UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Integrando Grandes Modelos de Língua na Modelagem de Tópicos para o Domínio Jurídico

Cláudio Hernandes Silva Lima

Monografia - MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

Cláudio Hernandes Silva Lima

Integrando Grandes Modelos de Língua na Modelagem de Tópicos para o Domínio Jurídico

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

Área de concentração: Inteligência Artificial

Orientadora: Prof. Dr. Ricardo Marcondes Marcacini

Versão original

São Carlos

2024

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi, ICMC/USP, com os dados fornecidos pelo(a) autor(a)

S856m

Lima, C.H.S.
Integrando Grandes Modelos de Língua na Modelagem de Tópicos para o Domínio Jurídico / Cláudio Hernandes Silva Lima ; orientador Ricardo Marcondes Marcacini. – São Carlos, 2024.
51 p. : il. (algumas color.) ; 30 cm.

Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2024.

1. Inteligência Artificial. 2. NPL. 3. Tópicos. 4. Agrupamento. 5. Processo judicial. 6. Petição inicial. 7. BERT. 8. LLM. 9. TCC. I. Marcacini, Ricardo Marcondes, orient. II. Título.

Cláudio Hernandes Silva Lima

Integrating LLMs in Topic Modeling for Law Domain

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

Concentration area: Artificial Intelligence

Advisor: Prof. Dr. Ricardo Marcondes Marcacini

Original version

São Carlos

2024

Este trabalho é dedicado aos que trabalham,
nos mais diversos segmentos,
à promoção da Jurisdição no Brasil,
tarefa árdua e fundamental para a viabilização
da vida em sociedade.

AGRADECIMENTOS

Agradeço à minha família, Neusa, Beatriz e Eduardo, pelo tempo de convívio que lhes foi subtraído durante os estudos desta Pós-Graduação.

Agradeço à Des[a] Maria de Nazaré Gouveia dos Santos, Presidente, e a Márcio Góes, Secretário de Informática, do Tribunal de Justiça do Estado do Pará, pelo amplo acesso à base de dados e recursos necessários à realização dos experimentos que fundamentam esses trabalho.

"Quem deve enfrentar monstros deve permanecer
atento para não se tornar também um monstro.
Se olhares demasiado tempo dentro de um abismo,
o abismo acabará por olhar dentro de ti."
Friedrich Nietzsche, em "Além do Bel e do Mal"

RESUMO

LIMA, C.H.S. **Integrando Grandes Modelos de Língua na Modelagem de Tópicos para o Domínio Jurídico**. 2024. 51 p. MBA em Inteligência Artificial e Big Data - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

A crescente quantidade de documentos jurídicos gerados diariamente representa um desafio significativo para advogados, juízes e profissionais da área legal, que precisam processar, analisar e compreender grandes volumes de informações de maneira eficiente. A falta de ferramentas automatizadas capazes de organizar e classificar esses documentos com precisão e rapidez pode resultar em atrasos nos processos judiciais. Nesse sentido, este trabalho propõe uma metodologia que combina técnicas de extração de tópicos e classificação de textos no domínio jurídico, com foco em petições iniciais de processos cíveis. Foi desenvolvido e avaliado um pipeline de cinco etapas, desde a amostragem de dados até a classificação de novos documentos. Inicialmente, durante uma fase de treinamento, documentos jurídicos são amostrados e submetidos a um LLM (Large Language Model) para extração de *tags* representativas. Essas *tags* permitem extrair conhecimento relevante dos documentos jurídicos com base em *prompts* definidos pelo usuário. Em seguida, essas *tags* são vetorizadas usando *embeddings* gerados por um modelo SBERT e agrupadas utilizando o algoritmo k-Means. Cada grupo é mapeado para um tópico, nomeado pela LLM, o que já permite uma análise exploratória da organização dos documentos jurídicos. Além disso, os tópicos são organizados em níveis hierárquicos, permitindo uma estrutura de navegação dos documentos por similaridade semântica. Para lidar com o processamento em grande escala, novos documentos são pré-processados e classificados com base nesses tópicos, utilizando um modelo de *fine-tuning* do BERT, mais leve que uma LLM. Resultados experimentais em uma base de documentos reais indicam que a metodologia proposta é eficaz para a organização e análise de grandes volumes de documentos jurídicos, proporcionando uma ferramenta promissora para a organização automatizada de processos judiciais.

Palavras-chave: Modelagem de Tópicos. Domínio Jurídico. Large Language Models.

ABSTRACT

LIMA, C.H.S. **Integrating LLMs in Topic Modeling for Law Domain**. 2024. 51 p. MBA in Artificial Intelligence and Big Data - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

The increasing volume of legal documents generated daily presents a significant challenge for lawyers, judges, and legal professionals who need to process, analyze, and comprehend large amounts of information efficiently. The lack of automated tools capable of organizing and classifying these documents with accuracy and speed can lead to delays in judicial processes. In this context, this paper proposes a methodology that combines topic extraction and text classification techniques in the legal domain, focusing on initial petitions for civil lawsuits. A five-stage pipeline was developed and evaluated, ranging from data sampling to the classification of new documents. Initially, during a training phase, legal documents are sampled and processed using a Large Language Model (LLM) to extract representative tags. These tags enable the extraction of relevant knowledge from legal documents based on user-defined prompts. Subsequently, these tags are vectorized using embeddings generated by an SBERT model and grouped using the k-Means algorithm. Each group is mapped to a topic, named by the LLM, which allows for exploratory analysis of the organization of legal documents. Additionally, the topics are organized into hierarchical levels, enabling a semantic similarity-based navigation structure for the documents. To handle large-scale processing, new documents are pre-processed and classified based on these topics, using a fine-tuned BERT model, which is lighter than an LLM. Experimental results on a real document base indicate that the proposed methodology is effective for organizing and analyzing large volumes of legal documents, providing a promising tool for the automated organization of judicial processes.

Keywords: Topic Modeling. Law Domain. Large Language Models.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização

A modelagem de tópicos em documentos jurídicos é uma tarefa que tem ganhado cada vez mais atenção, pois simplifica o acesso e compreensão de vastas coleções de informações legais. Esta abordagem permite organizar os documentos em temas, facilitando a recuperação. Ao aplicar modelos de tópicos a conjuntos de documentos, busca-se identificar conceitos semânticos similares e uma sumarização de alto nível, de forma a atender a hipótese de que, se o usuário ou analista está interessado em um documento de um tópico, provavelmente está interessado em outros documentos similares do mesmo tópico. Isso ocorre, por exemplo, na atividade jurisdicional, no agrupamento de processos similares a fim racionalizar a atividade do julgador ou ainda a fim de identificar a ocorrência de demandas de massa ou demandas predatórias (Silveira *et al.*, 2021).

Recentemente, métodos para modelagem de tópicos têm se beneficiado do emprego de representações textuais baseadas em *embeddings*. Os métodos atuais mais populares de modelagem de tópicos, como o *BERTopic*, englobam três principais etapas: pré-processamento, no qual o texto é transformado em representações vetoriais de *embeddings*; *clustering*, que organiza documentos em grupos segundo a similaridade nos espaços de *embeddings*; e *cluster labeling*, que atribui conjuntos de descritores a cada grupo para definir o tópico correspondente (Grootendorst, 2022).

Apesar das melhorias alcançadas recentemente, a modelagem de tópicos no contexto jurídico enfrenta desafios particulares. Documentos legais são caracterizados por um vocabulário e uma estrutura específica para este domínio. Diferentemente de textos convencionais, os documentos jurídicos são compostos por seções específicas, tais como identificação das partes, revisão dos fatos, fundamentação jurídica, valor da causa, pedido, relatório e dispositivo (Li *et al.*, 2023). A importância relativa de cada segmento varia, impactando diretamente no conceito de similaridade entre tais documentos. Outro desafio está na própria definição de um tópico. A tradicional determinação de descritores de tópicos baseados em palavras-chave revela-se restrita para esse domínio. Em outras palavras, a simples identificação por meio de palavras-chave revela-se insuficiente, tornando importante a incorporação de descrições mais detalhadas, como sentenças ou parágrafos que sumarizam de maneira mais semântica para o analista o conteúdo dos documentos alocados aos tópicos em questão.

Embora existam iniciativas notáveis voltadas para a língua portuguesa, como é o caso do LegalBERT, (Chalkidis *et al.*, 2020), é relevante observar que essas abordagens se concentram nos desafios associados ao vocabulário específico do domínio jurídico.

Contudo, há uma lacuna significativa, uma vez que essas iniciativas não atendem a análise segmentada dos documentos legais ou à geração de tópicos mais semânticos. A necessidade de compreender a importância diferenciada de cada parte de um documento jurídico, bem como a demanda por tópicos que incorporem maior interpretação semântica, permanecem como áreas de pesquisa em aberto no âmbito de modelagem de tópicos para o domínio jurídico.

1.2 Justificativa e Motivação

O surgimento e popularização dos Grandes Modelos de Língua (LLMs, do inglês *Large Language Models - Grandes Modelos de Língua*) marca uma evolução importante em tarefas de processamento de linguagem natural, superando os modelos anteriores, como o BERT (*Bidirecional Encoder Representations from Transformers*). A principal vantagem desses modelos reside na sua capacidade de compreensão contextual e na representação rica e multidimensional das relações semânticas dentro do texto. Enquanto modelos anteriores, como o BERT, focavam principalmente em prever palavras seguintes em uma sequência, os LLMs conseguem capturar nuances semânticas mais complexas, principalmente por permitir condicionar a geração de textos por meio de *prompts*. Essa melhoria substancial na representação semântica torna os LLMs uma alternativa promissora para melhorar tarefas como modelagem de tópicos, em que a extração de informação dos textos e o conceito de proximidade entre documentos são cruciais.

Uma consideração relevante surge com o fato de que os primeiros Grandes Modelos de Língua (LLMs), como o GPT (3.5 e superiores) e o BARD, são proprietários. Lidar com dados sensíveis do domínio jurídico utilizando modelos proprietários, muitas vezes hospedados em infraestruturas externas à organização, torna-se um procedimento inviável em virtude das restrições legais impostas pela LGPD (Lei Geral de Proteção de Dados - Lei 13.709/2018), além das limitações financeiras que podem surgir. Contudo, atualmente estão em andamento iniciativas direcionadas à democratização de LLMs, possibilitando o uso de hardware mais acessível e a execução local desses modelos. Exemplos incluem modelos baseados em Llama, OPT e Mixtral. Entretanto, dado que esses modelos foram recentemente disponibilizados, ainda há uma lacuna em termos de avaliação de seu desempenho na tarefa específica de modelagem de tópicos, especialmente quando se trata do domínio jurídico.

Dessa forma, os aspectos mais relevantes desta pesquisa envolvem a exploração de LLMs em duas etapas fundamentais da tarefa de modelagem de tópicos: pré-processamento e pós-processamento. No que diz respeito ao primeiro desafio, é importante investigar como condicionar o modelo através de *prompts*, garantindo a extração das informações mais pertinentes dos documentos jurídicos, em um formato previsível - *LLM format enforcing* -, para a entrada no modelo de tópicos. Nesse contexto, destaca-se a relevância

de determinar quais segmentos ou informações do documento devem ser utilizados para obter uma representação de *embedding*, calcular a proximidade e realizar os agrupamentos correspondentes.

No segundo desafio, a questão central é a seleção de documentos representativos de um tópico, ou mesmo segmentos específicos desses documentos, para a subsequente geração de um sumário que seja relevante na representação daquele tópico. Embora a avaliação da qualidade dos tópicos seja uma tarefa subjetiva, a natureza específica do domínio jurídico permite uma análise mais aprofundada e deve contar com apoio de especialista da área jurídica.

1.3 Questões de Pesquisa e Objetivos

O objetivo geral deste projeto é avaliar o uso de Grandes Modelos de Língua (LLMs) nos processos de modelagem de tópicos para documentos jurídicos, considerando as nuances específicas do domínio legal, desde a extração de informações relevantes até a geração de resumos representativos para cada tópico. Para atender o objetivo geral, são propostos três objetivos específicos:

1. Investigar estratégias para condicionar LLMs na etapa de pré-processamento por meio de *prompts*, buscando extrair informações relevantes dos documentos jurídicos para a entrada no modelo de tópicos.

2. Desenvolver métodos para a seleção eficiente de documentos representativos de tópicos, considerando a complexidade e especificidade do domínio jurídico, e explorar técnicas para resumir e definir rótulos ou descrições sucintas para cada tópico.

3. Avaliar a qualidade dos tópicos gerados, levando em consideração a subjetividade inerente à avaliação, e verificar a representatividade desses tópicos no contexto jurídico.

1.4 Metodologia

A metodologia adotada para este projeto compreende várias etapas, desde a coleta e pré-processamento dos dados até a avaliação dos resultados obtidos. Abaixo é apresentado uma breve descrição das principais fases da pesquisa:

- Coleta de Dados:

A coleta de dados será realizada em fontes que disponibilizem documentos jurídicos representativos do contexto brasileiro. Esses documentos podem incluir jurisprudência, leis, contratos e decisões judiciais. Especificamente, serão utilizadas as petições iniciais de processos da área cível, que são os documentos que dão início ao processo judicial,

identificando-se as partes litigantes, os fatos que deram origem à demanda, os fundamentos jurídicos do pedido e o pedido ao órgão julgador.

- Pré-processamento de Dados

Os documentos coletados serão submetidos a um processo de pré-processamento para tornar os textos mais adequados à modelagem de tópicos. Nessa etapa, será realizada a extração de informações relevantes por meio de LLMs, bem como a avaliação de *prompts* para apoiar essa tarefa. Serão elaborados *prompts* específicos para condicionar esses modelos, visando a extração das partes mais críticas e informativas dos documentos.

- Modelagem de Tópicos

A fase de modelagem de tópicos é constituída de pré-processamento e *clustering*. Os documentos serão representados por *embeddings*, obtidos a partir das respostas das LLMs, facilitando a identificação de tópicos semânticos. A análise de similaridade será realizada submetendo os *embeddings* a um algoritmo de agrupamento vetorial.

- Melhorando a Descrição dos Tópicos:

Serão desenvolvidos métodos para a seleção eficiente de documentos representativos de tópicos, levando em consideração a importância diferenciada de cada segmento no contexto jurídico. Aqui, novamente serão avaliados *prompts* de LLMs abertas para gerar descritores e/ou sumários mais semânticos. A avaliação da qualidade dos tópicos será feita por especialista da área jurídica.

- Avaliação e Análise de Resultados:

Os resultados obtidos serão avaliados em termos de qualidade dos tópicos gerados, considerando a representatividade no contexto jurídico. Para tal, será considerado uma prova de conceito na área cível, mais especificamente em processos de Vara de Juizado da Fazenda Pública.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Uso de Modelos de Linguagem no Domínio Jurídico

Desde o seu lançamento, em novembro de 2022, o ChatGPT tem chamado a atenção da comunidade científica e as pessoas em geral pelas suas surpreendentes capacidades para solucionar variados problemas, o que resulta em setores da economia sendo submetidos a uma revolução sem precedentes. Prevê-se que a adoção de LLMs, de forma geral, e do ChatGPT, em particular, terá impacto no mercado de trabalho com repercussão tanto em postos quanto em remuneração de diversos setores (Felten; Raj; Seamans, 2023).

As Ciência Jurídicas e suas áreas de aplicação não estão imunes a esta tecnologia, sendo objeto de estudo, aplicação e preocupação, havendo iniciativas do uso de IA e, mais especificamente, LLMs desde o aconselhamento jurídico até a geração de minutas de decisão judicial ou predição de sentenças.

As LLMs, como parte mais específica da área de inteligência artificial, têm demonstrado capacidade de entendimento, geração e predição de texto, em estilo e qualidade próximos do produzido por um ser humano, a partir de uma entrada do usuário, os chamados *prompts*. Neste cenário, a aplicação das LLMs na área jurídica vai desde a elaboração de peças jurídicas, predição de desfecho de demandas judiciais até a ajuda na pesquisa acadêmica da área jurídica.

Quanto ao ensino e a prática jurídicas, Ajevski *et al.* (2023) avalia a implicação do uso do ChatGPT, sugerindo que as faculdades de Direito devem estabelecer estratégias para se adaptar à tecnologia, tanto quanto aos métodos de avaliação quanto incorporando a IA no currículo. A capacidade da ferramente em gerar textos sofisticados pode ter um papel importante na prática jurídica, tornando certas tarefas mais eficientes, sempre se tendo o cuidado, entretanto, de lidar com sua inabilidade de checar a acurácia de seus resultados, sendo, portanto, primordial a supervisão humana final.

Grossman *et al.* (2023) faz uma avaliação dos impactos abrangentes da tecnologia generativa na atividade judicante, preocupando-se com os impactos nos meios de provas a serem produzidos por documentos gerados por essa tecnologia, ante a evidente dificuldade em se distinguir provas legítimas daquelas produzidas por IA. Também aborda a possibilidade da existência de demandas geradas de forma massiva por ferramentas generativas. Por outro lado, também vislumbra-se a possibilidade de a ferramente ser usada de forma a facilitar o acesso ao judiciário a pessoas que não possuem advogado.

A partir de experimentos que demonstram que o ChatGPT é capaz de ser aprovado no UBE (*Uniform Bar Examination*), equivalente ao Exame Nacional da Ordem dos Advogados no Brasil, (Katz *et al.*, 2024) enfatiza que o sistema jurídico é uma área

desafiadora para o PLN, dada a complexidade e nuance do linguajar jurídico, o que indica o significativo avanço do ChatGPT, ao entender e gerar textos jurídicos complexos. Ao final, conclui que a tecnologia não pode ser entendida como substitutiva do ser humano, mas um complemento que pode melhorar os serviços jurídicos, tornando-os mais acessíveis e eficientes, sem se perder de vista seu uso seguro e ético. Em sentido semelhante (Perlman, 2022) e (Sun, 2023).

A engenharia de *prompts* ((Amazon, 2024)) tem sido uma área objeto de intensas pesquisas. No domínio jurídico, Ioannidis *et al.* (2023) utiliza engenharia de *prompt*, juntamente com *embeddings* de textos, para desenvolver ferramentas para a geração de *newsfeeds* de informações regulatórias, registro de obrigações decorrentes da legislação e ferramentas de consulta legal, com o objetivo de manter empresas e escritórios de advocacia atualizados sobre mudanças regulatórias dentro de sua área de atuação.

Yu, Quartey e Schilder (2022) utiliza a técnica *Chain-of-Thought - COT* juntamente com métodos de ajuste fino (*fine tunning*) da LLM, com ou sem explicações, para construir raciocínios (*reasoning*) em atividades jurídicas, com resultados promissores. Também utilizam *prompts* baseados em técnicas de raciocínio jurídico, tal como IRAC (*Issue, Rule, Application, Conclusion*), os quais apresentaram melhores resultados.

Kuppa, Rasumov-Rahe e Voses (2023) propõe uma estratégia para a construção de *prompts* destinados à melhoria do processamento de questões legais pela LLM. A técnica sugerida foi denominada de CoR - *Chain of Reference* - e consiste em se encadear dois *promts*. O primeiro, chamado de estágio 0, requer que a LLM quebre a entrada em partes pré-definidas (*issue, thesis/claim, rule/law, evaluation/evidente/principle, application/apply, reasoning/evaluation e conlsuion/outcome/policy*). Com o resultado do primeiro estágio e a entrada do usuário, submete-se à LLM a tarefa efetivamente desejada. Segundo os autores, essa técnica aumenta em 12% o desempenho da LLM em relação a uma estratégia *zero-shot*.

Voltado para explorar o uso do ChatGPT no auxílio a autores de processos judiciais sem o acompanhamento de advogado, Steenhuis, Colarusso e Willey (2023) elabora algumas abordagens para o preenchimento de formulários judiciais, concluindo que a solução mais adequada é a que usa a IA para gerar um rascunho do formulário a ser submetido a uma revisão humana. Algo semelhante foi explorado por Yuan *et al.* (2023)

Outra estratégia de uso de modelo de linguagem no domínio jurídico é o treinamento com *corpus* jurídicos, ajustando, *fine-tunnig*, os modelos para a área jurídica. Por exemplo, (Chalkidis *et al.*, 2023) disponibiliza: um *corpus* de textos jurídicos da Comunidade Europeia, Canadá, Corte Europeia de Direitos Humanos, Estados Unidos, Inglaterra e Índia, chamado de LeXFiles; duas LLMs treinadas com esse corpus, a partir de RoBERTAa; um *benchmark* específico para o domínio jurídico chamado de LegalLAMA, além da avaliação de 7 LLMs, tanto com LeXFiles quanto com LegalLAMA. Iniciativas semelhantes

foram feitas, a partir do BERT, para o Italiano (Licari; Comandè, 2022), Francês canadense (Garneau *et al.*, 2021) e Português (Silveira *et al.*, 2023).

Preocupado com o compartilhamento de informações sigilosas, Yue *et al.* (2023a) propõe a utilização do esquema de treinamento denominado de aprendizado federado (*federated learning*) em que o ajuste fino do modelo é feito localmente, sem a disponibilização de dados sensíveis, compartilhando apenas os parâmetros do modelos em um servidor centralizado.

Considerando que textos jurídicos possuem uma estrutura bem definida, com dependências linguísticas distantes entre suas diversas seções (partes, fato, fundamentação, dispositivo e identificação), Li *et al.* (2023) propõe um modelo de linguagem pré-treinado para o domínio jurídico voltado para a recuperação de casos judicais, que leva em conta os segmentos que compõem uma sentença, em especial os fatos, a fundamentação e a parte dispositiva.

Outra abordagem é a de Savelka *et al.* (2023b). Com o GPT-4 e uma técnica de aumentação (*augmentation*), é avaliada a capacidade de o modelo gerar explicações para termos legais, sob os aspectos de aderência aos fatos, clareza, relevância, riqueza de informação e foco. Em que pese a qualidade dos textos gerados ser muito boa, conclui-se que há problemas com a acurácia fática das respostas. O uso da técnica de *augmentation*, que fornece ao modelo texto legal com relevância contextual, melhora significativamente as métricas dos resultados. Os mesmos autores, em Savelka *et al.* (2023a), concluem que as anotações de textos legais geradas pelo GPT-4 são comparáveis em qualidade às de estudantes de Direito, o que abre a possibilidade do uso da tecnologia para gerar anotações em domínios especializados, embora reconheçam que o modelo é sensível a pequenas mudanças no *prompt*.

Buscando predizer o julgamento de casos criminais, Wu *et al.* (2023) utiliza LLMs e modelos de domínios (CNN, BERT, Roberta e outros), em conjunto com um banco de dados de precedentes, para propor um *framework* para prever o julgamento de casos criminais, com indicação do artigo legal, crime e pena a ser aplicada. Já Trautmann, Petrova e Schilder (2022) propõe o uso de engenharia de *prompts* para predizer o julgamento contido em documentos longos com o uso de LLMs. Caso os documentos excedam o limite de *tokens* do modelo, o texto é truncado.

Yue *et al.* (2023b) ressalta a possibilidade de amplo uso de IA no domínio legal, desde o público em geral - fornecendo consultas legais e solução de disputas -, passando pelos estudantes de Direito - como tutores, ajudando a consolidar conhecimento legal e provendo solução para questões legais - até os profissionais da área - fornecendo ferramentas avançadas para consultas legais, análise de casos e sumarização de documentos. Diante deste cenário e para atender a estas tarefas, propõe uma LLM (DISC-LawLLM) treinada a partir de *prompts* construídos com base em silogismos legais (premissa maior - lei, premissa

menor - fato). Com esta estratégia, foi montado um *dataset* (DISC-LawLLM), utilizado para treinamento de uma LLM de 13 bilhões de parâmetros de propósito geral, que também dispõe de um módulo de recuperação de informações jurídicas atualizadas (*augmentation*), a fim gerar respostas mais confiáveis. A fim de avaliar o desempenho de LLMs no contexto legal, é proposto um *benchmark* denominado DISC-Law-Eval, que contém uma avaliação subjetiva e objetiva dos resultados obtidos.

Dai *et al.* (2023) propõe um *benchmark* específico para o domínio jurídico, *LAiW*, a fim de avaliar as LLMs, também baseado na noção de silogismo jurídico. As capacidades ligadas ao domínio legal são divididas em três níveis: recuperação de informações básicas (BIR), inferência de fundamentos legais (LFI) e aplicação legal complexa (CLA). Cada nível possui várias tarefas, em um total de 14 tarefas. Concluem que, apesar de as LLMs, quer gerais quer treinadas no domínio jurídico, conseguirem desempenhar atividades jurídicas complexas, falham em tarefas básicas, o que representa obstáculo para seu uso em atividades práticas. Propõe, ao final, o desenvolvimento de tarefas de treinamento das LLMs a fim de ela melhor lidar com a lógica jurídica. Foram avaliadas 18 LLMs: 7 especializadas no domínio jurídico; as 6 LLMs que serviram de base (*baseline*) para o treinamento das especializadas e 5 de propósito geral, entre elas GPT-4 e ChatGPT. Na média, GPT-4 e ChatGPT mostraram melhor desempenho em todas as tarefas.

Fei *et al.* (2023), da mesma forma que Dai *et al.* (2023), propõe um *benchmark* para avaliação das LLMs no domínio jurídico, denominado *LawBench*. Os níveis de capacidade propostos são: memorização, entendimento e aplicação de conhecimento jurídico. Tais níveis são avaliados em 20 tarefas divididas em cinco grupos: classificação de uma categoria, classificação de múltiplas categorias, regressão, extração e geração. Com o *benchmark* proposto, foram testadas 51 LLMs, tanto específicas do domínio legal quanto de propósito geral, concluindo que o GPT-4 teve melhor desempenho, conclusão semelhante ao trabalho de Dai *et al.* (2023).

De forma colaborativa, Guha *et al.* (2024) desenvolveram o *benchmark* LegalBench, que avalia 162 tarefas, agrupadas em 6 tipos de raciocínio jurídicos (*legal reasoning*): identificação de problema (*issue-spotting*), identificação da norma (*rule-recall*), conclusão da norma (*rule-conclusion*), aplicação da norma (*rule-application*), e interpretação (*interpretation*). Com o índice, foram avaliadas 20 LLMs, de 11 famílias diferentes, entre comerciais e *opensource*. Os resultados são categorizados por tipo (comercial ou *open-source*) e tamanho das LLMs (grande, 13, 7 e 3 bilhões de parâmetros). O melhor desempenho geral em todas as categorias e tipos de raciocínio foi do GPT-4.

Com o uso o algoritmo T5, baseado em *transformers*, Burman e Bradford () desenvolveram um algoritmo para a sumarização de documentos jurídicos, bem como uma interface de usuário em que se pode inserir o documento a ser sumarizado. Vários algoritmos de sumarização foram testados, como *TextRank*, *Latent Semantic Analysis*, *T5*

transformers, *BART transformers*, *GPT-2 Transformers*, *XLM Transformers*, *PEGASUS* e *BERT*, os quais foram avaliados com a métrica ROUGE. Embora *BERT* tenha apresentado resultados ligeiramente melhores que o *T5* e *Pegasus*, deu-se preferência ao *T5* por ser mais rápido e fácil de treinar.

Oliveira e Nascimento (2022) utilizou seis modelos baseados em *transformers*, *BERT*, *GPT-2* e *RoBERTa*, para agrupar processos judiciais. Foram usadas duas versões de cada modelo, uma treinada com pt-BR de uso geral e outra treinada com 210.000 documentos jurídicos do Tribunal Regional do Trabalho da 5ª Região. Os modelos foram usados gerar *embeddings* das petições de recursos ordinários na Justiça do Trabalho que foram agrupados usando o cosseno como medida de similaridade e o algoritmo clusterização *k-means*. Os pesquisadores concluíram que o uso de técnicas baseadas em *transformers* são promissoras paras identificação de padrões em documentos jurídicos, tendo o modelo *RoBERTa* pt-BR apresentado os melhores resultados.

2.2 Modelagem de Tópicos com Modelos de Língua

Silveira *et al.* (2021) aplicou o método descrito em Grootendorst (2022) para extrair os tópicos de documentos legais. A modelagem de tópicos é útil para o tratamento de grandes quantidades de documentos, em que um modelo de tópicos é usado para extrair conceitos semânticos interpretáveis, ou tópicos, dos diversos documentos analisados. Os tópicos podem ser usados para a elaboração de resumos de alto nível da coleção, pesquisa de documentos de interesse ou agrupamento de documentos. Para a representação dos textos (*embeddings*) foi usado o LEGAL-BERT, modelo pré-treinado no contexto legal, acrescendo-se ao texto de decisões de um *dataset* da Suprema Corte Americana os artigos das leis por eles referidos. Cada documento foi dividido em parágrafos, cujos *embeddings* foram agrupados, retirando-se as palavras mais representativas de cada agrupamento de parágrafos para representar o grupo na seleção do tópicos mais importantes de cada documento. Ao final, os tópicos gerados pelo métodos foram avaliados por especialistas humanos, que indicaram se os tópicos selecionados representam o tema principal do documento, utilizando-se para tanto, como métrica, o coeficiente *Kappa*, chegando-se a um nível de concordância de 0,78.

Aguiar *et al.* (2022) utilizou a modelagem de tópicos e informações sobre a legislação citada nos documentos para identificar o assunto neles tratado a fim de classificar petições iniciais de processos criminais. Foram usados 16 mil petições iniciais, inclusive denúncias criminais, do Tribunal de Justiça do Estado do Ceará, a quais foram classificadas de acordo com as cinco classes de processos mais representativas da trabela padronizada do CNJ (Conselho Nacional de Justiça): procedimento cível comum, execução de título extrajudicial, ação criminal - procedimento ordinário, procedimento judicial especial cível e execução fiscal. Concluem que o uso de *embeddings* gerados pelo BERT juntamente com

a legislação mencionada na representação do documento pode melhorar a acurácia da classificação.

2.3 LLMs como Modelos de Tópicos Implícitos

Fora especificamente do domínio legal, Wang *et al.* (2023) avalia que as LLMs podem ser vistas como modelos de tópicos implícitos, objetivando melhorar o entendimento e implementação de aprendizado contextual, que consiste no aprendizado a partir de alguns exemplos ou demonstrações, sem atualização dos parâmetros do modelo, em que pese a sensibilidade dos modelos às escolhas dos formatos e ordem dos exemplos.

Os autores estudam as LLMs sob a ótica Bayesiana e postulam ser elas modelos de tópico implícitos que inferem uma variável latente a partir dos *prompts*. Baseado nisso, propõem um algoritmo em dois passos que, primeiramente, extrai *tokens* conceituais latentes de uma pequena LLM e seleciona as demonstrações que têm a maior probabilidade de predizer os correspondentes *tokens* conceituais. As demonstrações selecionadas podem, então, ser generalizadas por outras LLMs.

É neste sentido que a proposta deste trabalho atua, na qual LLMs terão uma tarefa primária na identificação dos tópicos relevantes dos textos jurídicos.

3 METODOLOGIA

No âmbito jurídico, a extração de tópicos e classificação de textos é útil em vários cenários. De modo geral, trabalha-se com textos longos, em especial se considerando os tamanho dos contextos das LLMs atualmente disponíveis, que demandam tempo de leitura a fim de se apreender seu conteúdo.

A extração de tópicos que captem as principais ideias semânticas de um documento jurídico é uma tarefa útil a fim de se apreender de forma célere e eficaz seu conteúdo. Em um segundo momento, considerando uma coleção de vários documentos, os tópicos de cada um deles podem ser usados para agrupá-los por semelhança, permitindo que se dê um tratamento semelhante a documentos com conteúdo semântico próximos.

Um processo judicial pode ser entendido como conjunto de documentos, produzidos sequencialmente no tempo e que podem ser de vários tipos: petições das partes envolvidas no processo, certidões, despachos, decisões, sentenças, acórdãos, relatórios, laudos periciais, dentre outros. Os documentos produzidos pelas partes, a depender da fase processual, pode ter nomes específicos, tais como petição inicial, denúncia, contestação, resposta à acusação, memoriais, alegações finais, razões recursais, contrarrazões recursais e pareceres.

Em se considerando uma solução de automatização de tarefas do domínio jurídico, pode-se utilizar um ou outro tipo de documento para agrupamento. Assim, se se deseja agrupar recursos destinados a um tribunal, as razões recursais das partes podem ser as peças a serem tratadas.

No presente trabalho, foram utilizadas petições iniciais de processos cíveis em trâmites em uma vara do Juizado da Fazenda Pública. As petições iniciais definem em grande parte o escopo de um processo judicial, de modo que a extração de seus tópicos pode ser de grande valia para o rápido entendimento da demanda, bem como para outras tarefas como agrupamento de processos por semelhança, identificação de demandas em massa ou demandas predatórias.

A solução proposta é baseada em um *pipeline* composto de cinco etapas, que vão da extração da amostra dos dados até a classificação de novos documentos. Na 1ª etapa, é feita a amostragem dos documentos a serem utilizados no experimento, de onde serão retirados o conjunto de treinamento e o conjunto de teste.

Na 2ª etapa, os documentos de treinamento são submetidos à LLM para extração das *tags*. As *tags* de saída da 2ª etapa são, na 3 etapa, agrupadas em dois níveis, gerando os tópicos. Os tópicos gerados na 3ª etapa serão usados para o treinamento de um modelo indutivo (classificador de textos), na 5ª etapa.

Os documentos a serem classificados são pré-processados na 4ª etapa, a fim de serem classificados na 5ª etapa, com o uso do modelo indutivo, cujo treinamento foi feito com os tópicos gerados na 3ª etapa. Por fim, na etapa final, a 6ª, é atribuído um tópico para o documento. A Figura 1 contém um fluxograma do *pipeline* utilizado. A seguir, passa-se a explicar cada uma das etapas de forma mais detalhada.

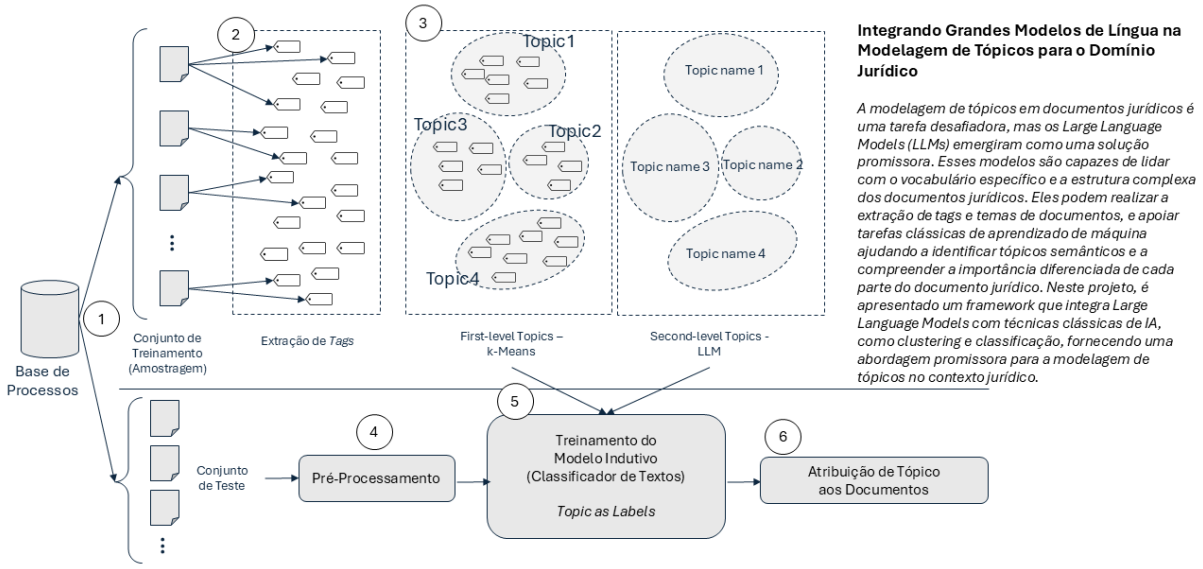

Figura 1 – Pipeline da metodologia proposta

3.1 1ª Etapa - Amostragem dos Dados

Foram utilizadas 500 petições iniciais, obtidas de forma aleatória, de uma única unidade jurisdicional de demanda de alta mensal, em torno de 550 processos novos por mês, e com 6000 processos em andamento, entre os anos de 2015 e 2022. Por se tratar de um juizado da Fazenda Pública e dada a natureza das demandas processadas, há assuntos recorrentes, de modo que se espera que existam *clusters* bem definidos de processos com tópicos semelhantes.

Antes de se fazer a seleção aleatória das petições, excluiu-se tão somente petições muito pequenas ou muito grandes, vez que, dada a forma com que os documentos são extraídos e ocerizados do banco de dados original, tais petições podem conter arquivos corrompidos.

3.2 2ª Etapa - Extração dos *Tags*

A extração de *tags* é o primeiro passo para a obtenção dos tópicos que serão utilizados para treinamento do modelo indutivo. Os *tags* são obtidos a partir das respostas ao *prompt* de uma LLM. O *prompt* utilizado apresenta à LLM o texto do documento seguido do comando para que sejam extraídos os tópicos representativos do texto, devendo a LLM retornar os tópicos enumerados. A Figura 2 ilustra o processo de extração de *tags* de um documento.

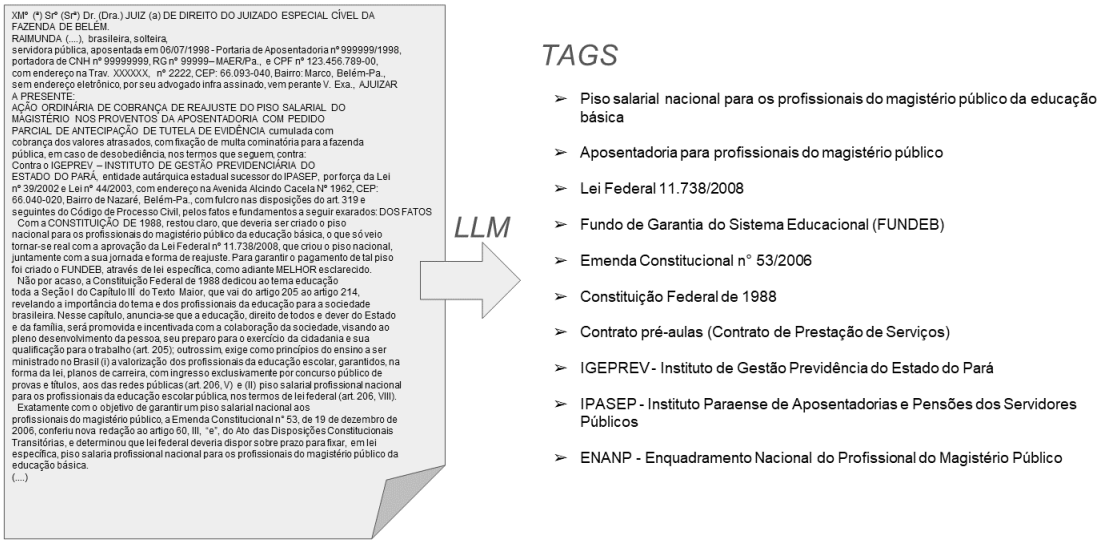

Figura 2 – Extração de *Tags* com a LLM

Os documentos utilizados no experimento - petições iniciais de processos judiciais - possuem comumente tamanhos maiores que o tamanho do contexto das LLMs disponíveis no início dos experimentos deste trabalho. Por esta razão, utilizou-se uma solução de contorno para o caso de o tamanho do documento exceder o tamanho do contexto. Nestes casos, uma janela deslizante era utilizada para apresentar à LLM trechos sucessivos do documentos limitados ao tamanho do contexto. A Figura 3 contém uma ilustração dos *tags* gerados para um processo de tamanho maior que o contexto da LLM. Cada linha possuía o número do processo e os *tags* obtidos em cada iteração com a LLM, separados por vírgula, de modo que o mesmo processo podia ter mais de uma entrada no *dataset* e, em cada entrada, mais de um *tag*.

Ocorre que, dada a rápida evolução das LLMs, ainda durante a realização dos experimentos, tivemos acesso ao LLama 3.0 da Meta, cujo tamanho de contexto dispensou o uso das janelas deslizantes no experimento final.

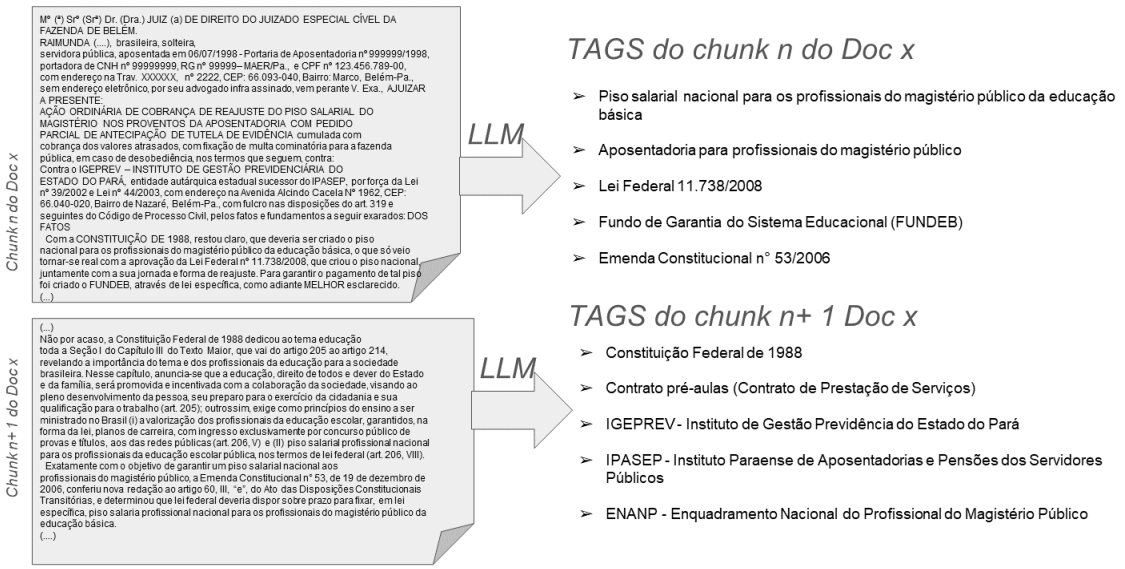

Figura 3 – Documentos maiores que o contexto da LLM são submetidos em janelas deslizantes - *chunks*

3.3 3ª Etapa - Geração dos Tópicos

A partir dos *tags* dos documentos, os tópicos foram gerados seguindo as seguintes sub-etapas. Inicialmente, os *tags* de cada processo foram individualizados, de modo que cada entrada do *dataset* possuía o número de um processo e um único *tag*. Em seguida, os *tags* foram vetorizados através de *embeddings*, de modo que, neste momento, puderam ser dispostos em um espaço vetorial, conforme se ilustra na Figura 4.

Os *embeddings* foram gerados com o *SBERT*, (Reimers; Gurevych, 2019), um modelo de língua derivado do *BERT*, (Devlin *et al.*, 2019). O BERT (*Bidirectional Encoder Representations from Transformers*) é um modelo de linguagem baseado em *transformers*, apresentado em outubro de 2018 por pesquisadores do Google. Apesar de a arquitetura dos *transformers* possuir inerentemente módulos de codificadores e decodificadores, o BERT usa somente os codificadores. Simplificadamente, o BERT possui três módulos: o de *embedding*, a pilha de codificadores e o de *un-embedding*. No módulo de *embedding*, o texto de entrada, representado na forma de codificação *one-hot*, é convertido para a forma vetorial que, a seguir, sofre transformações na pilha de transformadores, sendo a representação final convertida nos *tokens* de saída codificados na forma *one-hot*, na camada final. A camada final é muito útil na fase de pré-treinamento, porém, muitas vezes, a saída da pilha de transformadores é utilizada como uma representação vetorial da entrada e empregada para treinar um modelo menor, montado logo após os transformadores, especializado em outras tarefas, como classificação de textos. No *SBERT*, o modelo pré-treinado original do *BERT* foi modificado especificamente com o propósito de gerar *embeddings* de sentenças a fim

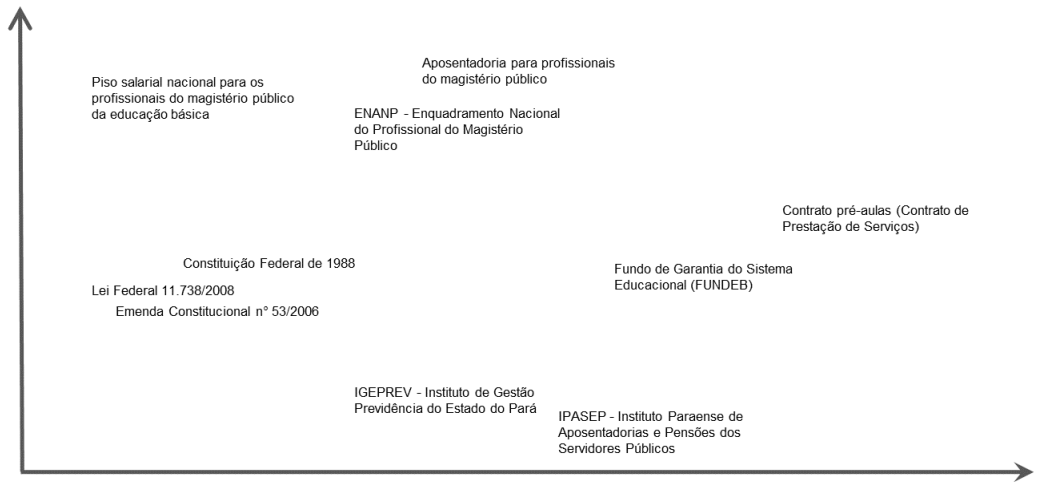

Figura 4 – *Embeddings* dos TAGS dispostos um espaço vetorial, após vetorização

de serem comparadas com a medida de cosseno. Em uma coleção de 10.000 sentenças, o *SBERT* reduz o tempo computacional de encontrar um par de frases mais semelhantes de 65 horas para 5 segundos. Segundo a documentação ((SentenceTransformers. . . ,)), o SBERT é um módulo para "acessar, usar e treinar modelos de *embedding* texto e imagem estado da arte".

Com os *embeddings* obtidos com o *SBERT*, passa-se à sua clusterização utilizando-se o algorítimo *k-Means*. Este algorítimo não supervisionado, a partir de um número pré-definido de *clusters* de um conjunto de dados vetorizados, aloca cada um dos vetores do conjunto ao *cluster* de pontos mais próximos, usando a medida de distância euclidiana entre o documento e o representante (centroide). No primeiro passo do algoritmo, são escolhidos aleatoriamente K centroides dentro do conjunto de dados. Em seguida, itera-se pelos dois passos seguintes: no primeiro, associa-se cada amostra ao centroide mais próximo; no segundo, novos centroides são criados a uma distância média de cada uma das amostras anteriores, associadas ao mesmo centroide. Computa-se, então, a distância entre os dois centroides e os dois últimos passos são repetidos até que os centroides não se movam significativamente (Clustering,). Ao final desta etapa, os *tags* mais próximos estarão etiquetados com o respectivo número de *cluster*, atribuído pelo algorítimo, conforme se ilustra na Figura 5

Os *tags* que compõem os mesmo *clusters* são então submetidos à LMM para obtenção dos tópicos de cada *cluster* de *tags*. O tópico de um *cluster* é um texto que possua pertinência semântica com todos os *tags* do *cluster*, obtido de forma automática com a LLM. Esses tópicos são chamados de tópicos de primeiro nível, ou *first level topics*,

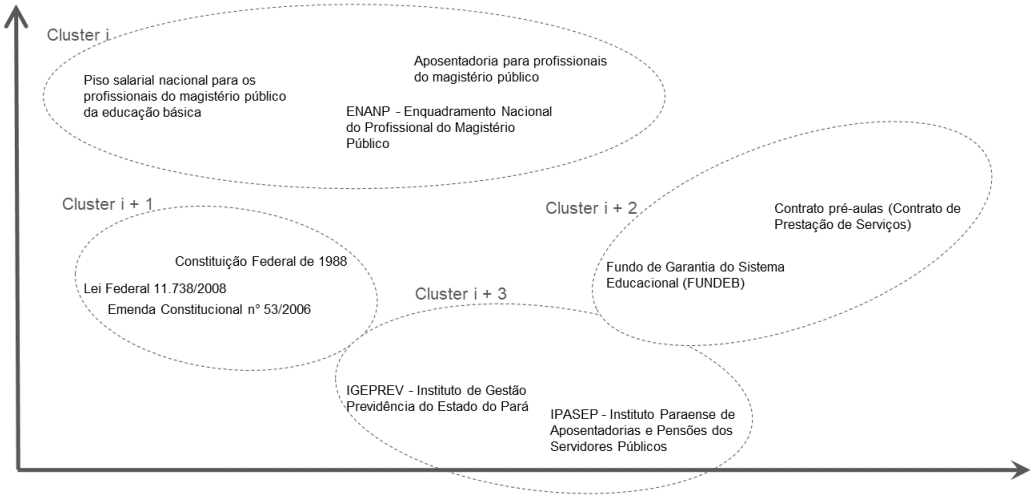

Figura 5 – *Tags* agrupados pelo *kmeans*

que estão ilustrados na Figura 6

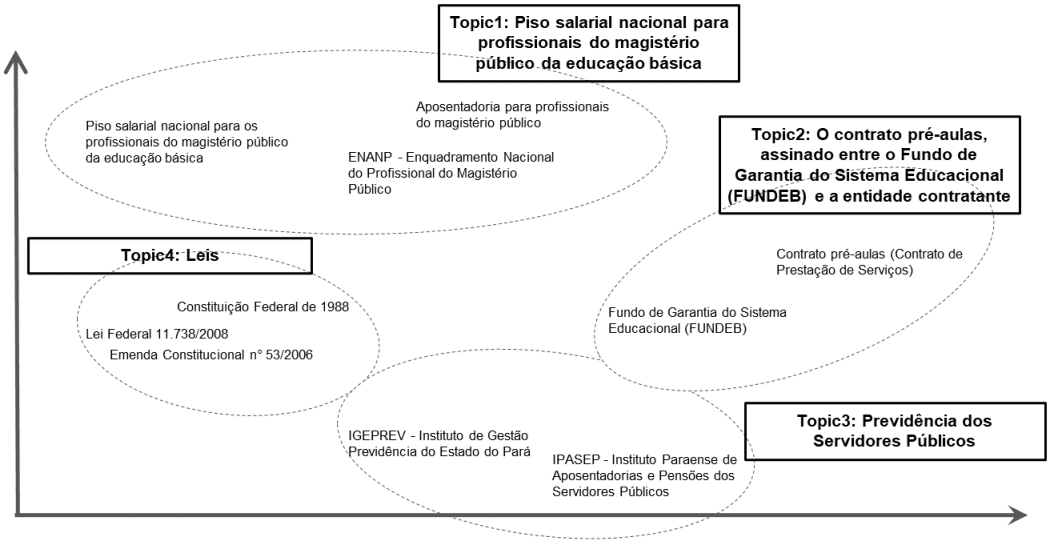

Figura 6 – Tópicos de primeiro nível

De forma recursiva, os tópicos de primeiro nível são vetorizados e mais uma vez clusterizados com o *kmeans*, gerando assim os tópicos de segundo nível, ou *second-level topics*. O objetivo de se obter tópicos de segundo nível, ou mesmo níveis mais altos, é, em grandes coleções de documentos, explorar o acervo em uma estrutura assemelhada a uma árvore de tópicos, do mais genérico até o mais específico, chegando-se até cada documento agrupado, conforme Figura 7.

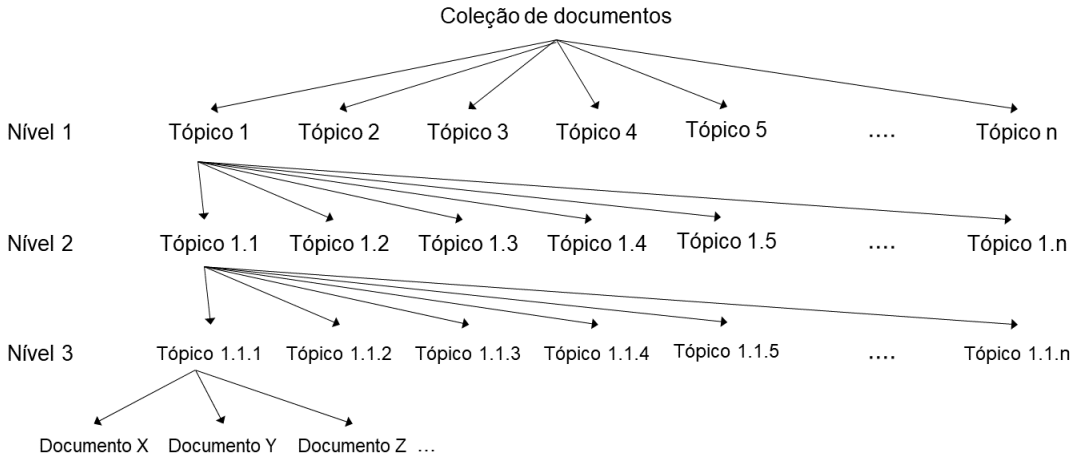

Figura 7 – Organização da coleção em vários níveis de tópicos

Os tópicos são, em seguida, usados para treinar o classificador de textos.

3.4 4ª Etapa - Pré-processamento de Novos Textos

Para que um novo documento seja classificado, fez-se necessário seu pré-processamento. O pré-processamento dos textos consiste essencialmente na extração dos tópicos com a LLM e sua vetorização criando-se *embeddings* representativos do conteúdo de cada documento. De mesma foram que na vetorização dos *tags*, os *embeddings* dos novos textos a serem clusterizados foram gerados usando o *SBERT*.

A *embedding* de cada texto a ser classificado será submetida ao classificador, identificando-se a quais tópicos o novo texto mais se aproxima. A Figura 8 ilustra as 4ª e 5ª etapas - pré-processamento e classificação.

3.5 5ª Etapa - Classificação de Textos

O classificador é um *fine-tuning* do modelo BERT, no qual as *embeddings* são ajustadas conforme o erro de classificação dos documentos aos tópicos. Para isso, utilizamos como treinamento os tópicos e as estruturas de agrupamento da etapa anterior. Essa abordagem permite uma atualização dinâmica dos tópicos, aproveitando a estrutura de agrupamento já existente e ajustando um modelo de linguagem para essa estrutura. Para realizar o *fine-tuning*, foi utilizada a biblioteca *ktrain* (Maiya, 2020), que oferece uma interface de alto nível para ajuste fino de modelos baseados em BERT.

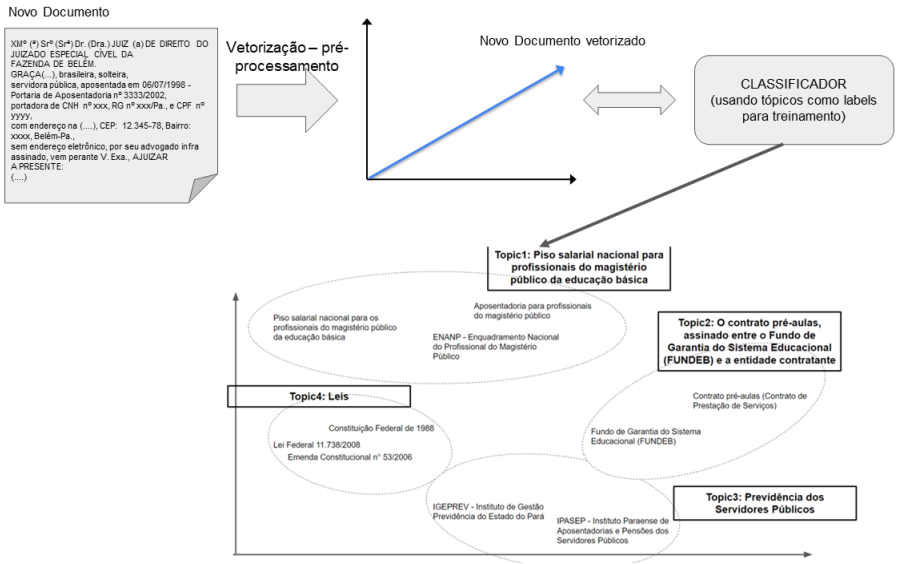

Figura 8 – Pré-processamento e classificação de um novo documento

3.6 6ª Etapa - Atribuição de Tópicos

Finalmente, na etapa de inferência, o documento pré-processado é submetido ao classificador ajustado na etapa anterior. O modelo, após realizar a classificação, produz uma saída com as probabilidades associadas a cada tópico. Utilizando a função *argmax*, identificamos os cinco tópicos com as maiores probabilidades, selecionando-os como os tópicos representativos do novo documento. Esse processo garante que os tópicos mais relevantes sejam atribuídos ao documento, refletindo com precisão seu conteúdo. A figura 9 ilustra um exemplo desse processo, mostrando como o tópico representativo é escolhido por meio de com base nas probabilidades calculadas pelo modelo.

```
# Predição de um novo texto
predictor.predict(Novo_Documento)
[[('Piso salarial nacional para profissionais do magistério público da educação básica', 0.5491581),
  ('Assistência Judiciária', 0.02454061),
  ('Fundo de Garantia da Gestão Previdenciária', 0.084347874),
  ('Lei Federal 11.738/2008', 0.4110818),
  ('Previdência dos Servidores Públicos', 0.17229997),
  ('Educação Básica', 0.08519211)]]
```

Figura 9 – Predição de tópicos de um novo documento

Comparado a um modelo de linguagem grande (LLM), o classificador obtido com o *fine-tuning* oferece respostas significativamente mais rápidas. Isso permite usar a LLM para gerar estruturas de *tags, clusters* e tópicos e então treinar um classificador eficiente. Este classificador é capaz de processar milhões de documentos, tornando-o ideal para aplicações em larga escala na etapa de inferência.

4 AVALIAÇÃO EXPERIMENTAL

O *pipeline* proposto foi implementado em módulos separados, dentre outras razões, para contornar as incompatibilidades de dependências das diversas bibliotecas utilizadas. Ao final de cada módulo, o resultado foi armazenado em um arquivo *pickle*, que foi utilizado como entrada do módulo seguinte.

A avaliação experimental seguiu várias etapas. Inicialmente, os documentos foram extraídos de uma base de dados Postgres e processados por uma LLM para extrair os tópicos principais. Esses tópicos foram convertidos em *embeddings* de 768 dimensões usando a biblioteca SentenceTransformers (SBERT) e normalizados. Em seguida, foram clusterizados com K-Means e cada *cluster* foi nomeado por uma LLM. Por fim, os dados clusterizados foram usados para treinar um classificador, visando transferir a organização em tópicos para grandes bases de dado e futuras classificações.

Esta organização em etapas também tem a vantagem de contornar as incompatibilidades de dependências das diversas bibliotecas utilizadas, uma vez que podem ser modularizadas. A seguir, são apresentados detalhes de cada etapa.

4.1 Extração dos documentos da base de dados

A extração dos documentos foi feita diretamente de uma base de dados *Postgres*, que contém petições iniciais de processos do Tribunal de Justiça do Estado do Pará, muitos deles ainda em andamento, de modo que, a fim de se preservar dados sensíveis das partes envolvidas, as restrições da Lei Geral de Proteção de Dados impedem a divulgação. Os processos a serem usados nos experimentos foram selecionados de forma aleatória diretamente pelo comando SQL, com o uso da expressão RANDOM().

4.2 Extrator de tópicos

O módulo de extração de tópicos utilizou uma LLM para extrair os tópicos dos documentos a serem classificados. Após vários experimentos, por tentativa e erro, chegou-se a um *prompt* que retorna os cinco principais pedidos das petições iniciais. Os pedidos contidos na petição inicial melhor caracterizam o objeto de um processo, para fins de agrupamento dos documentos. Foi desenvolvida uma pequena classe em Python que contém as funções de processamento do pedido e interface com a LLM, vez que tais funções são usadas em vários momentos do *pipeline* proposto. Foram feitos testes com a Llama e ChatGPT, dando-se preferências àquela por ser aberta. Na versão final do experimento, foi utilizada a Llama 3.1, com 40 bilhões de parâmetros, abrigada no LABIC da USP/São Carlos.

O *prompt* utilizado continha, na primeira parte, o texto da petição inicial, seguido do comando com algumas instruções a fim de gerar respostas mais significativas, ao mesmo tempo que sintéticas, para fins de agrupamento das petições (LLM *format enforcing*). As respostas da LLM seguiram o formato "PEDIDO n: (texto)", onde n é um número de 1 a 5, e texto é o conteúdo do pedido (tópico) extraído pela LLM. Após diversos experimentos, o último *prompt* nos experimentos, foi o seguinte:

```
>>><documento>
Você é um modelo de linguagem treinado para processar e analisar textos jurídicos.
Sua tarefa é extrair e listar até cinco principais pedidos de uma petição inicial fornecida.
Siga as instruções abaixo para realizar esta tarefa:
Leia atentamente o documento jurídico fornecido.
Identifique até os cinco principais pedidos discutidos no documento.
Exlua os pedidos de natureza processual, tais como:
  1) pedido de citação da parte contrária;
  2) pedido de assistência judiciária;
  3) Pedido de tutela antecipada;
  4) Pedido de tutela de evidência.
Mantenha apenas os pedidos de direito material.
Estruture sua resposta de forma clara e organizada.
Não responda nada além da lista dos pedidos de direito material.
Use como modelo de resposta a seguinte estrutura:
PEDIDO 1: (segue o pedido 1)
PEDIDO 2: (segue o pedido 2)
PEDIDO 3: (segue o pedido 3)
PEDIDO 4: (segue o pedido 4)
PEDIDO 5: (segue o pedido 5)
```

Figura 10 – *Prompt* usado para a extração dos tópicos

Cada parágrafo da resposta gerada pela LLM correspondia a um tópico, que era relacionado ao processo a que pertence. Desta forma, a saída do módulo de extração foi armazenada um *dataframe* Pandas, onde a primeira coluna continha o número do processo e a segunda um de seus tópicos, podendo cada processo ter até cinco entradas na tabela. Para o conjunto de documentos usado no experimento, foram gerados 2449 tópicos, o que represente uma média de 4,898 tópicos por processo.

Não se percebeu, na análise qualitativa dos *clusters* gerados, grande influência de alucinação da LLM. Imagina-se que isso ocorreu porque a geração dos tópicos está vinculada ao documento que é fornecido junto com o *prompt*. Em outras palavras, no experimento, à LLM não foi demandada a geração de um texto livre, mas um texto que está vinculado ao documento passado no *prompt*. Assim, os tópicos gerados estão, de uma forma ou outra, presentes no documento fornecido. Percebe-se que a qualidade do

resultado do *pipeline* está mais vinculado, de fato, à pertinência dos tópicos para os fins de agrupamento. Explico. A geração de uma tópico, por exemplo, como "*Que seja condenado, ainda, a Autarquia demandada ao pagamento das custas processuais e dos honorários advocatícios de sucumbência, que se requer sejam arbitrados em 20% (vinte por cento) do valor da causa*", em que pese a rigor estar contido no documento analisado, não o distingue de forma significativa, do ponto de vista do negócio, dos demais documentos, já que presente em praticamente todas as petições.

Em seguida, o *dataframe* com os pedidos foi armazenado em um arquivo *Pickle*.

4.3 Módulo de geração e nomeação dos *clusters*

A fim de serem clusterizados, foram gerados *embeddings* de cada um dos tópicos, com dimensão de 768. As *embeddings* foram geradas utilizando-se a biblioteca Sentence-Transformers (SBERT).

A seguir, as *embeddings* foram normalizadas, para se limitarem ao intervalo [0,1]. A quantidade de *clusters* foi determinada utilizando-se a regra prática da raiz quadrada do número de observações (neste caso, tópicos), ou seja, raiz quadrada de 2449, o que resulta em 49 *clusters* (Joseph; Jeberson; Jeberson, 2010).

Criados os *embeddings*, passou-se à clusterização com o uso do algoritmo K-Means. Utilizou-se a implementação do K-Means da biblioteca Scikit-Learn (Pedregosa *et al.*, 2011).

Ao final, utilizou-se novamente a LLM para nomear cada um dos *clusters*. Para tanto, o texto de todos os tópicos com compunham um *cluster* foram concatenados e apresentados à LLM para gerar um título com até 20 palavras que expressasse seu conteúdo. Segue o *prompt* usado para nomear os *clusters*:

```
>>><documento>
Você é um modelo de linguagem treinado para processar e analisar textos jurídicos.
Sua tarefa é dar um título para o texto que se segue, em até 20 palavras, que expresse  o seu
conteúdo.

==========
```

Figura 11 – *Prompt* usado para a nomeação dos *clusters*.

O *dataframe* final foi salvo em um arquivo *pickle* que serviu de entrada para o módulo seguinte, o classificador.

4.4 Módulo de definição e treinamento do modelo do classificador

Utilizou-se a biblioteca *ktrain* para gerar um classificador, que foi treinado com os dados gerados na etapa anterior. O modelo gerado possuía 177 milhões de parâmetros, com 12 camadas de codificação, além das camadas de entrada e saída. O seguinte relatório do *ktrain/keras* resume a arquitetura do modelo gerado:

O treinamento foi feito em seis épocas, tendo a acurácia partido de 20,82%, na primeira, chegando a 72,38% na sexta. Cada época levou em média 14 minutos de tempo de processamento no Google Colab, em um ambiente de execução com CPU e RAM alta. Finalizou-se o treinamento com seis épocas porque a melhora na acurácia saiu de cerca de 23% entre as duas primeiras épocas, para cerca de 3% entre as duas últimas.

A função *validate()* retorna um relatório do desempenho do classificador durante o treinamento, reportando a precisão, *recall*, *f1-score* e suporte, de todos os *clusters*, conforme se segue:

Tabela 1 – Métricas dos *clusters* geradas pelo *ktrain*.

	precision	**recall**	**f1-score**	**suporte**
0	0.67	0.64	0.65	25
1	0.64	0.43	0.51	21
2	1.00	0.80	0.89	5
3	0.50	0.57	0.53	14
4	1.00	1.00	1.00	1
5	1.00	0.25	0.40	4
6	0.58	1.00	0.74	7
7	0.71	0.83	0.77	12
8	0.80	0.89	0.84	18
9	1.00	1.00	1.00	5
10	0.71	0.81	0.76	21
11	0.82	0.53	0.64	17
12	0.40	0.62	0.48	13
13	0.58	0.32	0.41	22
14	0.75	0.56	0.64	16
15	0.10	0.12	0.11	8
16	1.00	0.77	0.87	13
17	1.00	0.75	0.86	16
18	0.44	0.58	0.50	19
19	1.00	0.33	0.50	6

Continua na próxima página

Tabela 1 – continuação da página anterior

	Precision	**recall**	**f1-score**	**suporte**
20	0.55	0.30	0.39	20
21	0.94	1.00	0.97	15
22	0.91	1.00	0.95	29
23	0.94	0.91	0.93	35
24	0.87	1.00	0.93	27
25	0.76	0.87	0.81	15
26	0.86	0.95	0.90	20
27	1.00	1.00	1.00	21
28	1.00	0.92	0.96	13
29	0.52	0.71	0.60	17
30	0.88	1.00	0.94	37
31	1.00	1.00	1.00	9
32	0.43	0.25	0.32	12
33	0.65	0.88	0.75	25
34	0.60	0.90	0.72	10
35	0.88	0.88	0.88	17
36	0.67	0.31	0.42	13
37	0.00	0.00	0.00	4
38	0.50	0.80	0.62	5
39	1.00	0.57	0.73	7
40	0.47	0.67	0.55	12
41	0.50	0.56	0.53	9
42	0.30	0.86	0.44	7
43	0.88	0.71	0.79	21
44	0.62	0.38	0.48	13
45	0.00	0.00	0.00	4
46	0.56	0.31	0.40	16
47	0.70	0.82	0.76	17
48	0.75	0.82	0.78	22
accuracy			0.72	735
macro avg	0.70	0.68	0.67	735
weighted avg	0.73	0.72	0.71	735

Percebe-se que há um grande desbalanço entre os *clusters*, com suporte que varia de 1 a 37, o que indica que a escolha de dados para treinamento com um número de representantes de cada classe mais equilibrado pode levar a um desempenho melhor do classificador.

Ao final do treinamento, o modelo do preditor foi salvo em um arquivo h5, para que se possa usar o classificador em um momento posterior, sem a necessidade de novo treinamento a cada uso.

4.5 Utilização do classificador

Inicialmente, preparou-se o ambiente do Colab para que o modelo seja recarregado a partir do arquivo de saída do módulo anterior. Para tanto, foram instalados o TensorFlow e o Keras, para em seguida instalar o Keras e, somente após isso, foi feito o carregamento do modelo já treinado.

Em seguida, preparou-se o ambiente para executar a LLM, instalando-se a biblioteca Openai e a biblioteca LLMextractor, que contém as funções de interfaceamento com a LLM. Configuradas as variáveis da LLM, o módulo está pronto para a classificação de novos documentos.

Para tanto, faz-se necessária a extração dos tópicos com a LLM para, em seguida, ser submetido ao classificador.

5 CONCLUSÕES

Por resultado qualitativo do agrupamento dos documentos, entende-se a pertinência temática entre os documentos que compõe o cluster, o que deve ser avaliado com a participação de um expert do domínio do negócio a que os documentos se refiram. O agrupamento será tão melhor quanto menos documentos alheios ao tema principal do agrupamento houver.

Ao final dos experimentos, percebeu-se que os resultados qualitativos dos agrupamentos e classificação produzidos pelo pipeline estão intrinsecamente ligados à qualidade das respostas da LLM. Percebeu-se que dois fatores influenciaram de forma significativa na melhora das respostas da LLM. O primeiro diz respeito à evolução da tecnologia dos grandes modelos de linguagem em si. Nos cerca de dez meses que durou os trabalhos, chama a atenção a evolução das LLMs, tanto na qualidade das respostas, quanto tamanho de contexto por elas admitidos. Nos testes iniciais, por exemplo, com o ChatGPT 3.5, havia a limitação de uma janela de contexto de 4096 tokens. Atualmente, a versão do LLama 3.0 Gradient tem um tamanho de contexto entre 8.000 e 1.040.000 tokens. O tamanho do contexto, para o tipo de aplicação que se teve em mente neste trabalho, é fator extremamente limitante. Em razão disso, nos primeiros experimentos, ajustou-se os *prompts* para se adequar à limitação do tamanho de contexto do modelo utilizado, já que facilmente os documentos utilizados excediam o tamanho do contexto permitido. Chegou-se a implementar uma estratégia de segmentar o documento em "janelas" deslizantes, submetendo as diversas janelas à LLM a fim de se extrair os tópicos desejados, em mais de uma consulta por documento. Nos experimentos finais, tornou-se desnecessário se utilizar esta estratégia, pois a LLM utilizada, LLama 3.0, já permitia submeter os documentos em um só prompt para a extração dos tópicos.

Quanto à qualidade das respostas das LLMs, percebeu-se que os modelos estão cada vez mais aderentes às instruções dos prompts. No início, era comum obter-se respostas em inglês e com formatos completamente aleatórios, de modo que o tratamento das respostas ficava bastante prejudicado. Nos experimentos feitos com as versões mais recentes, a LLM produz respostas bastante padronizadas, a partir de instruções que indiquem um determinado padrão de resposta desejado ("LLM format enforcer").

Um aspecto positivo do pipeline proposto diz respeito à sua independência quanto ao modelo de LLM a ser usado. Com a utilização da biblioteca OpenAI, consegue-se abstrair o modelo da LLM usado, permitindo-se a alteração de um modelo a outro apenas pela modificação dos parâmetros de conexão. Com isso, é possível se utilizar o pipeline com uma LLM local, evitando-se o tráfego dos dados da instituição na rede externa, o que garante a privacidade no tratamento dos dados sensíveis do negócio.

Mesmo com a evolução das LLMs, ainda não se vislumbra a viabilidade próxima de se obter uma solução de agrupamento de grandes documentos com o uso exclusivo dos grandes modelos de língua. Portanto, o uso combinado de diferentes técnicas de IA – grandes modelos de língua e técnicas de ML clássicas – permite que se obtenha uma solução escalável e eficiente para o pipeline proposto.

Outro aspecto importante da solução é a possibilidade de se obter agrupamento de documentos grandes sem a intervenção humana. Mesmo que o resultado do agrupamento seja submetido à crítica do expert que eventualmente pode fazer ajustes aos resultados, o fato de o modelo apresentar uma primeira proposta de agrupamento sem intervenção humana representa um atrativo para a utilização prática do pipeline.

Várias linhas de pesquisa se abrem para aprimoramento dos resultados, tais como: avaliação de outros critérios para determinação da quantidade de *clusters*, havendo na literatura diversas alternativas; outras estratégias para agrupamento dos documentos, tal como o tratamento dos tópicos de um mesmo documento como uma unidade e não como tópicos autônomos; e, principalmente, novos experimentos com a engenharia de *prompts*, já que os *prompts* se mostraram fundamentais para a extração de tópicos de boa qualidade semântica.

REFERÊNCIAS

AGUIAR, A. *et al.* Using topic modeling in classification of brazilian lawsuits. *In*: SPRINGER. **International conference on computational processing of the portuguese language**. [*S.l.: s.n.*], 2022. p. 233–242.

AJEVSKI, M. *et al.* Chatgpt and the future of legal education and practice. **The Law Teacher**, Taylor & Francis, p. 1–13, 2023.

AMAZON. **O que é engenharia por prompt**. 2024. https://aws.amazon.com/pt/what-is/prompt-engineering/#:~:text=de%20IA%20generativa%3F-,O%20que%20%C3%A9%20engenharia%20de%20prompt%3F,para%20gerar%20os%20resultados%20desejados. Accessed: 04/03/2024.

BURMAN, A.; BRADFORD, E. Building an optimized algorithm that provides summaries of legal documents.

CHALKIDIS, I. *et al.* Legal-bert: The muppets straight out of law school. **arXiv preprint arXiv:2010.02559**, 2020.

CHALKIDIS, I. *et al.* Lexfiles and legallama: Facilitating english multinational legal language model development. **arXiv preprint arXiv:2305.07507**, 2023.

CLUSTERING. https://scikit-learn.org/stable/modules/clustering.html. Acessado em: 19/08/2024.

DAI, Y. *et al.* Laiw: A chinese legal large language models benchmark (a technical report). **arXiv preprint arXiv:2310.05620**, 2023.

DEVLIN, J. *et al.* **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. 2019.

FEI, Z. *et al.* Lawbench: Benchmarking legal knowledge of large language models. **arXiv preprint arXiv:2309.16289**, 2023.

FELTEN, E.; RAJ, M.; SEAMANS, R. How will language modelers like chatgpt affect occupations and industries? **arXiv preprint arXiv:2303.01157**, 2023.

GARNEAU, N. *et al.* Criminelbart: A french canadian legal language model specialized in criminal law. *In*: **Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law**. [*S.l.: s.n.*], 2021. p. 256–257.

GROOTENDORST, M. Bertopic: Neural topic modeling with a class-based tf-idf procedure. **arXiv preprint arXiv:2203.05794**, 2022.

GROSSMAN, M. R. *et al.* The gptjudge: Justice in a generative ai world. **Duke Law & Technology Review**, v. 23, n. 1, 2023.

GUHA, N. *et al.* Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. **Advances in Neural Information Processing Systems**, v. 36, 2024.

IOANNIDIS, J. *et al.* Gracenote. ai: Legal generative ai for regulatory compliance. 2023.

JOSEPH, W.; JEBERSON, W.; JEBERSON, K. Count based k-means clustering algorithm. **Int. J. Curr. Eng. Technol**, v. 5, p. 1249–1253, 2010.

KATZ, D. M. *et al.* Gpt-4 passes the bar exam. **Philosophical Transactions of the Royal Society A**, The Royal Society, v. 382, n. 2270, p. 20230254, 2024.

KUPPA, A.; RASUMOV-RAHE, N.; VOSES, M. Chain of reference prompting helps llm to think like a lawyer. *In*: **Generative AI+ Law Workshop**. [*S.l.: s.n.*], 2023.

LI, H. *et al.* Sailer: structure-aware pre-trained language model for legal case retrieval. *In*: **Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval**. [*S.l.: s.n.*], 2023. p. 1035–1044.

LICARI, D.; COMANDÈ, G. Italian-legal-bert: A pre-trained transformer language model for italian law. *In*: **CEUR Workshop Proceedings (Ed.), The Knowledge Management for Law Workshop (KM4LAW)**. [*S.l.: s.n.*], 2022.

MAIYA, A. S. ktrain: A low-code library for augmented machine learning. **arXiv preprint arXiv:2004.10703**, 2020.

OLIVEIRA, R. S. D.; NASCIMENTO, E. G. S. Brazilian court documents clustered by similarity together using natural language processing approaches with transformers. **arXiv preprint arXiv:2204.07182**, 2022.

PEDREGOSA, F. *et al.* Scikit-learn: Machine learning in python. **the Journal of machine Learning research**, JMLR. org, v. 12, p. 2825–2830, 2011.

PERLMAN, A. The implications of chatgpt for legal services and society. **Available at SSRN 4294197**, 2022.

REIMERS, N.; GUREVYCH, I. **Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks**. 2019.

SAVELKA, J. *et al.* Can gpt-4 support analysis of textual data in tasks requiring highly specialized domain expertise? **arXiv preprint arXiv:2306.13906**, 2023.

SAVELKA, J. *et al.* Explaining legal concepts with augmented large language models (gpt-4). **arXiv preprint arXiv:2306.09525**, 2023.

SENTENCETRANSFORMERS Documentation. https://sbert.net/. Acessado em: 01/09/2024.

SILVEIRA, R. *et al.* Topic modelling of legal documents via legal-bert. **Proceedings http://ceur-ws org ISSN**, v. 1613, p. 0073, 2021.

SILVEIRA, R. *et al.* Legalbert-pt: A pretrained language model for the brazilian portuguese legal domain. *In*: SPRINGER. **Brazilian Conference on Intelligent Systems**. [*S.l.: s.n.*], 2023. p. 268–282.

STEENHUIS, Q.; COLARUSSO, D.; WILLEY, B. Weaving pathways for justice with gpt: Llm-driven automated drafting of interactive legal applications. **arXiv preprint arXiv:2312.09198**, 2023.

SUN, Z. A short survey of viewing large language models in legal aspect. **arXiv preprint arXiv:2303.09136**, 2023.

TRAUTMANN, D.; PETROVA, A.; SCHILDER, F. Legal prompt engineering for multilingual legal judgement prediction. **arXiv preprint arXiv:2212.02199**, 2022.

WANG, X. *et al.* Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *In*: **Workshop on Efficient Systems for Foundation Models@ ICML2023**. [*S.l.: s.n.*], 2023.

WU, Y. *et al.* Precedent-enhanced legal judgment prediction with llm and domain-model collaboration. **arXiv preprint arXiv:2310.09241**, 2023.

YU, F.; QUARTEY, L.; SCHILDER, F. Legal prompting: Teaching a language model to think like a lawyer. **arXiv preprint arXiv:2212.01326**, 2022.

YUAN, M. *et al.* Bringing legal knowledge to the public by constructing a legal question bank using large-scale pre-trained language model. **Artificial Intelligence and Law**, Springer, p. 1–37, 2023.

YUE, L. *et al.* Fedjudge: Federated legal large language model. **arXiv preprint arXiv:2309.08173**, 2023.

YUE, S. *et al.* Disc-lawllm: Fine-tuning large language models for intelligent legal services. **arXiv preprint arXiv:2309.11325**, 2023.