ARMANDO CESAR MARQUES DE CASTRO

MANUAL DE INTELIGÊNCIA ARTIFICIAL EM DIREITO

Um guia prático para profissionais do Direito extraírem o máximo das ferramentas de inteligência artificial generativa

2026

EDITORA JusPODIVM

www.editorajuspodivm.com.br

Capítulo V

ENGENHARIA DE CONTEXTO

Ao tratar de prompts, vimos que a janela de contexto é um recurso finito e que sua capacidade efetiva é sistematicamente menor do que a anunciada, fenômeno ilustrado pelo *Lost in the Middle* e pela distância entre MCW (capacidade de entrada) e MECW (capacidade efetiva). A governança documental tratada no capítulo anterior, por sua vez, estabelece como o acervo jurídico se organiza em repouso: taxonomias, metadados, padrões, identificadores. Neste capítulo, fechamos o ciclo, tratando do que ocorre quando esse acervo se move, com suas partes sendo selecionadas, comprimidas e transportadas para a janela de contexto de um modelo de linguagem para responder uma pergunta concreta. A essa movimentação dá-se o nome técnico de **engenharia de contexto.** O termo descreve a habilidade de fornecer todo o contexto para que uma tarefa seja resolvida de forma plausível pelo modelo, representando a delicada "arte e ciência de preencher a janela de contexto a informação certa para o próximo passo".[1]

Em setembro de 2025 a Anthropic publicou documento técnico-conceitual que chancelou o paradigma sob o título *Effective Context Engineering for AI Agents.* A empresa afirma que enquanto *prompt engineering* designa os métodos para escrever e organizar instruções de modelos de linguagem visando a resultados ótimos, *context engineering* designa o

1. KARPATHY, Andrej. **Software is Changing (Again)** [palestra]. Y Combinator Ai Startup School. São Francisco, jun. 2025. Disponível em https://www.youtube.com/watch?v=LCEmiRjPEtQ acesso em 04 mai. 2026.

conjunto de estratégias para curar e manter o conjunto ótimo de tokens — isto é, de informação — durante a inferência do modelo, incluindo toda a informação adicional que pode aparecer ali além dos próprios prompts.[2]

O contexto contém, em geral:

- **instruções de sistema** (*system prompt*): a camada persistente que define papel, tom, restrições e diretrizes;
- **a tarefa atual do usuário**: o pedido específico daquela interação;
- **histórico da conversa**: turnos anteriores que permanecem na janela;
- **documentos recuperados**: trechos extraídos de bases vetoriais por sistemas RAG ou inseridos manualmente;
- **exemplos** *few-shot*: demonstrações de formato e padrão esperados;
- **definições de ferramentas**: descrições daquilo que o modelo pode acionar (busca na web, cálculo, execução de código);
- **resultados de chamadas de ferramentas**: dados retornados durante a execução;[3]
- **memória externa estruturada**: anotações persistentes que o modelo escreveu em interações anteriores e pode reler.[4]

2. ANTHROPIC. Effective context engineering for AI agents, op. cit.
3. Modelos de IA modernos podem acionar ferramentas externas durante uma interação — busca na web, consulta a base jurisprudencial, calculadora, execução de código, leitura de planilha. O conteúdo que essas ferramentas devolvem (uma ementa, um número, um trecho de documento) entra na janela de contexto e passa a ser "lido" pelo modelo na rodada seguinte do raciocínio. Funciona, em analogia, como o estagiário a quem se pede uma pesquisa rápida: o que ele traz de volta passa a integrar o material sobre o qual o profissional decide a próxima providência.
4. Para superar o limite da janela de contexto, sistemas mais avançados permitem que o modelo grave anotações em um espaço de armazenamento separado, que persiste entre sessões. Em interações futuras, o modelo pode reler essas anotações quando relevantes. Funciona, em analogia, como o caderno de anotações que o profissional mantém ao longo de meses sobre um caso complexo: as anotações sobrevivem ao final do expediente e estão lá no retorno. Exemplos atuais dessa funcionalidade incluem o recurso "memória" do ChatGPT, o sistema de memórias do Claude e a memória persistente do Gemini. Diferencia-se do histórico da conversa (que está dentro da janela atual) e dos documentos carregados em um Project ou Gem (que também estão dentro da janela): a memória externa estruturada vive fora da janela e é lida sob demanda.

Cada uma dessas camadas consome tokens, consumindo pela finita janela de atenção. Andrej Karpathy criou uma metáfora interessante: os parâmetros do modelo são como a CPU, fixos no momento da execução, fazendo o processamento e não sendo reprogramados a cada uso. A janela de contexto é a memória RAM, transitória, o espaço onde o modelo "carrega" o que precisa ver para realizar a tarefa a cada instante.

A governança documental, tratada na Parte III, organiza o acervo. A engenharia de contexto, tratada neste capítulo, decide o que do acervo entra na janela em cada caso concreto.

5.1. O APODRECIMENTO DE CONTEXTO (*CONTEXT ROT*)

O problema técnico que torna a engenharia de contexto fundamental foi definido em 2025 com a publicação do relatório técnico *Context Rot: How increasing Input Tokens Impacts LLM Performance.*[5] O documento inicia afirmando que existe uma presunção de que os modelos de linguagem ampla processam o contexto de maneira uniforme, lidando com o token 10.000 da mesma forma com que lidam com o 100, mas que essa afirmativa não se sustenta na prática, mesmo em tarefas simples.

O método partiu de uma crítica ao benchmark mais utilizado para testar capacidades de contexto longo, conhecido como *Needle in a Haystack* "agulha no palheiro". Esse teste, popularizado em 2023, mede se o modelo consegue localizar uma frase específica (a "agulha") inserida em um grande volume de texto (o "palheiro"). Sua limitação, apontou a Chroma, é que ele mede apenas recuperação lexical: encontrar uma frase que está literalmente presente. Não mede compreensão, raciocínio, resistência a informação enganosa ou capacidade de discriminar relevância, operações que correspondem ao trabalho jurídico real.

Para superar essa limitação, a Chroma desenhou quatro experimentos controlados que mantinham a complexidade da tarefa constante e variavam apenas o comprimento da entrada. O primeiro testou similaridade semântica entre pergunta e resposta — quanto a pergunta podia ser respondida por correspondência literal, quanto exigia compreensão. O segundo

5. HONG, Kelly; TROYNIKOV, Anton; HUBER, Jeff. *Context Rot: How Increasing Input Tokens Impacts LLM Performance.* ***Chroma Technical Report****, July 14, 2025.* Disponível em https://www.trychroma.com/research/context-rot acesso em 04 mai. 2026.

introduziu **distraidores**: trechos topicamente relacionados à resposta correta, mas que não respondem efetivamente à pergunta — o equivalente, em termos jurídicos, a precedentes superficialmente análogos, mas materialmente inaplicáveis. O terceiro variou a **estrutura do palheiro**, comparando textos logicamente coerentes a textos com sentenças embaralhadas. O quarto testou diálogo conversacional prolongado e replicação de texto.

Os resultados foram convergentes em todos os modelos testados. Modelos que atingiam acerto superior a 95% em testes simples de recuperação caíram para faixas entre 60% e 70% quando submetidos a tarefas semanticamente mais realistas, particularmente aquelas que envolviam fazer a distinção entre informação relevante e ruído plausível.

Um achado merece destaque por ser contraintuitivo. Os modelos performaram **melhor em palheiros embaralhados, com sentenças em ordem aleatória, do que em textos com fluxo lógico coerente**. A explicação proposta pelos pesquisadores é incômoda: textos coerentes parecem criar distraidores mais plausíveis, porque a coesão narrativa torna mais difícil para o modelo identificar qual trecho é, de fato, a resposta à pergunta. O fluxo lógico do texto compete pela atenção do modelo, e às vezes vence. Para quem trabalha com peças processuais e laudos técnicos, documentos cuja virtude é justamente a coerência narrativa, a observação é desconfortável.

O apodrecimento de contexto não é um defeito de um ou outro fornecedor que será corrigido na próxima versão. É um traço que acompanha os modelos de linguagem ampla em virtude de três fatores de arquitetura, próprios da forma como modelos de linguagem foram concebidos.

O primeiro é a **diluição da atenção**. À medida que o número de tokens no contexto cresce, o modelo precisa distribuir sua capacidade de relacionar cada palavra com cada outra palavra sobre um conjunto cada vez maior dessas relações. A precisão de cada relação individual diminui. O segundo é o **viés do treino**: os modelos aprenderam a partir de um universo textual dominado por sequências curtas — *posts*, artigos, parágrafos —, e por isso desenvolveram muito mais habilidade para raciocinar dentro de um parágrafo do que para conectar informação separada por dezenas de milhares de palavras. O terceiro é a **adaptação posicional**: quando os fabricantes ampliam a janela de um modelo originalmente treinado para textos menores, até comportar centenas de milhares ou um milhão de tokens, é preciso adaptar a forma como o modelo identifica a posição de

cada palavra na sequência, e essa adaptação introduz alguma perda de precisão, sobretudo nas regiões mais distantes da extensão original.

Por isso, **o contexto deve ser tratado como um recurso finito com retornos marginais decrescentes.** Como humanos, que têm capacidade limitada de memória de trabalho, modelos de linguagem operam com um "orçamento de atenção" do qual dispõem ao processar grandes volumes de contexto. Cada novo token introduzido esgota esse orçamento em alguma medida, ampliando a necessidade de curar com cuidado os tokens disponibilizados ao modelo.[6] Assim, considere sempre que a **capacidade efetiva de processamento de um modelo de linguagem é sistematicamente menor do que a capacidade anunciada por seu fabricante.** É preciso ser crítico em relação ao que colocar na janela de contexto, avaliando cada documento por seu custo (o quanto de espaço ocupa) e benefício (o quanto agrega para o desempenho da tarefa.

5.1.1. As quatro estratégias de gestão

As categorias ***write, select, compress, isolate*** são a síntese básica do campo da engenharia de contexto.[7] A apresentação a seguir mantém os termos originais em inglês, por se tratar de vocabulário técnico ainda em consolidação no Brasil, com tradução proposta entre parênteses. Para cada estratégia, tratamos da definição, do mecanismo e de uma aplicação concreta ao trabalho jurídico.

Write significa escrever fora do contexto, em armazenamento externo persistente, para que esteja disponível quando o modelo precisar sem ocupar a janela durante todo o tempo. O modelo (ou o profissional que o opera) registra notas, planos, conclusões parciais ou referências em um espaço externo: um arquivo, uma base de dados, uma memória persistente como as do ChatGPT e do Claude. Quando essa informação for relevante para uma tarefa futura, ela é trazida de volta para a janela. Quando não for, permanece fora, sem competir pela atenção do modelo nas tarefas em que sua presença seria ruído.

6. ANTHROPIC. Effective context engineering for AI agents, op. cit.
7. LANGCHAIN. **Context Engineering**. Disponível em https://www.langchain.com/blog/context-engineering-for-agents acesso em 04 mai. 2026.

O exemplo mais imediato é o uso de **memórias persistentes** dos chatbots para registrar instruções estáveis sobre o escritório, o gabinete ou a área de atuação, como preferências de estilo de redação, terminologia padronizada, jurisprudências de referência habitual, regras de formatação, ritos a observar. O profissional escreve a memória uma vez; ela está disponível em toda interação subsequente, sem ocupar a janela quando o assunto da vez não a invoca. Outro exemplo é o uso de **anotações estruturadas em arquivos** dentro de um Project no Claude, em um GPT personalizado em um Gem do Gemini: notas de andamento de caso, *checklists* de instrução, sumários de teses defendidas, material que o modelo lê quando necessário e ignora quando não. Em ambos os casos, o princípio é o mesmo: **a janela trabalha em cima do conhecimento; ela não é depositário do conhecimento**.

Select quer dizer selecionar o que entra. Trazer para a janela de contexto apenas a informação relevante para a tarefa em curso. Antes de inserir um processo em um Project para análise, o advogado pode **selecionar as peças centrais** (petição inicial, contestação, sentença, acórdão) e deixar de fora o que sabidamente é ruído (despachos administrativos, certidões cartorárias, juntadas formais). O ato de seleção, embora aparentemente trivial, é em si engenharia de contexto: cada peça periférica que se exclui é orçamento de atenção que se preserva para as peças centrais.

Compress é a compressão do que está dentro. A primeira é a **sumarização**: substituir um trecho longo por uma versão sintética, frequentemente gerada pelo próprio modelo. Em conversas multi-turno extensas, por exemplo, o histórico das primeiras dezenas de mensagens pode ser substituído por um sumário consolidado de duas ou três páginas — preservando as decisões e conclusões alcançadas, descartando as voltas e refinamentos que levaram até elas e iniciando um novo chat. A segunda é a **poda** (*trimming*): remoção de partes da informação segundo regras explícitas — eliminar mensagens mais antigas que certo limite, descartar trechos identificados como off-topic, retirar formatação verbosa que consome tokens sem trazer conteúdo.

A aplicação mais frequente, no trabalho jurídico, é a **substituição da íntegra de documento longo por um sumário focado**. Em vez de inserir um acórdão de oitenta páginas inteiro em uma sessão de análise, o profissional pode trabalhar primeiro em uma sessão dedicada à elaboração de

um sumário estruturado desse acórdão (relatório, voto condutor, fundamentos centrais, dispositivo) e usar esse sumário, não a íntegra, nas sessões subsequentes em que o acórdão precisar estar à vista.

Atenção: a sumarização é exatamente o ponto em que erros se tornam invisíveis. Um sumário que perde um detalhe relevante carrega esse erro silenciosamente para todas as análises posteriores. O profissional do Direito deve manter, sempre, o documento original acessível e revisar criticamente os sumários produzidos pelo modelo antes de tratá-los como insumo confiável.

Por fim, ***isolate***, ou isolar contextos paralelos. Em vez de manter um único contexto que tenta conter tudo, dividir o trabalho entre contextos separados, cada um com seu próprio escopo, instruções e conjunto reduzido de informação, integrando apenas os resultados. Esta é a estratégia mais sofisticada do conjunto, e a mais associada à arquitetura de agentes e multiagentes. A ideia é que tarefas complexas frequentemente decompõem-se em subtarefas relativamente independentes, cada uma das quais exigiria, sozinha, uma janela de contexto manejável. Em vez de empilhar todas as subtarefas na mesma janela, o sistema distribui cada uma a um "subagente", uma instância do modelo configurada especificamente para aquela subtarefa, com acesso apenas ao material que dela diz respeito. Os subagentes trabalham em paralelo ou em sequência; um agente coordenador integra os resultados.

Para o profissional individual, em 2026, a *isolate* manifesta-se em forma simplificada, sem necessidade de sistemas multi-agente. Há **três aplicações operáveis com as ferramentas atuais.** A primeira é a **separação por sessão**: em vez de discutir todas as facetas de um caso em uma única conversa que se estende indefinidamente, o profissional abre sessões distintas para subtarefas distintas uma para análise probatória, outra para enquadramento jurídico, outra para redação. Cada sessão começa com a janela limpa, focada na subtarefa. A segunda é a **separação por projetos, já amplamente discutida.** A terceira é a **separação por etapa do fluxo**: tratar pesquisa, análise e redação como etapas com contextos separados, transferindo entre elas apenas os resultados consolidados, não os rascunhos intermediários. Em todos os casos, o princípio é o mesmo: **contextos especializados produzem melhores respostas que contextos enciclopédicos.**

5.1.2. Fluxos no trabalho jurídico

A primeira aplicação prática da engenharia de contexto começa **antes** de qualquer interação com o modelo. Diante de um caso volumoso — autos de cinco mil páginas, *due diligence* sobre acervo contratual, inquérito civil instruído por dezenas de relatórios técnicos, a tentação inicial do profissional é carregar todo o material no ambiente da IA e perguntar. A engenharia de contexto inverte o ponto de partida: a primeira pergunta não é "como pergunto?", mas "o que **não** precisa estar à vista do modelo para que esta tarefa seja bem realizada?".

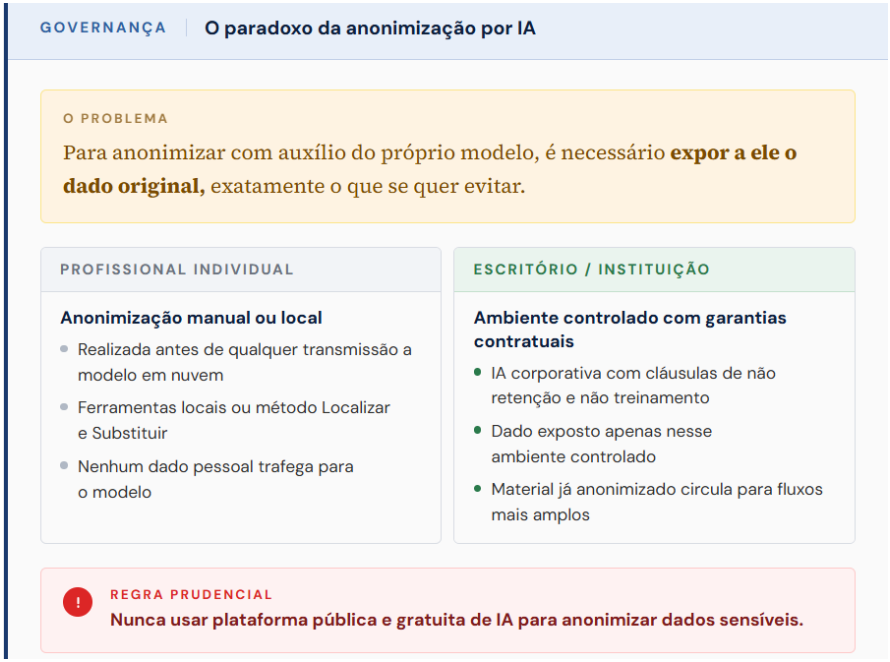
Duas operações compõem o pré-processamento bem feito. A primeira é a **extração**. Documentos jurídicos chegam em formatos heterogêneos como PDFs nativos, PDFs digitalizados, imagens, páginas de sistema. Extrair o texto bruto com qualidade, separar peças relevantes de juntadas formais, converter laudos digitalizados em texto pesquisável: tudo isso é trabalho prévio à interação com o modelo. Ferramentas gratuitas como o PDF24, o PDFsam ou o reconhecimento óptico de caracteres do Adobe Acrobat resolvem a parte mecânica. O que o modelo recebe depois desse trabalho é texto limpo.

A segunda é a **poda de irrelevantes**. Em processos brasileiros, é comum que a maior parte das páginas seja composta por elementos de baixa densidade informacional para a análise jurídica de mérito: certidões cartorárias, juntadas administrativas, ofícios de remessa, despachos formais. Em um inquérito de quinhentas páginas, tipicamente menos da metade contém peças cujo conteúdo importa para a tese jurídica em construção. Identificar e descartar o restante **antes** de inserir o material em uma sessão de IA é, sozinho, um dos maiores ganhos disponíveis de qualidade e não exige tecnologia avançada.

Eventualmente, a depender dos dados tratados e da ferramenta utilizada, pode ser necessário trabalhar com a anonimização. Antes de inserir documentos em ambientes de IA, sobretudo aqueles que não oferecem garantias contratuais robustas de não retenção e não treinamento sobre os dados, dados pessoais identificáveis devem ser substituídos por marcadores genéricos. A operação protege o titular dos dados e, por reduzir o

material a essência abstrata do problema, frequentemente melhora a qualidade da resposta do modelo, que passa a operar sobre os fatos juridicamente relevantes sem a interferência cognitiva de informação acessória.

Em síntese, cada página irrelevante que se exclui antes da inserção é orçamento de atenção que se preserva para as páginas que importam.



Concluído o pré-processamento, resta material que ainda pode ser volumoso. O Segundo movimento operacional é decidir se esse material entra de uma vez ou em partes. A decisão depende da natureza da tarefa. Tarefas que exigem **visão de conjunto**, como identificação de inconsistências entre peças, construção de linha do tempo processual, comparação entre versões de um contrato, exigem que o material esteja simultaneamente disponível, e o particionamento prejudica o resultado. Já tarefas que se decompõem em **subtarefas independentes** como análise de cláusula a cláusula em revisão contratual, exame ponto a ponto de quesitos periciais, elaboração de sumários por capítulo de relatório, beneficiam-se de particionamento, porque cada subtarefa demanda apenas uma fração do material e a presença simultânea do restante apenas dilui a atenção do modelo. Esta subseção combina ***select*** (escolha do que vai para cada partição) com ***isolate*** (cada partição opera em sua própria sessão).

Em conversas que se prolongam, pode ocorrer uma degradação significativa de resultado.[8] Para mitigar o problema, pode-se recarregar peças

8. LABAN, Philippe *et al*. **LLMs Get Lost in Multi-Turn Conversation**. Disponível em https://arxiv.org/abs/2505.06120 acesso em 04 mai. 2026.

centrais, reapresentando-as em mensagens posteriores, trazendo-as para a região da janela em que o modelo opera com mais atenção. Se a sessão se prolongar muito, frequentemente é melhor abrir nova sessão e começá-la com um sumário consolidado das conclusões alcançadas até ali, fornecido como ponto de partida, do que insistir na sessão antiga. O modelo recebe contexto reduzido e focado, em vez de histórico extenso e disperso.

Sessões em que o modelo passa a errar, contradizer-se, alucinar referências ou retornar ao mesmo ponto após correção têm, frequentemente, danos cumulativos no contexto que não se revertem por novos prompts de correção. O movimento mais eficaz, nesses casos, é encerrar a sessão e reabrir com material limpo. Insistir é, em regra, perda. Esta subseção opera ***compress*** (sumarizar para recomeçar) e ***isolate*** (separar em sessões distintas).

PARTE III

GOVERNANÇA DE DADOS E HORIZONTE INSTITUCIONAL

Capítulo I

ANTES DO ALGORITMO: A GOVERNANÇA DE DADOS COMO CONDIÇÃO PRÉVIA

A discussão sobre inteligência artificial no direito inverte frequentemente a ordem de importância de determinados conceitos. Fala-se em modelos, arquiteturas, *fine-tuning, c*omparam-se plataformas e medem-se capacidades. O debate absorve energia considerável em questões de infraestrutura computacional e escolha de fornecedores. Enquanto isso, o gargalo real permanece anterior, menos sofisticado e mais vinculado a um trabalho árduo e tedioso: **a qualidade dos dados que alimentam qualquer sistema.**

O princípio é antigo na computação. "Garbage in, garbage out" — dados ruins produzem resultados ruins — aparece na literatura técnica desde os primórdios do processamento eletrônico. A novidade é que modelos de linguagem ampliaram tanto o potencial quanto o risco dessa equação. Um sistema treinado em corpus jurídico mal estruturado erra com fluência, confiança e aparência de autoridade. A mesma capacidade generativa que torna esses modelos úteis os torna perigosos quando a base documental é deficiente.

A pesquisa recente sobre inteligência artificial aplicada ao direito confirma essa hierarquia de problemas. Estudos que tentaram criar bases de teste padronizadas para avaliar modelos jurídicos esbarraram sempre no mesmo obstáculo de que não faltam algoritmos sofisticados, mas faltam

documentos organizados.[1] As anotações são inconsistentes. As classificações variam entre tribunais. O que um sistema chama de "direito civil" outro registra como "obrigações". Mesmo os melhores modelos alcançam resultados modestos quando confrontados com a bagunça real dos acervos jurídicos. Ao contrário do que o marketing dos testes de performance fala, o cenário real enfrenta as dificuldades de dispersão dos acervos, documentos muito grandes que prejudicam a janela de contexto e linguagem complexa com terminologia própria, que afetam a compreensão dos modelos.[2]

O problema começa antes de qualquer processamento. Documentos jurídicos são, em tese, públicos. Na prática, raramente estão acessíveis de forma utilizável. Encontram-se dispersos entre tribunais, bases comerciais, sistemas internos de cada instituição. Quando disponíveis, apresentam problemas elementares como digitalização de má qualidade que transforma texto em imagem ilegível para sistemas automatizados, formatação que varia de um órgão para outro, informações cadastrais ausentes ou preenchidas de forma inconsistente.[3] Um estudo recente de pesquisadores do MIT e Stanford informou que os três desafios centrais da inteligência artificial jurídica não são de modelagem, mas de organização, classificação e verificação de dados.[4] A sofisticação do algoritmo é secundária diante dessas barreiras.

Para o jurista brasileiro, essa discussão é bastante concreta. O país realizou, nas últimas duas décadas, esforço significativo de digitalização judicial. O DataJud centraliza hoje mais de 280 milhões de processos, com 98,9% dos novos casos tramitando eletronicamente. A infraestrutura

1. GUHA, Neel et *al*. **LegalBench: a collaboratively built benchmark for measuring legal reasoning in large language models**. arXiv preprint *arXiv:2308.11462*, 2024. Disponível em: https://arxiv.org/abs/2308.11462. Acesso em: 20 jan. 2026.
2. NIKLAUS, Joel et *al*. LEXTREME: a multi-lingual and multi-task benchmark for the legal domain. In: **FINDINGS OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: EMNLP 2023**. Anais [...]. Singapore: Association for Computational Linguistics, 2023. p. 7508-7538. Disponível em: https://arxiv.org/abs/2301.13126. Acesso em: 20 jan. 2026.
3. ARIAI, FARID; MACKENZIE, JOEL; DEMARTINI, GIANLUCA. **Natural language processing for the legal domain: a survey of tasks, datasets, models and challenges**. preprint *arXiv:2410.21306*, 2024. Disponível em: https://arxiv.org/abs/2410.21306. Acesso em: 20 jan. 2026.
4. SANTOSH, T.Y.S.S. et *al*. **Tasks and roles in legal AI: data curation, annotation, and verification**. arXiv preprint *arXiv:2504.01349*, abr. 2025. Disponível em: https://arxiv.org/abs/2504.01349. Acesso em: 20 jan. 2026.