray(centroid_vector).flatten()

    # Calculando as distancias e encontrando o texto mais
       proximo do centroide
    distances = cosine_distances(cluster_vectors,
       centroid_vector.reshape(1, -1))
    closest_index = np.argmin(distances)
    centroid_text = cluster_texts.iloc[closest_index]

    # Armazenando o texto do centroide
    centroids.append({'issue_area': label, 'centroid_text':
       centroid_text})

# Criando DataFrame com os centroides
centroids_df = pd.DataFrame(centroids)

# Exibindo as centroides no terminal
print("Centroides:")
print(centroids_df)
```

O processo de cálculo dos centroides inicia-se com a vetorização dos textos jurídicos utilizando a técnica TF-IDF. Isso é implementado no código por meio da

função `TfidfVectorizer()`, onde o *corpus* completo (representado pela coluna *facts* do *DataFrame* `filtered_df`) é transformado em uma matriz TF-IDF. Cada linha da matriz representa um texto e cada coluna contém o peso de uma palavra com base em sua relevância dentro do *corpus*. O resultado dessa vetorização é armazenado na variável `X`.

A seguir, os centroides são calculados para cada categoria jurídica. Para cada rótulo único (`issue_area`), o código seleciona os vetores TF-IDF dos textos pertencentes àquela categoria. O vetor centroide é então definido como a média desses vetores, representando o "padrão central"dos textos daquela categoria. Isso é realizado no trecho:

```
centroid_vector = cluster_vectors.mean(axis=0)
```

Após calcular o vetor centroide, o código identifica o documento mais representativo da categoria, ou seja, o texto cujo vetor TF-IDF está mais próximo do centroide, com base na menor distância cosseno. Isso é feito por meio da função `cosine_distances`:

```
distances = cosine_distances(cluster_vectors, centroid_vector
   .reshape(1, -1))
closest_index = np.argmin(distances)
centroid_text = cluster_texts.iloc[closest_index]
```

Esse texto, denominado documento centroide, é armazenado no *DataFrame* `centroids_df`, juntamente com o rótulo da categoria correspondente (`issue_area`). O *DataFrame* `centroids_df` se torna, então, uma coleção estruturada de textos que encapsulam as características centrais de cada categoria.

Após a identificação e armazenamento dos documentos centroides, eles são utilizados diretamente no processo de classificação. O `PROMPT_CLASSIFICACAO` desempenha um papel importante nesse contexto, pois fornece as instruções para o modelo de linguagem realizar a classificação com base no texto fornecido. O trecho relevante do *prompt* é:

```
    "Retorne apenas a categoria e nada mais. "
    "Texto: {texto} "
    "Categoria:"
```

Aqui, o `texto` é substituído pelo texto do centroide correspondente à categoria que está sendo avaliada. Esse texto serve como entrada para o LLM e atua como uma referência clara e representativa da categoria jurídica.

No código, isso é implementado na etapa de classificação, onde para cada novo texto (`evaluation_df`), o centroide correspondente à categoria real do texto é recuperado:

```
centroid_text = centroids_df[centroids_df['issue_area'] ==
    row['issue_area']]['centroid_text'].values[0]
```

O texto do centroide é então incorporado no *prompt* e enviado ao modelo para que ele retorne a categoria apropriada:

```
result = chain_classificacao.invoke({'texto': centroid_text})
    .strip()
```

O texto centroide atua como um exemplo central da categoria e ajuda o modelo a compreender melhor as características da classe durante o processo de classificação. O modelo utiliza o texto centroide como entrada para tomar uma decisão sobre a categoria mais adequada para o novo texto.

A.5 *Prompt* para Descrição

```
PROMPT_DESCRICAO = (
    "Classifique a peticao processual nas seguintes
       categorias: "
    "- AGRAVO: Peticoes que contestam decisoes judiciais,
       solicitando a revisao dessas decisoes, geralmente por
       inconformidade com a decisao proferida em primeira
       instancia. "
    "- CONTRARRAZOES-AO-AGRAVO: Peticoes que apresentam
       argumentos contrarios a um agravo interposto,
       defendendo a manutencao da decisao judicial atacada e
       refutando os pontos levantados pelo agravante. "
    "- EMBARGOS: Peticoes que visam esclarecer ou corrigir
       omissoes, contradicoes ou obscuridades em decisoes
       judiciais, buscando a perfeita compreensao e aplicacao
        da decisao. "
    "- EXTINCAO-DE-PUNIBILIDADE: Peticoes que tratam da
       extincao da punibilidade de uma pena por razoes como
       prescricao, anistia, perdao judicial ou cumprimento
       integral da pena imposta. "
    "- HABEAS-CORPUS: Peticoes que visam garantir a liberdade
        de um individuo, alegando prisao ilegal ou abuso de
       poder, e buscando a concessao do beneficio de soltura
       imediata. "
    "- IMPUGNACAO: Peticoes que questionam a precisao de
       relatorios ou decisoes processuais, solicitando
       correcoes especificas para assegurar a justica e a
       veracidade dos documentos processuais. "
    "- INDULTO-COMUTACAO: Peticoes que solicitam a comutacao
       ou reducao de pena com base em decretos presidenciais,
        leis especificas ou outros fundamentos legais que
       justifiquem o beneficio. "
    "- INTIMACAO-NEGATIVA: Peticoes que tratam da intimacao
       do sentenciado que nao foi localizado, solicitando
       nova tentativa de intimacao ou a adocao de outras
       medidas para assegurar a comunicacao processual. "
    "- LIVRAMENTO-CONDICIONAL: Peticoes que solicitam a
       concessao de livramento condicional, alegando que o
```

```
        apenado cumpriu os requisitos legais, como bom
        comportamento e cumprimento de parte da pena. "
    "- OFICIOS: Solicitacoes formais de providencias ou
        informacoes sobre o andamento de processos penais,
        destacando a importancia da celeridade e da razoavel
        duracao do processo para a garantia dos direitos dos
        envolvidos. "
    "- PROGRESSAO-DE-REGIME: Peticoes que solicitam a
        progressao de regime prisional para um apenado,
        alegando que este cumpriu os requisitos legais, como o
         cumprimento de parte da pena e bom comportamento. "
    "- REMICAO-DE-PENA: Peticoes que solicitam a remicao de
        pena por estudo ou trabalho realizado durante o
        cumprimento da pena, destacando o esforco do apenado
        em se reintegrar a sociedade. "
    "- TRANSFERENCIA-DE-EXECUCAO: Peticoes que solicitam a
        transferencia da execucao da pena para uma comarca
        mais proxima da residencia do apenado, visando
        facilitar o acompanhamento familiar e a reintegracao
        social. "
    "- UNIFICACAO-DE-PENAS: Peticoes que tratam da unificacao
         de penas decorrentes de diferentes processos, visando
         a fixacao de um regime de cumprimento unico e
        adequado a situacao do apenado. "
    ...
    "---------------------- "
    "Retorne apenas a classe e nada mais. "
    "Peticao: {texto} "
    "Classe:"
)
```