UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS QUIXADÁ

CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

RUBENS CAUAN FIGUEREDO DE CASTRO

O USO DA GERAÇÃO AUMENTADA POR RECUPERAÇÃO NA VALIDAÇÃO DA EXTRAÇÃO DE ARGUMENTOS DE DOCUMENTOS JURÍDICOS

QUIXADÁ

2025

RUBENS CAUAN FIGUEREDO DE CASTRO

O USO DA GERAÇÃO AUMENTADA POR RECUPERAÇÃO NA VALIDAÇÃO DA EXTRAÇÃO DE ARGUMENTOS DE DOCUMENTOS JURÍDICOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Regis Pires Magalhães.

QUIXADÁ

2025

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C353u Castro, Rubens Cauan Figueredo de.
O uso da geração aumentada por recuperação na validação da extração de argumentos de documentos jurídicos / Rubens Cauan Figueredo de Castro. – 2025.
59 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Sistemas de Informação, Quixadá, 2025.
Orientação: Prof. Dr. Regis Pires Magalhães.

1. Modelos de linguagem. 2. RAG. 3. Validação. 4. Documentos juridicos. 5. Similaridade . I. Título.

CDD 005

RUBENS CAUAN FIGUEREDO DE CASTRO

O USO DA GERAÇÃO AUMENTADA POR RECUPERAÇÃO NA VALIDAÇÃO DA EXTRAÇÃO DE ARGUMENTOS DE DOCUMENTOS JURÍDICOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação do Campus Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas de Informação.

Aprovada em: 29/07/2025.

BANCA EXAMINADORA

Prof. Dr. Regis Pires Magalhães (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Luis Gustavo Coutinho do Rêgo
IFCE

Prof. Dr. Alexandre Antônio Bruno da Silva
UECE

Profa. Dra. Lívia Almada Cruz
UFC

À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Kailane, seu amor e presença significou segurança e certeza de que não estou sozinho nessa caminhada.

AGRADECIMENTOS

A Deus, pela dádiva da vida, pelas oportunidades e pelos ensinamentos que me guiaram ao longo desta jornada.

À minha mãe e namorada, pelo amor incondicional, pelo suporte incansável e por todo o esforço e dedicação que tornaram possível a minha melhoria como ser humano. À minha família, pelo apoio constante e pela força que me deram ao longo dos anos de universidade.

Ao Prof. Dr. Régis Pires Magalhães, por suas valiosas orientações e direcionamentos fundamentais para a realização deste trabalho, juntamente com o Prof. Dr. Luis Gustavo Coutinho do Rêgo e Dr. Alexandre Antônio Bruno da Silva, que auxiliaram durante toda a execução desse trabalho.

Aos professores da banca examinadora, Dr. Alexandre Antônio Bruno da Silva, Dr. Luis Gustavo Coutinho do Rêgo e Dra. Lívia Almada Cruz , pela generosidade do tempo dedicado, pelas contribuições enriquecedoras e pelas sugestões que aprimoraram este trabalho.

"Uma lição sem dor é inútil. Isso porque não se pode ganhar algo sem sacrificar outra coisa em troca." (ARAKAWA, 2009)

RESUMO

Grandes Modelos de Linguagem (LLMs) representam uma tecnologia transformadora em diversas áreas, incluindo o setor jurídico. No entanto, a propensão desses modelos a gerar informações não factuais, conhecidas como "alucinações", impõe um risco significativo à sua aplicação em contextos que demandam alta precisão. Este trabalho propõe e avalia uma estratégia que adapta a arquitetura de Geração Aumentada por Recuperação (RAG) como um mecanismo para validar argumentos extraídos de documentos jurídicos. O objetivo principal é extrair argumentos de peças de defesa e, em seguida, validá-los ao confrontá-los com os trechos mais similares do próprio documento fonte, aumentando a confiabilidade do processo. A metodologia incluiu uma análise comparativa entre modelos de código aberto e proprietários, e o desenvolvimento de um sistema que fornece rastreabilidade explícita para cada argumento gerado. Os resultados indicam que a validação é crucial, e que a abordagem proposta aumenta a confiança na ferramenta, permitindo que até mesmo modelos gratuitos alcancem um desempenho robusto. Conclui-se que a integração eficaz de IA no direito depende de sistemas híbridos que aliem a automação à supervisão humana qualificada.

Palavras-chave: grandes modelos de linguagem; juridico; embeddings; similaridade; validação.

ABSTRACT

Large Language Models (LLMs) represent a transformative technology in several areas, including the legal sector. However, the propensity of these models to generate non-factual information, known as "hallucinations,"poses a significant risk to their application in contexts that demand high precision. This work proposes and evaluates a strategy that adapts the Retrieval-Augmented Generation (RAG) architecture as a mechanism to validate arguments extracted from legal documents. The main objective is to extract arguments from defense briefs and then validate them by cross-referencing them with the most similar excerpts from the source document itself, thereby increasing the process's reliability. The methodology included a comparative analysis between open-source and proprietary models, and the development of a system that provides explicit traceability for each generated argument. The results indicate that validation is crucial and that the proposed approach increases confidence in the tool, allowing even free models to achieve robust performance. It is concluded that the effective integration of AI in law depends on hybrid systems that combine automation with qualified human supervision.

Keywords: large language models; legal; embeddings; similarity; validation.

SUMÁRIO

1 INTRODUÇÃO

Os Grandes Modelos de Linguagem (LLMs) são sistemas de inteligência artificial projetados para entender e gerar texto em linguagem natural. Eles são treinados em grandes volumes de dados textuais, permitindo que desenvolvam capacidades avançadas de raciocínio, planejamento e interação, semelhantes às habilidades humanas. (Guo *et al.*, 2024). O crescimento exponencial da inteligência artificial nesta década traz consigo muitas facilidades em diversas áreas, como na saúde (Thirunavukarasu *et al.*, 2023), no mercado financeiro (Yang *et al.*, 2023) e no setor jurídico (Fabre, 2024).

Na área da saúde, os LLMs têm auxiliado na análise de prontuários médicos, identificação de padrões em diagnósticos e até no suporte ao paciente, como no caso de sistemas de triagem automatizada (Thirunavukarasu *et al.*, 2023). Já no mercado financeiro, modelos avançados são utilizados para prever tendências de mercado, elaborar relatórios automatizados e até detectar fraudes em tempo real (Yang *et al.*, 2023). No setor jurídico, os LLMs têm revolucionado o processo de análise documental, permitindo que grandes volumes de contratos e processos sejam examinados em minutos, otimizando o tempo de advogados e juízes (Fabre, 2024).

Por mais que os LLMs se mostrem eficazes e na maioria das vezes forneçam respostas adequadas, eles apresentam limitações importantes que tornam seu uso em contextos específicos inviável, muito por conta da falta de acesso a dados pessoais e privados na hora do seu treinamento (Xiao *et al.*, 2024). Além de tudo, quando um modelo não tem informações sobre determinado contexto, ele pode produzir informações incorretas, conhecidas como "alucinações" (Bruno *et al.*, 2023), o que pode comprometer a veracidade das respostas, especialmente em áreas críticas e que necessitam de informações rigorosas, como na área jurídica.

Nesse contexto, a área do Direito tem se destacado dentre as de maior potencial de desenvolvimento e uso de IA, tendo em vista a necessidade de lidar com milhares de dados e documentos relacionados à legislação, normas setoriais, processos judiciais, contratos e diversos outros elementos jurídicos, assim como com a grande quantidade de atos e atividades voltadas aos serviços jurídicos, tanto no setor público quanto no privado (Coelho, 2024). Nesse âmbito, as alucinações devem ser evitadas ao máximo, tendo em vista que esses documentos devem ter veracidade total. Sendo assim, para melhorar a geração de texto e evitar erros, devem ser realizadas técnicas que possibilitem a melhoria da eficiência, como engenharia de prompt (Vatsal; Dubey, 2024), ajuste fino do modelo (Han *et al.*, 2024) e Geração Aumentada por Recuperação

(RAG) (Lewis *et al.*, 2021), ou até mesmo de forma mais simples, fornecer ao usuário a garantia que a informação extraída esteja presente no texto.

Visto isso, é necessário analisar as desvantagens de cada técnica nesse contexto de aplicação. Embora o ajuste fino seja extremamente eficaz para criar e adaptar novos modelos, precisamos pensar que a legislação é flexível e pode sofrer pequenas alterações frequentemente, enfraquecendo fortemente essa abordagem, que necessita de um novo treinamento do modelo quando são adicionados novos dados (Barakat; Huang, 2023). Em contraparte, a técnica de *RAG* é usada principalmente para trabalhar com modelos que necessitam de informações atualizadas, buscando aumentar a eficiência dos modelos de linguagem, fornecendo um banco de dados vetorial para consulta de dados semelhantes. Pode ser uma opção mais viável, visto que não precisaremos treinar um modelo novo para trabalhar com os dados adicionados (Wang *et al.*, 2024), aumentando significativamente a eficiência e o desempenho da geração de texto em diversos contextos, especialmente no jurídico, que usaremos como exemplo no presente trabalho.

No entanto, a eficácia de qualquer sistema RAG depende fundamentalmente da qualidade do seu componente de recuperação. Um contexto mal recuperado levará a uma resposta final de baixa qualidade. Portanto, este trabalho foca em aplicar e avaliar o pilar central da abordagem RAG: o processo de recuperação e verificação. Investigaremos como o uso de busca por similaridade, para confrontar as informações geradas pelo LLM com os trechos do documento original, pode ser usado como uma estratégia robusta para garantir a fidelidade da extração. Dessa forma, propomos um sistema que não apenas extrai argumentos, mas que também fornece ao usuário um mecanismo de validação explícito, aumentando a confiabilidade e a transparência do processo.

1.1 Objetivos

Esta seção é referente ao objetivo geral e aos objetivos específicos do presente trabalho.

1.1.1 Objetivo Geral

Avaliar o uso de grandes modelos de linguagem (LLMs) no setor jurídico, por meio da aplicação da técnica de Recuperação Aumentada por Geração (RAG), utilizando bancos de dados vetoriais como estratégia para aprimorar a confiabilidade e a precisão na extração de

argumentos em documentos de defesa da jurisdição brasileira.

1.1.2 Objetivos Específicos

- Avaliar como modelos de linguagem variados se comportam na tarefa de extração de argumentos jurídicos.
- Criar um sistema que consiga extrair argumentos e forneça os trechos relacionados ao argumento no documento original, facilitando a revisão.
- Validar a implementação experimental por meio de métricas de desempenho quantitativas.
- Validar a implementação experimental por meio da avaliação de um especialista, quantitativamente.

1.2 Estrutura do trabalho

Os próximos capítulos estão organizados da seguinte maneira: no Capítulo 2 será apresentada a fundamentação teórica, onde se faz uma conceituação sobre a estruturação do auto de infração e das defesas administrativas; em seguida será abordado sobre grandes modelos de linguagem (LLMs) e o conceito de alucinações; em seguida, é apresentada a técnica de recuperação aumentada por geração (RAG), conceituando as estruturas fundamentais para seu funcionamento. Além disso, no final do capítulo são apresentadas as principais métricas de avaliação para técnicas de processamento de linguagem natural. No Capítulo 3 serão apresentados os trabalhos relacionados, com descrição e comparação de projetos e pesquisas que possuem aspectos similares com os propostos neste trabalho. No Capítulo 4 serão apresentados os procedimentos metodológicos, onde se tem a descrição de atividades relacionadas ao desenvolvimento e validação do modelo de avaliação proposto; o Capítulo 5 descreverá os resultados obtidos durante a execução dos passos metodológicos; por fim, no capítulo 6 será discutido a conclusão e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados os principais conceitos necessários para o desenvolvimento deste trabalho. Inicialmente, será abordado o conceito e a estruturação do auto de infração e das defesas administrativas.Logo após, será abordado o conceito de Grandes Modelos de Linguagem (LLMs), suas características e estrutura de funcionamento com *Transformers*. Em seguida, serão contextualizadas as alucinações, um dos maiores desafios enfrentados por esses modelos atualmente, destacando as principais estratégias de mitigar esse problema. Posteriormente, será discutida a recuperação aumentada por geração (RAG), abordando os principais componentes da sua estrutura, como *chunking*, *embedding* e banco de dados vetoriais. Por fim, serão exploradas as métricas de avaliação que podem ser aplicadas para comparação de similaridade de textos gerados.

2.1 Auto de infração

O auto de infração é um instrumento administrativo utilizado por autoridades fiscalizadoras, como os Auditores-Fiscais do Trabalho, para formalizar a constatação de uma infração à legislação vigente. No âmbito trabalhista, trata-se de um documento oficial por meio do qual o Auditor-Fiscal registra uma violação às normas previstas na Consolidação das Leis do Trabalho (CLT), convenções internacionais ratificadas pelo Brasil, decretos, portarias ou outras normas infralegais aplicáveis às relações laborais. O auto representa o início de um processo administrativo sancionador, podendo resultar na imposição de penalidades à empresa autuada, como a aplicação de multas.

Em termos estruturais, o auto de infração apresenta a identificação do empregador autuado, o local e a data da fiscalização, a descrição circunstanciada dos fatos que motivaram a autuação, a capitulação legal da infração (ou seja, o dispositivo legal infringido), a base legal para a penalidade e os elementos de convicção que fundamentam o ato administrativo. Esse documento é lavrado e assinado por um Auditor-Fiscal do Trabalho, que atua como agente público investido de fé pública e competência legal para tal atividade.

2.2 Defesa Administrativa

A defesa administrativa apresentada contra um auto de infração é um instrumento jurídico por meio do qual o autuado exerce seu direito constitucional ao contraditório e à ampla

defesa. Trata-se de uma manifestação formal, redigida por escrito, cujo objetivo é contestar os fatos e fundamentos jurídicos que embasaram a lavratura do auto de infração, buscando sua anulação, modificação ou mitigação.

Em regra, a defesa deve iniciar-se com o correto endereçamento à autoridade administrativa competente, identificando-se o número do processo, do auto de infração e os dados da empresa autuada. Em seguida, é fundamental que a peça registre a sua tempestividade, ou seja, demonstre que foi apresentada dentro do prazo legal com base nas normas administrativas aplicáveis.

Na sequência, costuma-se apresentar uma breve síntese do conteúdo do auto de infração, com a exposição objetiva dos fatos descritos pelo Auditor-Fiscal do Trabalho, bem como a identificação da norma supostamente violada. A defesa então passa à exposição da realidade fática, na qual descreve como a empresa atua em relação ao tema da autuação, por exemplo, o método adotado para controle de jornada, a estrutura organizacional, ou as circunstâncias específicas que explicam a conduta fiscalizada.

A fundamentação jurídica constitui o núcleo da defesa e pode abordar dois aspectos principais: questões formais e de mérito. Nas questões formais, argumenta-se, por exemplo, que o auto foi lavrado fora do local da inspeção ou enviado fora do prazo legal, o que pode acarretar a sua nulidade, conforme disposto no art. 629 da Consolidação das Leis do Trabalho (CLT). Já no mérito, são desenvolvidas teses como a inexistência de obrigação legal expressa que imponha determinado comportamento, bem como a violação aos princípios da razoabilidade, proporcionalidade e legalidade, fundamentos estes consagrados na Constituição Federal e na Lei nº 9.784/1999, que regula o processo administrativo federal.

É comum que a defesa seja instruída com documentos comprobatórios, como folhas de ponto, contratos, registros internos e declarações, os quais visam reforçar a veracidade dos fatos alegados. Além disso, a jurisprudência dos Tribunais Regionais e do Tribunal Superior do Trabalho pode ser citada como reforço argumentativo para demonstrar a aderência da tese defensiva ao entendimento consolidado do Judiciário. Por fim, são formulados os pedidos, que geralmente incluem a declaração de nulidade do auto de infração. Subsidiariamente, a defesa pode pleitear a conversão da penalidade em advertência, a aplicação de multa em seu valor mínimo ou a concessão de prazo para regularização da suposta infração. A peça se encerra com a qualificação e assinatura dos advogados legalmente constituídos para representar a empresa no processo administrativo.

Em suma, uma defesa administrativa eficaz deve aliar precisão jurídica, clareza na exposição dos fatos e uso criterioso das normas legais e da jurisprudência, de modo a assegurar a plena proteção dos direitos do autuado perante a Administração Pública.

2.3 Grandes Modelos de Linguagem

Grandes modelos de linguagem (LLMs) são modelos computacionais que têm a capacidade de compreender, processar e gerar linguagem natural. Eles representam uma evolução dos modelos de linguagem tradicionais, diferenciando-se pelo grande número de parâmetros e capacidade de aprendizado baseado no volume de dados (Chang *et al.*, 2024).

Inicialmente, os modelos de linguagem visavam principalmente a capacidade de gerar e modelar dados de texto. Por outro lado, modelos mais recentes, como o GPT-4, se concentram basicamente na resolução de tarefas complexas, como raciocínio lógico, tomada de decisão e compreensão textual (Zhao *et al.*, 2024b).

Com o avanço da tecnologia, diversos modelos são criados e melhorados constantemente, cada um com uma arquitetura e objetivo específico. Modelos como o GPT-4 exemplificam a evolução na variedade e sofisticação dos *LLMs*, usando bilhões de parâmetros e técnicas de treinamento (Zheng *et al.*, 2024). Estruturando-se principalmente numa arquitetura de *transformers*, o modelo utiliza mecanismos de atenção (Guo *et al.*, 2022) para processar e relacionar diferentes partes de um texto, permitindo uma compreensão contextual mais profunda.

Embora esses modelos tenham alcançado avanços notáveis, eles ainda apresentam desafios significativos. Um dos principais problemas enfrentados pelos *LLMs* é a confiabilidade das informações geradas, uma vez que esses modelos dependem de padrões estatísticos na incorporação de palavras. Ao inverso de verdadeiros processos cognitivos, os modelos podem produzir respostas imprecisas ou até mesmo completamente fictícias (Huang *et al.*, 2025).

2.3.1 Transformers

O *Transformer* é definido como um modelo de arquitetura para transdução[1] sequencial que depende inteiramente de mecanismos de atenção, eliminando a necessidade de camadas recorrentes ou convolucionais (Vaswani *et al.*, 2023). Essa abordagem permite que o *Transformer* capture dependências globais em sequências de entrada e saída, facilitando uma maior paralelização durante o treinamento, resultando em tempos de treinamento significativamente

[1] Transformar uma sequência de entrada em uma sequência de saída de forma específica.

mais curtos e desempenho superior em diversas tarefas, especialmente em tradução automática (Lin *et al.*, 2022).

Com o avanço dos grandes modelos de linguagem no campo do processamento de linguagem natural (PLN), os *Transformers* desempenham um papel cada vez mais crucial no desenvolvimento dessas ferramentas, resultando em melhorias significativas de desempenho (Fedus *et al.*, 2022). Modelos como GPT-4, BERT e T5 exemplificam a aplicação dessa arquitetura, consolidando seu papel fundamental no desenvolvimento de modelos de linguagem modernos.

2.3.2 Mecanismos de atenção

Os mecanismos de atenção são técnicas que permitem que os modelos concentrem-se automaticamente em partes-chave dos dados de entrada, mostrando vantagens significativas em muitos campos, como processamento de linguagem natural (PLN), reconhecimento de imagens e áudios. (Lu, 2024). Esse conceito é a base do modelo *Transformer*, permitindo que ele processe texto paralelamente e identifique relações semânticas entre palavras, independentemente da distância entre elas (Vaswani *et al.*, 2023).

2.4 Alucinações em LLMs

O conceito de alucinação tem suas raízes nos campos da patologia e da psicologia, sendo definido como a percepção de uma entidade ou evento que está ausente na realidade (Platchias, 2013). No contexto dos Grandes Modelos de Linguagem (LLMs), alucinação refere-se à geração de respostas que não se baseiam em informações factuais ou precisas. Elas podem ocorrer quando o modelo produz texto que inclui detalhes, fatos ou alegações que são fictícios, enganosos ou totalmente fabricados, em vez de fornecer informações confiáveis e verdadeiras (Rawte *et al.*, 2023). Em cenários em que é exigida extrema fidelidade e precisão nas respostas, como no direito, isso pode ser prejudicial, visto que qualquer erro pode causar uma grande confusão futuramente (Dahl *et al.*, 2024).

Como os *LLMs* não possuem um entendimento semântico profundo, eles geram respostas com base em padrões linguísticos presentes nos dados utilizados para treinamento, o que pode levar à produção de informações incorretas ou enganosas (Orgad *et al.*, 2024). Para mitigar esse problema, diversas abordagens têm sido exploradas, incluindo o refinamento dos

dados de treinamento, o uso de técnicas de engenharia de prompt, geração aumentada por recuperação (RAG), e ajuste fino (Fine tuning) (Tonmoy *et al.*, 2024).

2.5 Geração aumentada por recuperação (RAG)

A técnica de Geração Aumentada por Recuperação (RAG), inicialmente proposta no artigo "*Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*"de Lewis *et al.* (2021), combina métodos de recuperação de informações com modelos de geração, para melhorar a precisão e a qualidade das respostas em tarefas que demandam informações externas e atualizadas. Essa abordagem visa mitigar uma das principais limitações dos *LLMs*: a dependência de informações armazenadas durante o treinamento, o que pode levar à geração de respostas não factuais e imprecisas, conhecidas como "alucinações"(Rawte *et al.*, 2023).

O funcionamento do *RAG* ocorre em duas etapas principais. Primeiro, um módulo de recuperação busca informações relevantes em uma base de conhecimento externa, como documentos, bancos de dados ou até mesmo a web. Em seguida, um modelo generativo usa o conteúdo recuperado para gerar respostas mais contextualizadas e factualmente corretas (Zhao *et al.*, 2024a) como descrito na Figura 1.

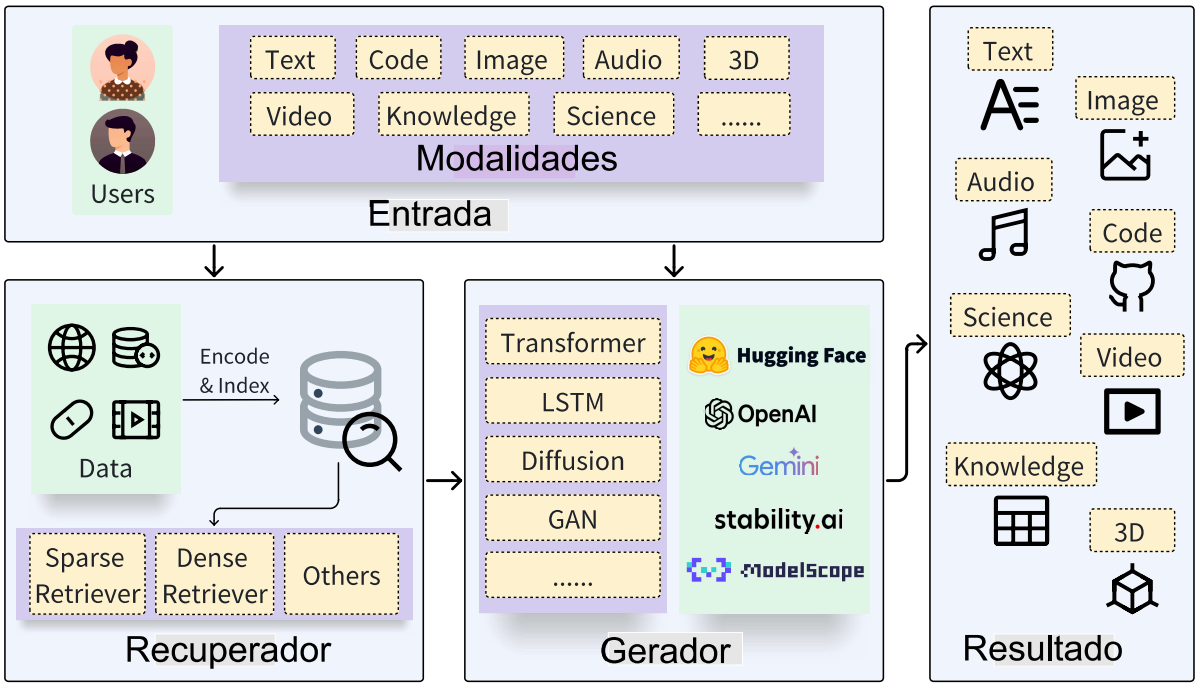
No contexto do *RAG*, é crucial recuperar eficientemente documentos relevantes da fonte de dados. Diversos fatores entram em discussão nesse processo, incluindo a escolha da fonte de dados para recuperação, a técnica de *chunking* utilizada para dividir os dados de maneira otimizada, o pré-processamento adequado dos dados recuperados e a seleção do modelo de *embedding* que será utilizado para representar semanticamente a informação (Gao *et al.*, 2024). O objetivo é equilibrar essas técnicas de maneira a resolver o problema de forma eficaz e eficiente.

2.5.1 Chunking

A fragmentação de informações ou *chunking* é uma etapa fundamental na Geração Aumentada por Recuperação (RAG) (Yepes *et al.*, 2024). O *chunking* envolve a divisão de textos ou documentos grandes em segmentos menores de tamanho fixo, ou adaptativo. Isso permite que o recuperador se concentre em unidades menores por vez, facilitando o processamento e a análise do texto (Kshirsagar, 2024).

Existem diferentes métodos de *chunking*, cada um com características próprias (Qu *et al.*, 2024):

Figura 1 – Uma arquitetura RAG genérica.



Fonte: Adaptado de Zhao *et al.* (2024a).

– Tamanho fixo: divide o texto em blocos de um número pré-determinado de palavras ou tokens.
– Baseado em sentenças: utiliza pontuação e estrutura gramatical para segmentação.
– Baseado em parágrafos: mantém a divisão natural do texto.
– Com janelas deslizantes: sobrepõe trechos para preservar contexto entre *chunks*.
– Semântico: Utiliza modelos de *embedding* para realizar a divisão do texto, considerando não só tamanha dos trechos, mas também sua semântica.
– Baseado em agentes de IA: Utiliza modelos de IA para definir dinamicamente os melhores pontos de segmentação, considerando a estrutura e o significado do texto.

A escolha do método adequado de *chunking* impacta diretamente a qualidade da recuperação, pois *chunks* muito pequenos podem perder contexto relevante, enquanto *chunks* muito grandes podem introduzir informações irrelevantes na geração da resposta (Kshirsagar, 2024).

2.5.2 Embedding

Embeddings são representações vetoriais de alta dimensionalidade que mapeiam dados não estruturados, como palavras, frases ou documentos, para um espaço contínuo de menor dimensão. Essa transformação é essencial para o armazenamento em banco de dados vetoriais(Rau *et al.*, 2024). Essa conversão preserva as semelhanças semânticas e sintáticas entre os dados originais, permitindo que relações e contextos complexos sejam capturados de maneira eficiente. Em modelos de aprendizado de máquina, especialmente em Processamento de Linguagem Natural (PLN), os *embeddings* facilitam o processamento e a extração de informações, tornando a modelagem mais eficiente em tarefas como classificação, recuperação de informações e geração de texto. (Gao *et al.*, 2024)

2.5.3 Banco de dados vetorial

Os bancos de dados vetoriais são sistemas de gerenciamento de dados especializados em armazenar e recuperar informações representadas como vetores de alta dimensão. Esses vetores são representações matemáticas de atributos extraídos de diferentes tipos de dados, como texto, imagens, áudio e vídeo. Cada vetor pode ter várias dimensões, variando de dezenas a milhares, dependendo da granularidade dos dados e da complexidade do domínio (Pan *et al.*, 2024). Para gerar esses vetores, são aplicados modelos de *embeddings*, que transformam dados brutos em representações numéricas, utilizando métodos como aprendizado de máquina, *embeddings* de palavras e algoritmos de extração de características. Esses bancos de dados possuem características que possibilitam comparar vetores por similaridade, auxiliando especialmente em tarefas que exigem buscas por terchos similares e análise de grandes volumes de dados não estruturados (Han *et al.*, 2023).

2.6 Métricas de Avaliação

A avaliação da similaridade dos textos é essencial para medir a qualidade das respostas geradas pelo modelo. Diferentes métricas podem ser utilizadas para comparar a similaridade entre a resposta gerada e a referência esperada, bem como para avaliar a qualidade semântica e estrutural do texto. As principais métricas utilizadas incluem Similaridade do Cosseno, BERTScore incluindo métricas calculados por ela como *precision, recall e f1-score*, completude e corretude.

2.6.1 Similaridade do Cosseno

A Similaridade do Cosseno é uma métrica utilizada para medir a proximidade entre dois vetores de texto, sendo amplamente usada na recuperação de informações e na avaliação de *RAG*. Ao representar textos como vetores em um espaço multidimensional, essa métrica calcula o cosseno do ângulo entre eles, indicando a semelhança semântica. Quanto mais próximo de 1 (um) for o valor da similaridade do cosseno, maior é a similaridade entre os textos (Juvekar; Purwar, 2024).

$$\cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} \tag{2.1}$$

onde:

- $A \cdot B$ é o produto escalar dos vetores;

- $\|A\|$ e $\|B\|$ são as normas dos vetores.

2.6.2 BERTScore

O *BERTScore* utiliza representações contextualizadas de palavras geradas por modelos de linguagem baseados em *transformers*, como o *BERT*. Essa métrica calcula similaridades entre *embeddings* das palavras, permitindo uma avaliação mais semântica das respostas geradas. Ao contrário de *BLEU* e *ROUGE*, o *BERTScore* não depende diretamente da coincidência exata de palavras, tornando-se mais robusto para avaliar significados similares (Zhang *et al.*, 2020).

$$BERTScore = \frac{1}{N}\sum_{i=1}^{N} \max_{j} \cos(\mathbf{x}_i, \mathbf{y}_j) \tag{2.2}$$

onde:

- $\mathbf{x}_i$ e $\mathbf{y}_j$ são os vetores de *embeddings* da palavra i do texto gerado e da palavra j da referência;

- $\cos(\mathbf{x}_i, \mathbf{y}_j)$ é a similaridade do cosseno entre os *embeddings* das palavras.

- Podendo variar de 0 a 1, quanto mais próximo de 1 mais similar.

2.6.2.1 Precision

Para calcular a precisão, cada token na frase candidata é pareado com o token mais similar na frase de referência. A pontuação de precisão é a média dessas pontuações de

similaridade máxima. A precisão, portanto, avalia se os tokens na frase candidata são relevantes em relação à frase de referência (Zhang *et al.*, 2020).

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \cos(\hat{x}_j, x_i) \tag{2.3}$$

onde:

- $\hat{x} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_m\}$ são os embeddings dos tokens da frase candidata;
- $x = \{x_1, x_2, \ldots, x_n\}$ são os embeddings dos tokens da frase de referência;
- $\cos(\hat{x}_j, x_i)$ representa a similaridade de cosseno entre os tokens;
- $|\hat{x}|$ é o número total de tokens na frase candidata.

2.6.2.2 Recall

Para cada token na frase de referência, o recall do BERTSCORE identifica o token mais similar (através da similaridade de cosseno) na frase candidata. A pontuação de recall é a média dessas pontuações de similaridade máxima. Essencialmente, o recall mede o quão bem cada palavra na frase de referência está representada na frase candidata (Zhang *et al.*, 2020).

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \cos(x_i, \hat{x}_j) \tag{2.4}$$

onde: - $x = \{x_1, x_2, \ldots, x_n\}$ são os embeddings dos tokens da frase de referência;

- $\hat{x} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_m\}$ são os embeddings dos tokens da frase candidata;
- $\cos(x_i, \hat{x}_j)$ representa a similaridade de cosseno entre os tokens;
- $|x|$ é o número total de tokens na frase de referência.

2.6.2.3 F1-Score

O F1-score é a média harmônica da precisão e do recall, combinando as duas métricas para fornecer uma pontuação geral e equilibrada, sendo uma das medidas mais confiáveis em diversas configurações de avaliação de processamento de linguagem natural (Zhang *et al.*, 2020).

$$F_{BERT} = 2 \cdot \frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}} \tag{2.5}$$

onde: - P_{BERT} é a precisão calculada entre os tokens da candidata e os tokens mais similares da referência;

- R_{BERT} é o recall calculado entre os tokens da referência e os tokens mais similares da candidata;

- O F1-score pondera os dois para obter uma medida equilibrada da qualidade da similaridade semântica.

2.6.3 Completude

Esta métrica é relevante para a qualidade das respostas geradas. A pontuação é uma pontuação média das respostas em todos os prompts do seu conjunto de dados. Completude significa responder e resolver todos os aspectos das perguntas. Quanto maior a pontuação, mais completas são, em média, as respostas geradas. Quanto menor a pontuação, menos completas são, em média, as respostas geradas.

$$\text{Completude} = \frac{1}{N}\sum_{i=1}^{N} c_i \tag{2.6}$$

onde: - N é o número total de prompts avaliados;

- $c_i \in [0,1]$ representa a nota de completude atribuída à resposta do i-ésimo prompt;

- O valor de c_i pode ser binário (0 para incompleto, 1 para completo) ou contínuo, baseado em uma escala (por exemplo, de 0 a 1).

2.6.4 Corretude

Esta métrica é relevante para a qualidade das respostas geradas. A pontuação é uma pontuação média das respostas em todos os prompts do seu conjunto de dados. Correção significa responder às perguntas com precisão. Quanto maior a pontuação, mais corretas são, em média, as respostas geradas. Quanto menor a pontuação, menos corretas são, em média, as respostas geradas.

$$\text{Corretude} = \frac{1}{N}\sum_{i=1}^{N} r_i \tag{2.7}$$

onde: - N é o número total de prompts avaliados;

- $r_i \in [0,1]$ representa a nota de corretude atribuída à resposta do i-ésimo prompt;

- O valor de r_i também pode ser binário (0 para incorreto, 1 para correto) ou contínuo, com base em uma escala de avaliação.

3 TRABALHOS RELACIONADOS

Neste capítulo, serão apresentados alguns trabalhos relacionados, destacando as semelhanças e diferenças com o que foi desenvolvido neste trabalho.

3.1 Automatic Information Extraction From Employment Tribunal Judgements Using Large Language Models

Faria *et al.* (2024) realizaram um estudo abrangente sobre a aplicação do GPT-4 para a extração automática de informações de julgamentos do Tribunal de Emprego do Reino Unido (UKET). O objetivo era avaliar o desempenho do modelo na extração de informações cruciais para especialistas jurídicos e para o público.

A pesquisa foi estruturada em duas tarefas principais de extração. A primeira, mais geral, focou em extrair oito aspectos-chave dos casos:

- Fatos do caso
- Alegações feitas
- Referências a estatutos legais
- Referências a precedentes
- Resultado geral do caso
- Rótulos correspondentes (reclamante ganha, reclamante ganha parcialmente, reclamante perde e outro)
- Ordem detalhada e soluções
- Razões da decisão

A segunda tarefa foi mais focada, analisando os fatos, reivindicações e resultados extraídos para avaliar a viabilidade de desenvolver uma ferramenta de previsão de resultados de disputas trabalhistas. Para isso, os autores utilizaram o modelo GPT-4 sobre um subconjunto de 260 casos do Cambridge Law Corpus, selecionados por amostragem estratificada. A extração foi guiada por um processo iterativo de engenharia de prompt e os resultados foram verificados manualmente por especialistas jurídicos para garantir a acurácia.

A avaliação quantitativa demonstrou uma alta precisão do GPT-4. O modelo alcançou uma acurácia perfeita (100%) na extração de referências a estatutos legais e precedentes. Para outros aspectos como reivindicações, resultados gerais, resultados detalhados e razões da decisão, a acurácia foi quase perfeita, superando 0,98. A extração de fatos e a classificação do resultado

em um dos quatro rótulos, embora mais desafiadoras, ainda obtiveram uma alta acurácia, superior a 0,9. Qualitativamente, os autores notaram que o modelo apresentou algumas inconsistências, especialmente na aplicação dos rótulos "outro"e "reclamante ganha parcialmente"em cenários processuais complexos.

O estudo conclui que modelos como o GPT-4 podem alcançar alta precisão na extração de informações jurídicas, destacando seu potencial para revolucionar como a informação legal é processada e utilizada. Além disso, a análise revelou que aproximadamente 47,7% dos casos extraídos seriam adequados para a criação de um conjunto de dados para uma tarefa de previsão de resultados.

3.2 Large Language Models for Judicial Entity Extraction: A Comparative Study

O trabalho de Hussain e Thomas (2024) propõe uma abordagem para iniciar o processo de extração de entidades legais com envolvimento mínimo de especialistas de domínio. O trabalho visa resolver o desafio de criar grandes conjuntos de dados anotados, um processo tradicionalmente caro e demorado que requer o conhecimento de especialistas.

A metodologia central combina um modelo semântico chamado SEMLEG v2 com um grande modelo de linguagem, o GPT-4. O modelo semântico foi desenvolvido para formalizar as regras de manutenção legal, unificando conceitos dos domínios jurídico e de manutenção industrial, servindo como base para guiar a extração. O conhecimento dos especialistas foi encapsulado nas definições dos conceitos do modelo semântico, que foram então fornecidas ao GPT-4 por engenharia de prompt.

Para a avaliação, utilizaram um conjunto de dados existente de sentenças da lei de trânsito de Luxemburgo. O trabalho avalia a precisão, diferenciando entre correspondência perfeita e correspondência parcial, sendo a parcial criada para validar extrações que identificam corretamente o conceito, mas com limites de texto ligeiramente diferentes dos anotados por especialistas, utilizando a distância de Levenshtein normalizada para medir a variação.

A abordagem com modelos de linguagem reduziu drasticamente a necessidade de intervenção humana. Os autores identificaram como principal limitação do modelo a dificuldade em delimitar com precisão as fronteiras das entidades, resultando em mais correspondências parciais. O trabalho conclui que a flexibilidade dos modelos, mesmo com uma precisão de fronteira menor, oferece um método prático e de baixo custo para iniciar o processo de extração de informações legais.

3.3 Leveraging Large Language Models for Relevance Judgments in Legal Case Retrieval

Ma *et al.* (2025) abordam o desafio de realizar julgamentos de relevância na recuperação de casos jurídicos, uma tarefa que exige conhecimento de domínio e a análise de textos longos e repletos de detalhes. O trabalho propõe um método para alavancar Grandes modelos de linguagem para automatizar este processo com mínima supervisão de especialistas, gerando dados anotados de alta qualidade e de forma interpretável.

Os autores desenvolveram um fluxo de trabalho de quatro etapas que decompõe o processo de julgamento de relevância, imitando o raciocínio de especialistas humanos. O processo inicia com a análise de fatos materiais e fatos legais. As etapas incluem:

- Uma análise factual preliminar por especialistas
- Uma busca adaptativa por demonstrações
- A extração de Fatos
- a anotação da relevância dos fatos entre um par de casos

A avaliação, realizada no conjunto de dados de casos jurídicos chineses LeCaRD (Ma *et al.*, 2021), demonstrou que os julgamentos gerados pelo modelo possuem alta consistência com as anotações feitas por humanos, validada pela métrica Kappa de Cohen. O estudo também demonstrou a utilidade prática do método ao gerar um grande conjunto de dados sintético. Treinar modelos de recuperação de casos (como BERT e Longformer) com esses dados sintéticos resultou em melhorias de desempenho significativas. Os autores mostraram que é possível transferir a experiência de análise de caso do GPT-3.5 para modelos menores como Llama-2 e Qwen-2 por um processo de destilação de conhecimento, usando as anotações e os raciocínios gerados.

3.4 Extracting Legal Norm Analysis Categories from German Law Texts with Large Language Models

O trabalho de Bachinger *et al.* (2024) investiga o uso de grandes modelos de linguagem para automatizar a extração de categorias de análise de normas jurídicas em textos de leis alemãs. O estudo visa apoiar a digitalização de serviços públicos na Alemanha, um processo que depende da identificação de entidades como atores, ações e condições nos documentos legais, tarefa esta que atualmente é manual e de alto custo.

A pesquisa adota uma abordagem sistemática, começando com a seleção de cinco

modelos adequados como LeoLM e BLOOM-CLP German, a partir de uma lista inicial de 61 candidatos. Em seguida, os autores realizam uma extensa engenharia de prompt, testando inicialmente nove estruturas de saída diferentes e, por fim, criando cinco variantes de prompt com diferentes níveis de informação, como: descrições da tarefa, diretrizes de anotação e número de exemplos. Já que o modelo era solicitado para prever uma categoria por vez, foi desenvolvida uma estratégia de consolidação otimista e pessimista para fundir as múltiplas previsões de uma mesma sentença.

A avaliação focou nos dois modelos com melhores resultados, LeoLM e BLOOM-CLP German. O BLOOM-CLP German alcançou o maior F1-score (0,91), porém com uma baixa acurácia balanceada, indicando desempenho desigual entre as categorias. Por outro lado, o LeoLM, embora com um F1-score menor (0,82), apresentou uma acurácia balanceada significativamente maior, demonstrando uma capacidade de anotação mais consistente entre todas as categorias legais. O melhor desempenho do LeoLM foi obtido com o prompt que continha mais informações, incluindo três exemplos e diretrizes de anotação.A combinação do modelo LeoLM com o prompt mais informativo foi considerada a melhor para a tarefa.

3.5 Análise Comparativa

Para contextualizar o trabalho desenvolvido, é de suma importância comparar os trabalhos recentes que operam na criação e melhoria de sistemas que utilizem extraiam informações com modelos de linguagem de alguma forma. O Quadro 1 traz uma análise comparativa entre o presente estudo e os trabalhos relacionados. Para realizar essa comparação, serão usados cinco principais critérios de comparação:

- **Objetivo:** Este parâmetro descreve o propósito principal de cada trabalho.
- **Modelos de linguagem:** Representa os grandes modelos de linguagem usados no trabalho.
- **Estratégia de principal:** Define como os documentos serão passados para os modelos de linguagem.
- **Método de validação:** Indica a forma de validar as respostas geradas pelo modelo.
- **Embedding:** Refere-se ao modelo de *embeddings* usado para transformar o texto em representações vetoriais que possam ser utilizadas no banco vetorial.
- **Métricas de Avaliação:** Representam as ferramentas usadas para medir a eficácia do sistema.

Quadro 1 - Análise comparativa dos trabalhos relacionados com base em extração textual no domínio jurídico.

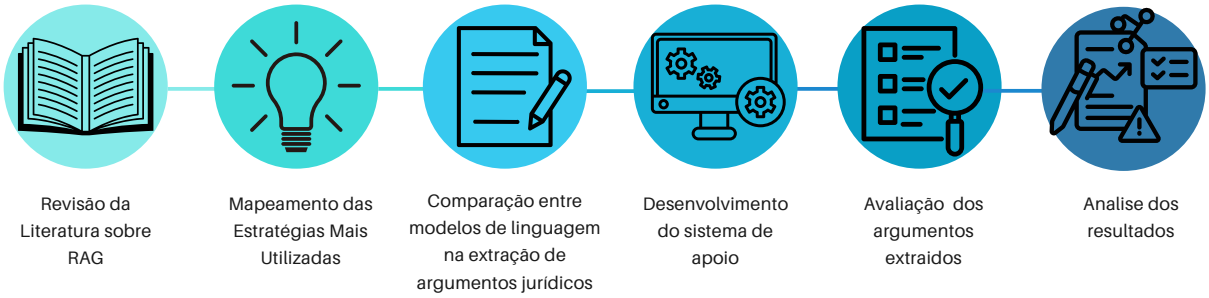
Trabalho	**Objetivo**	**Modelos de Linguagem**	**Estratégia Principal**	**Método de validação**	**Modelo de embedding**	**Métricas de Avaliação**
Automatic Information Extraction From Employment Tribunal Judgements Using LLMs (Faria *et al.*, 2024)	Extrair 8 aspectos-chave de julgamentos	GPT-4	Engenharia de prompt para extrair campos de informação de forma estruturada	Verificação manual da acurácia por especialistas, comparando a saída do modelo com o conteúdo do documento original.	Não possui	Acuracia
Large Language Models for Judicial Entity Extraction: A Comparative Study (Hussain; Thomas, 2024)	Extração de entidades legais com o mínimo de envolvimento de especialistas	GPT-4	Combinação de um modelo semântico para guiar o LLM com prompts do tipo one-shot	Comparação da extração com resultados previamente anotado, "correspondência parcial"para validar a correção semântica.	Não possui	F1-score
Leveraging Large Language Models for Relevance Judgments in Legal Case Retrieval (Ma *et al.*, 2025)	Automatizar julgamentos de relevância para a Recuperação de Casos Jurídicos.	GPT-3.5, Llama-2 e Qwen-2.	Análise, Busca de Demos, Extração de Fatos, Anotação de Fatos	Medição da consistência estatística entre os julgamentos do LLM e os de especialistas humanos.	Não possui	Kappa de Cohen e NDCG
Extracting Legal Norm Analysis Categories from German Law Texts with Large Language Models (Bachinger *et al.*, 2024)	Automatizar a extração de categorias de análise de normas jurídicas	LeoLM, BLOOM-CLP German, mT0, PolyLM, Falcon 7B.	Comparação de modelos e teste extensivo de 5 prompts com diferentes níveis de informação	Análise de erros detalhada para investigar	Não possui.	F1-score e acurácia balanceada
Este Trabalho	**Extrair corretamente argumentos de defesas juridicas**	**Modelos proprietários e públicos**	**Aplicação de *chunking* semantico para processamento do modelo**	**Recuperação de trechos do proprio documento similares com o argumento gerado**	**STJ IRIS**	**Similaridade do cosseno, BertScore, corretude e completude**

Fonte: Elaborado pelo autor

4 METODOLOGIA

Nesta seção serão apresentadas as etapas necessárias para a execução do trabalho, além de um planejamento de condução dessas tarefas, para que a solução presente neste trabalho possa ser modelada, desenvolvida e validada. A Figura 2 apresenta a sequência das etapas definidas para o projeto.

Figura 2 – Fluxos das atividades a serem executadas



Fonte: Elaborado pelo autor

4.1 Revisão da Literatura sobre RAG

Nesta primeira etapa, o objetivo é identificar e analisar artigos e estudos que envolvam extração textual de forma prática e teórica. Essa análise é fundamental para orientar a escolha de parâmetros e *frameworks* mais adequados ao contexto planejado. Considerando que a área de LLMs está em constante evolução, faz-se necessário o estudo das práticas consolidadas e emergentes, para assegurar que este trabalho esteja alinhado com os avanços mais recentes e contribua de maneira significativa para o progresso futuro.

Para esse estudo, será inicialmente realizada uma revisão da literatura atual, buscando artigos em fontes acadêmicas conhecidas, como, por exemplo: *ACM digital library*, *arXiv* e *Google scholar*. Essa prática possibilita identificar as principais abordagens discutidas no âmbito acadêmico. Além disso, será realizada uma pesquisa detalhada em repositórios de códigos, especialmente na plataforma *Github*, visando observar implementações e documentações estabelecidas pelos desenvolvedores.

Ao concluir esta etapa, teremos uma visão completa das práticas estabelecidas no uso de modelos de linguagem na extração de informações textuais, permitindo-nos tomar decisões baseadas na seleção dos modelos e na implementação do código-fonte para as próximas etapas da metodologia de trabalho.

4.2 Mapeamento das Estratégias Mais Utilizadas

Nesta etapa, realiza-se um levantamento das principais técnicas utilizadas na construção de sistemas baseados na recuperação textual, considerando abordagens amplamente adotadas em estudos e aplicações práticas. O objetivo é identificar e classificar estratégias, baseando-se no que os estudos consolidam como métodos eficazes de realizar extração de informações de textos complexos utilizando modelos de linguagem, sem realizar uma escolha definitiva. Esse mapeamento permitirá uma análise comparativa na próxima etapa.

4.2.1 Estratégias de Chunking

A segmentação do texto em partes menores (*chunks*) é essencial, visto que muitos modelos de linguagem não conseguem se portar bem com textos muito extensos. Algumas abordagens mapeadas incluem:

Chunking baseado em tamanho fixo de tokens: Esta abordagem segmenta o texto em partes de tamanho fixo, o que é simples e eficiente. Contudo, pode cortar informações importantes no meio de sentenças, prejudicando o contexto semântico.

Chunking baseado em sentenças ou parágrafos: Segmenta o texto em *chunks* maiores, como sentenças ou parágrafos. Essa abordagem preserva a estrutura lógica do documento, mas pode resultar em segmentos desbalanceados, especialmente em textos com sentenças ou parágrafos longos.

Chunking semântico: Esta estratégia consiste na utilização de modelos de *embedding* para a vetorização do texto, preservando seu tamanho até que seja identificada uma variação significativa entre os vetores gerados, essa variação é baseada em um limiar definido pelo usuário. Quando a diferença entre os vetores atinge ou ultrapassa esse limiar, entende-se que há uma mudança semântica no conteúdo, e o texto é segmentado nesse ponto.

Chunking recursivo (*Recursive Chunking*): Segmenta o texto de maneira hierárquica, reduzindo progressivamente os tamanhos dos *chunks* até atingir um limite ideal. Esta abordagem ajuda a manter a coerência do contexto, especialmente em textos com estrutura hierárquica, mas pode ser complexa de configurar.

Chunking baseado em agentes: Utiliza agentes especializados para determinar dinamicamente a melhor forma de segmentar o texto, considerando a semântica e a estrutura do documento. Essa técnica exige maior custo computacional devido ao alto processamento.

4.2.2 Técnicas de Embeddings

A escolha do modelo de *embeddings* impacta diretamente a recuperação semântica do sistema. As principais estratégias identificadas são:

Modelos gerais de embeddings (ex.: `text-embedding-ada-002`, `all-MiniLM-L6-v2`): Esses modelos são treinados em grandes corpora de textos diversos, tornando-os aplicáveis a uma variedade de domínios. Embora sejam eficientes em termos de desempenho, podem não capturar nuances específicas de determinados setores, como o jurídico.

Modelos especializados em textos jurídicos (ex.: `CaseLaw-BERT`, `Legal-BERT`): Esses modelos são otimizados para o vocabulário e a estrutura jurídica, oferecendo maior precisão em tarefas relacionadas a textos jurídicos. No entanto, tendem a ser mais caros e podem ter uma disponibilidade limitada, o que pode ser um desafio em algumas implementações.

4.3 Comparação entre modelos de linguagem na extração de argumentos jurídicos.

Os grandes modelos de linguagem atualmente já conseguem realizar a extração de argumentos sem o auxílio de técnicas externas, contudo, podem alucinar e trazer para a sua resposta final argumentos que não existem na defesa apresentada. Para avaliar esses modelos usaremos uma estratégia padrão para ser justo, seguindo o seguinte passo a passo:

- Um Prompt padrão que será usado em todos os modelos.
- Garantir que os modelos não contenham contexto anterior relativo a qualquer outro processo.
- Passar para eles o texto da defesa de forma plana, evitando que os modelos lidem com diferentes formatos de extração direta do PDF, o que poderia causar inconsistências.
- Validar os argumentos extraídos pelo modelo com os argumentos verdadeiros extraidos por um especialista, utilizando similaridade do cosseno com 4 *embeddings* diferentes.
- Avaliação por um especialista, qualitativamente.

4.4 Desenvolvimento do sistema de apoio

O objetivo desse passo metodológico é desenvolver um sistema que integre processamento de linguagem natural, análise semântica e armazenamento estruturado para:

1. Extrair automaticamente argumentos de defesa de documentos jurídicos em formato PDF utilizando modelos de linguagem (LLM) especializados em contexto legal brasileiro;

2. Validar a correspondência entre argumentos extraídos e o documento original por meio de:
   - Segmentação semântica do texto usando *embeddings* jurídicos especializados (BERT Legal Português)
   - Cálculo de similaridade cosseno entre argumentos e trechos do documento
   - Classificação automática por níveis de confiança (Validado/Precisa Revisão/Não Encontrado), baseado na similaridade do cosseno.
3. Fornecer rastreabilidade completa permitindo:
   - Pontuação de similaridade para cada correspondência
   - Adição manual de argumentos não identificados automaticamente
   - Validação humana com sistema de confirmação/rejeição
   - Visualização dos trechos específicos do documento que fundamentam cada argumento.
4. Manter histórico estruturado com:
   - Armazenamento persistente em banco de dados relacional
   - Metadados completos (modelo utilizado, versão do sistema, usuário, processo)
   - Interface de consulta e filtragem para análise posterior
   - Controle de versão, exibindo a versão do sistema que o argumento foi extraido.

Diferencial Técnico

O sistema irá combinar múltiplas abordagens tecnológicas:

- *Embeddings* jurídicos especializados: para compreensão contextual
- *Chunking* semântico: para segmentação inteligente do texto
- Validação por similaridade: para verificação de correspondência
- Interface interativa: para revisão e correção humana
- Armazenamento estruturado: para análise posterior

O objetivo é facilitar a revisão e análise de argumentos jurídicos ao automatizar a identificação e organização de elementos defensivos, reduzindo tempo de análise manual e aumentando a precisão na identificação de estratégias argumentativas em documentos legais.

4.5 Avaliação da extração dos argumentos obtidos pelo sistema

Nesta etapa serão realizados as avaliações dos argumentos, abordando métricas quantitativas e qualitativas.

4.5.1 Avaliação quantitativa

A avaliação quantitativa dos argumentos gerados pelos modelos será realizada utilizando um conjunto de métricas de desempenho. Essas métricas foram escolhidas para refletir tanto a qualidade da segmentação quanto a relevância das representações geradas para o contexto jurídico. As métricas utilizadas são:

- **Bert-score**: métrica de avaliação que aprimora a medição de similaridade de texto. Ao combinar valores de precisão e recall, ela torna a medição de similaridade de texto mais precisa e equilibrada.
- **Similaridade de Cosseno**: que calcula a similaridade angular entre os vetores de *embeddings* gerados para os textos, sendo útil para avaliar a proximidade semântica entre sentenças.

4.5.2 Avaliação qualitativa automática e por especialista

A avaliação qualitativa visa analisar, de forma subjetiva e interpretativa, a qualidade dos argumentos extraídos a partir das defesas fornecidas pelo usuário. Essa análise busca identificar aspectos que não são completamente capturados pelas métricas quantitativas, como a adequação jurídica, a coerência argumentativa e a precisão terminológica.

Essa analise será dividida em duas etapas, sendo elas uma avaliação com uma bilioteca de *LLM as judge* e avaliação de um especialista:

4.5.2.1 Avaliação de completude e corretude com biblioteca DeepEval

A biblioteca *DeepEval* [1] é uma biblioteca de código aberto, voltada para a avaliação de respostas de modelos de linguagem. Com ela, é possível criar métricas personalizadas conforme critérios descritivos definidos pelo usuário, a avaliação será feita por um modelo de linguagem escolhido pelo usuário. Com isso foram criados duas métricas para a avaliação:

- **Completude**: Mede se a resposta gerada pelo modelo contém todos os argumentos

[1] https://deepeval.com

relevantes que estão presentes na referência (gabarito). A métrica avalia a *cobertura* dos argumentos esperados, ignorando a presença de argumentos extras ou irrelevantes. O foco está na recuperação correta dos elementos essenciais da defesa.

- **Corretude**: Avalia se os argumentos fornecidos pelo modelo estão de fato corretos, ou seja, se estão presentes na referência e mantêm o mesmo sentido e contexto. Essa métrica penaliza informações inventadas, distorcidas ou alucinações, mesmo que a maioria dos argumentos esperados tenha sido coberta.

4.5.2.2 Avaliação qualitativa por especialista

Para essa etapa, será conduzida uma análise manual por um especialista da área jurídica, que avaliará as respostas considerando os seguintes critérios:

- **Adequação jurídica**: verifica se o conteúdo da resposta está juridicamente correto e alinhado com os fundamentos legais aplicáveis ao caso.
- **Coerência textual**: avalia se a resposta apresenta uma estrutura lógica e encadeada, sem contradições ou rupturas no fluxo argumentativo.
- **Clareza e objetividade**: Analisa se a resposta é redigida de forma clara, compreensível e direta, sem ambiguidade ou redundâncias desnecessárias.
- **Precisão terminológica**: verifica o uso adequado da terminologia jurídica, assegurando que termos técnicos são aplicados corretamente no contexto.

Com isso, espera-se complementar a análise quantitativa, proporcionando uma visão abrangente do desempenho dos sistemas no domínio jurídico, com foco tanto em métricas automáticas quanto na percepção especializada sobre a qualidade do conteúdo gerado.

4.6 Análise dos resultados

Após a execução dos experimentos com as diferentes estratégias de *chunking* e modelos de *embeddings* selecionados, será realizada uma análise comparativa para identificar quais combinações apresentaram melhor desempenho na tarefa de automação de pareceres jurídicos.

Os resultados serão avaliados com base nas métricas definidas na metodologia, considerando tanto a qualidade textual das respostas geradas quanto a preservação do contexto jurídico essencial ao domínio dos pareceres analisados. Para cada métrica, serão calculadas as

médias e desvios padrão dos scores obtidos, permitindo uma análise quantitativa precisa.

Além da análise numérica, também será conduzida uma avaliação qualitativa, por meio da inspeção manual de uma amostra representativa das respostas geradas. Esse processo visa identificar aspectos que as métricas automáticas possam não captar, como coerência argumentativa, terminologia jurídica adequada e completude das informações.

Por fim, a partir da análise integrada dos resultados quantitativos e qualitativos, serão discutidas as melhores práticas para aplicação de Recuperação Aumentada por Geração (RAG) em documentos jurídicos, destacando as estratégias de *chunking* e *embeddings* mais adequadas ao contexto estudado, bem como possíveis limitações e sugestões para trabalhos futuros.

5 EXPERIMENTOS E RESULTADOS

Nesta seção, serão apresentados os resultados obtidos ao decorrer do trabalho. Os experimentos foram realizados utilizando um corpus de documentos de defesa considerados complexos por um especialista, onde modelos de linguagem não conseguiam extrair os argumentos precisamente. Inicialmente, procuramos entender como modelos de linguagem comuns no dia a dia estão se comportando na atividade de extração de argumentos, para logo em seguida desenvolver o sistema de extração e avaliar se os resultados do sistema seriam melhores do que os resultados padrão.

5.1 Avaliação de modelos de linguagem na tarefa de extração de argumentos.

Para essa etapa, conforme mapeado na seção 4.3, foi estabelecida uma estratégia para garantir que todos os modelos de linguagem tenham comparação justa, evitando viés e injustiças entre eles. A seguir, detalhamos a configuração do experimento e a execução do pipeline.

5.1.1 Seleção dos Modelos e Ferramentas

A seleção dos modelos para este estudo visou abranger uma gama diversificada de arquiteturas, origens e especializações, permitindo uma análise comparativa completa.

5.1.1.1 Modelos de Linguagem Avaliados

Foram selecionados oito grandes modelos de linguagem, contemplando tanto modelos de domínio público quanto proprietários. A escolha baseou-se em modelos já utilizados por especialistas jurídicos em suas práticas profissionais, diversificando custos, características técnicas e disponibilidade. A seleção incluiu:

- **Modelos da OpenAI:** GPT-4o, GPT-4o-mini. Amplamente utilizados devido à sua disponibilidade comercial e desempenho consolidado em tarefas de processamento de linguagem natural.
- **Modelos da Google:** Gemini 2.5 Pro, Gemini 2.0 Flash. Representam a abordagem tecnológica do Google, oferecendo alternativas competitivas com diferentes características de custo-benefício.
- **Modelos da Anthropic:** Claude 3.7 Sonnet. Reconhecido por seu desempenho superior

em *benchmarks*[1] e capacidade de raciocínio complexo.

- **Modelos DeepSeek:** DeepSeek-R1 com e sem a funcionalidade de *reasoning*. Modelos de custo acessível que permitem avaliar o impacto de capacidades de raciocínio explícito na extração de argumentos.
- **Modelos de Código Aberto:** Llama3-70B. Representa a categoria de modelos open-source, amplamente utilizado em implementações personalizadas e pesquisa acadêmica.

Durante a fase inicial de testes, foram também avaliados modelos executados localmente através do servidor Ollama, incluindo *Phi4*, *Llama3.1:8b*, *Llama3.2:3b* e *Qwen3:14b*. No entanto, estes modelos apresentaram limitações significativas no processamento de textos extensos (que poderiam chegar até 20.000 tokens), característicos dos documentos jurídicos analisados, resultando em desempenho inadequado para a tarefa proposta. Consequentemente, foram excluídos da análise comparativa final.

Essa diversidade de modelos selecionados permite avaliar como diferentes filosofias de treinamento e arquiteturas impactam a tarefa específica de extração de argumentos jurídicos.

5.1.2 Prompt padrão

Para garantir a extração dos argumentos de forma que os modelos tenham o mesmo contexto, usaremos um prompt padrão definido com a ajuda de um especialista jurídico que trabalha nessa extração no seu dia a dia.

[1] https://www.anthropic.com/news/claude-3-5-sonnet

Código-fonte 1 – Prompt para Extração de Argumentos de Defesa

```
OBJETIVO:
Atue como um ESPECIALISTA em direito, linguística, ciências cognitivas e
sociais. CONSIDERE o contexto jurídico específico, sua tarefa é
EXTRAIR exclusivamente os argumentos de defesa presentes no texto
fornecido.

FORMATO DO RETORNO:
- Liste cada argumento numerado, um por linha.
- Apenas os argumentos, sem adição de informações extras.
- Não inclua explicações ou introduções, apenas a lista numerada.
- Não reescreva, não interprete e não resuma: apenas transcreva ou adapte
o trecho que contenha o argumento, mantendo o sentido original.

INSTRUÇÕES:
- Considere como argumento qualquer justificativa, explicação, razão,
fato ou circunstância que tenha sido usado pela parte para se defender da
infração.
- Leve em consideração o porte da empresa.
- Não incluir informações meramente procedimentais como dados da empresa.
- Certifique-se de que os argumentos são reais e diretamente extraídos do
contexto fornecido.
- Não retorne nada além dos argumentos numerados.

CONTEXTUALIZAÇÃO:
{Defesa}
```

Fonte: Elaborado pelo autor.

O prompt do codigo-fonte 1 será usado na chamada em todos os modelos, garantindo assim a padronização da resposta, onde o campo defesa, é o texto do documento que será usado.

5.1.3 Texto em forma plana

O PDF não será passado diretamente para o modelo, pois muitos deles o tratam de um jeito diferente dos outros, então pode ocorrer a perda de informação, o que pode prejudicar o formato do retorno. Para a conversão de PDF em texto plano será usada a ferramenta PDF2Go[2], extraindo o texto e adicionando no prompt 1. Antes de passar o documento para o modelo, ele não deve ter outras requisições feitas num mesmo contexto, garantindo assim que ele não vai usar informações obtidas anteriormente.

5.1.4 Avaliação dos argumentos com similaridade do cosseno

Os argumentos obtidos por cada modelo serão avaliados com base na similaridade do cosseno, usando quatro diferentes tipos de *embedding*, possibilitando uma análise mais abrangente, já que todo *embedding* é treinado com modelos textuais diferentes. Os *embeddings* utilizados serão:

- **STJ IRIS:** Escolhido por seu treinamento específico com dados jurídicos brasileiros, permitindo capturar nuances terminológicas e contextuais do direito nacional, essencial para a análise de argumentos de defesa em processos administrativos.
- **Mini LM V6:** Modelo compacto e eficiente da Microsoft, amplamente utilizado para tarefas de similaridade semântica em textos gerais, oferecendo um baseline robusto para comparação com modelos especializados.
- **Nomic V2:** Modelo de código aberto treinado em dados textuais diversificados, proporcionando uma perspectiva complementar na avaliação de similaridade, especialmente relevante para capturar variações linguísticas em argumentações.
- **LaBSE - Google:** *Language-agnostic BERT Sentence Embedding* desenvolvido pelo Google, treinado multilingualmente, garantindo robustez na análise semântica independentemente de variações estilísticas ou regionais na linguagem jurídica.

Esta combinação de *embeddings* permite uma análise multidimensional: desde a especialização jurídica (STJ IRIS) até a generalização multilíngue (LaBSE), passando por modelos otimizados para eficiência (Mini LM V6) e diversidade textual (Nomic V2), proporcionando uma avaliação mais robusta e menos enviesada da qualidade dos argumentos extraídos.

5.1.5 Resultados

A Tabela 1 sintetiza os resultados agregados, permitindo uma análise do comportamento de cada modelo na tarefa de extração de argumentos de um corpus de defesas variadas. Os dados revelam diferenças significativas no desempenho, correlacionadas com:

[2] https://www.pdf2go.com/pt/pdf-para-texto

1. A arquitetura dos modelos
2. O domínio de treinamento

Tabela 1 - Médias dos Modelos por Métrica de Similaridade do cosseno

Modelo	STJ IRIS	Mini LM V6	Nomic V2	Labse - Google
OPENAI 4o	**0.861**	**0.774**	0.724	0.821
OPENAI 4o-mini	0.854	0.766	**0.731**	0.812
DEEPSEEK R1 S/T	0.848	0.715	0.660	0.774
DEEPSEEK R1 C/T	0.835	0.721	0.669	0.795
LLAMA3-70B-8192	0.846	0.760	0.689	**0.822**
GEMINI 2.5 PRO	0.857	0.760	0.721	0.802
GEMINI 2.0 FLASH	0.859	0.754	0.713	0.793
CLAUDE 3.7 SONNET	0.844	0.724	0.646	0.786

Fonte: Elaborado pelo autor

A análise da Tabela 1 permite identificar três padrões relevantes para o objetivo proposto:

1. **Superioridade de modelos especializados**: O STJ IRIS, treinado com dados jurídicos, apresentou as maiores correlações com os modelos avaliados (média de 0,853), colaborando com a hipótese de que *embeddings* de domínio específico capturam melhor nuances argumentativas no contexto jurídico.
2. **Variação por *embedding***: O desempenho relativo dos modelos muda significativamente conforme o *embedding* utilizado. Por exemplo, o Claude 3.7 Sonnet tem baixo desempenho no Nomic V2 (0,646), mas resultados competitivos no STJ IRIS (0,844), sugerindo que a escolha do *embedding* impacta diretamente a avaliação.
3. **Competição entre modelos proprietários**: OpenAI 4o e Gemini 2.5 Pro mostraram diferenças inferiores a 0,01 em três métricas, indicando que modelos comerciais de última geração atingem patamares similares, embora com variações em tarefas específicas.

Estes resultados demonstram que o comportamento dos modelos varia não apenas por sua arquitetura, mas também pela interação entre seu treinamento prévio e a métrica de avaliação adotada. O STJ IRIS, como esperado, mostrou-se particularmente eficaz para aplicações jurídicas, enquanto modelos genéricos de grande escala, como Llama3-70B, podem ser alternativas viáveis quando combinados com técnicas adequadas.

5.1.6 Análise Qualitativa por Especialista

Apesar de as métricas de similaridade fornecerem uma visão quantitativa, elas não capturam totalmente a qualidade das respostas dos modelos. A avaliação qualitativa realizada por um especialista jurídico revelou padrões de comportamento que os números por si só não demonstram. A análise focou na precisão, completude e aderência às instruções do prompt.

5.1.6.1 Tipos de Erros Observados

Os erros mais comuns foram categorizados da seguinte forma:

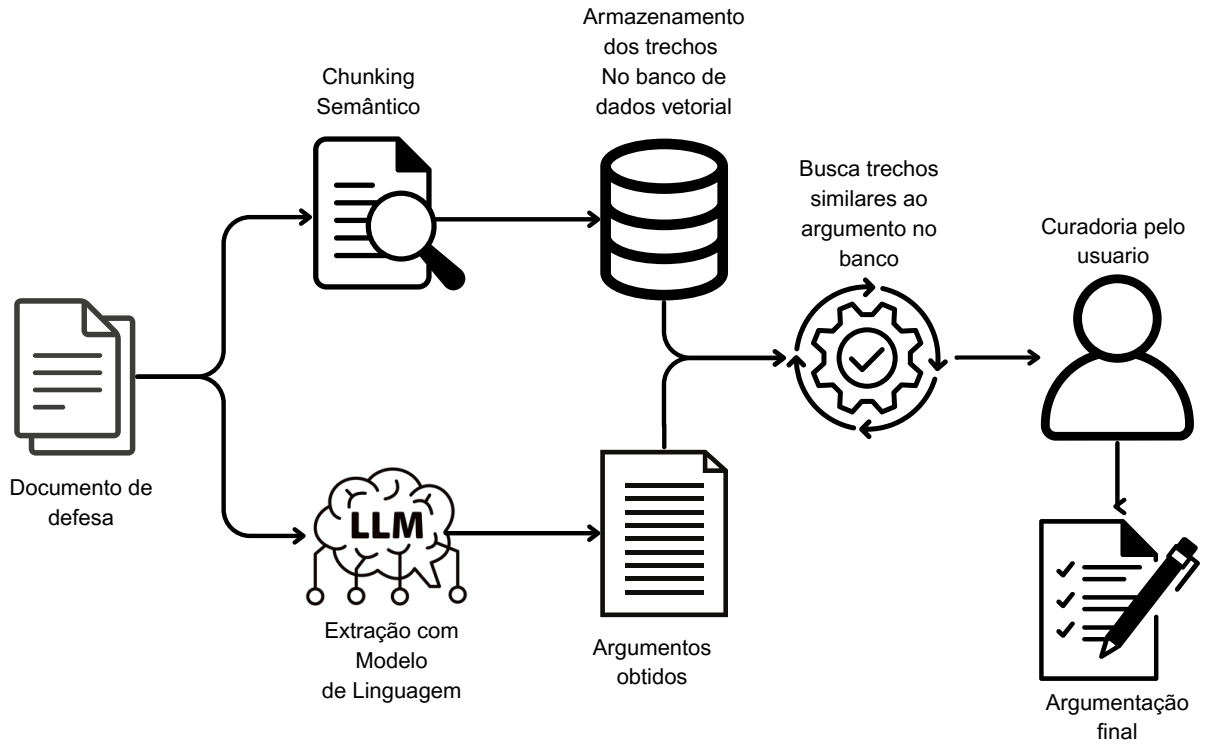
- **Alucinação de Argumentos:** O modelo inventa um argumento que não consta no texto original. Isso acontece em alguns modelos, principalmente em textos mais longos onde tentava "completar"com argumentos inexistentes a defesa.
- **Interpretação vs. Extração:** Alguns modelos tenderam a resumir ou reinterpretar o argumento em vez de transcrevê-lo, violando a instrução do prompt. Embora o resultado fosse semanticamente próximo (justificando uma boa pontuação na similaridade de cosseno), ele não atendia ao requisito de extração literal.
- **Repetição de argumentos:** Modelos frequentemente adicionavam argumentos que já tinham sido citados por ele mesmo anteriormente.

Esta análise qualitativa demonstra que, para um caso de uso jurídico onde a precisão literal é fundamental, o score de similaridade deve ser ponderado com uma avaliação manual dos resultados, confirmando que os modelos não apenas são semanticamente similares, mas também fazem sentido com o que foi pedido ao prompt.

5.2 Sistema proposto

A implementação experimental materializa uma plataforma de análise jurídica baseada na arquitetura RAG (*Retrieval-Augmented Generation*). O sistema foi projetado para otimizar a extração e validação de argumentos de defesa a partir de documentos, combinando a capacidade generativa de LLMs com a rastreabilidade de dados via similaridade vetorial e um robusto sistema de curadoria humana.

Figura 3 – Arquitetura do sistema proposto



Fonte: Elaborado pelo autor

O sistema foi elaborado com base na arquitetura da Figura 3, com isso o usuário submete um documento de defesa, e esse documento é processado paralelamente em dois passos, extração dos argumentos pelo modelo de linguagem e segmentação do texto em partes, baseado em *chunking* semântico. Após a segmentação os trechos são armazenados em um banco de dados vetorial. Com os argumentos extraídos pelo modelo e os trechos do documento original obtidos, conseguimos trabalhar com busca por similaridade para procurar trechos semanticamente parecidos com os argumentos extraídos, fornecendo assim ao usuário os três trechos mais similares com o argumento extraído, com isso o usuário consegue determinar se aquilo é ou não um argumento com base no nível de similaridade, ou vendo o trecho relacionado. Após isso, o usuário confirma se cada argumento está ou não correto e eles são salvos no banco de dados.

5.2.1 Arquitetura e Tecnologias

Nessa seção serão abordadas as técnicas utilizadas para construir o ecossistema usado no sistema.

5.2.1.1 Framework e Orquestração

O sistema é construído usando o framework LangChain[3], que gerencia a integração entre os componentes do pipeline de processamento como os LLMs. A interface interativa foi desenvolvida em Streamlit[4], permitindo a criação de uma aplicação web unificada que encapsula tanto o *frontend* quanto o *backend*.

5.2.1.2 Processamento Semântico e Vetorização

Para a representação vetorial dos textos, foi utilizado o modelo de *embedding stjiris/bert-large-portuguese-cased-legal-mlm-sts-v1.0*, especializado no domínio jurídico brasileiro. Este modelo é responsável por gerar os vetores de alta dimensionalidade que fundamentam o cálculo de similaridade semântica.

5.2.1.3 Armazenamento de Dados

A arquitetura de dados emprega uma abordagem dupla:

- **Supabase**[5]: utilizado como o banco de dados relacional principal para a persistência de todos os argumentos curados, metadados associados (número do processo, status, notas) e informações de rastreabilidade.
- **ChromaDB**[6]: empregado como um *vector store* temporário, em memória, durante a sessão de validação para realizar os cálculos de similaridade de forma eficiente e isolada.

5.2.2 Pipeline de Análise e Validação

O fluxo de processamento de um documento segue um pipeline estruturado em cinco etapas:

1. **Extração de Texto**: O sistema realiza o processamento de documentos em formato PDF para extrair o conteúdo textual bruto.
2. **Segmentação (Chunking)**: O texto extraído é segmentado em trechos menores e sobrepostos (*chunks*) utilizando o `chunking` semântico, uma estratégia que preserva a coesão contextual dos parágrafos.
3. **Geração de Argumentos**: Utilizando um LLM e o prompt 1, definido anteriormente, o sistema extrai os argumentos de defesa do documento. Um detalhe importante a se analisar nessa parte é

[3] https://www.langchain.com
[4] https://streamlit.io
[5] https://supabase.com
[6] https://www.trychroma.com

a limitação de *tokens* por modelos gratuitos, com isso foi necessário adaptar uma abordagem de segmentação de textos maiores que cinco mil *tokens*, quebrando eles com *semantic chunking* e realizando a extração em partes e depois juntando os argumentos extraídos.

4. **Validação por Similaridade**: Cada argumento extraído é comparado com todos os *chunks* do documento original através do cálculo da similaridade do cosseno. O sistema então apresenta os três trechos mais relevantes e a porcentagem de similaridade do cosseno obtido. Como pode se ver na Figura 4.

Figura 4 – Interface de exibição de argumentos extraidos

Fonte: Elaborado pelo autor

5. **Curadoria Humana e Anotação**: Com base nos dados de similaridade, o usuário especialista realiza a curadoria final, classificando os argumentos como “Confirmado” ou “Rejeitado” e, neste último caso, adicionando notas ou correções.

5.2.3 Persistência e Rastreabilidade

Os dados curados são persistidos no Supabase com um esquema estruturado que garante a rastreabilidade e a organização da informação, incluindo:

- Número do processo associado à análise.
- O texto do argumento e seu status de validação (confirmado/rejeitado).
- O *score* de similaridade obtido.
- Anotações do usuário para argumentos rejeitados.
- Metadados como referência do documento original e horário de extração.

5.2.4 Configurabilidade do Sistema

A plataforma oferece flexibilidade por meio de parâmetros configuráveis na interface:

- Seleção do modelo de LLM a ser utilizado.
- Ajuste do limite de similaridade para validação.
- Controle da temperatura do LLM para modular a criatividade da geração.

Esta arquitetura modular e configurável garante que o sistema possa ser adaptado e refinado continuamente.

5.3 Validação com base em métricas quantitativas

Utilizando o sistema proposto, foram extraídos argumentos de defesas consideradas complexas por um especialista, destacando não somente argumentos corretos, mas também argumentos errados (que estão abaixo do limiar ou não sejam considerados exatamente argumentos). Com base nisso, podemos calcular métricas quantitativas, comparando os argumentos extraídos pelo sistema com o "gabarito"fornecido por um especialista. Foram utilizadas as métricas:

- Bert score: calculando *precision, recall* f1 *score*, utilizando o modelo de *embedding ModernBERT-base*, que permite realizar o cálculo de *bert score* com textos maiores que 512 *tokens*.
- Similaridade do cosseno: Utilizando os quatro modelos de *embeddings* já utilizados anteriormente na seção 5.1 (STJ IRIS, Mini LM V6, Nomic V2, Labse - Google)

Para ilustrar o processo de comparação, a Tabela 2 apresenta um exemplo da saída gerada pelo Modelo X em contraste com o padrão de referência para uma das defesas analisadas.

Quadro 2 - Exemplo de correspondência entre padrões de referência e argumentos extraídos.

Padrão de Referência (criado por especialista)	**Argumentos Extraídos (saída do Modelo X)**
Defesa tempestiva	O prazo de 10 dias para apresentação da defesa teve início em 09/12/2024, motivo pelo qual resta tempestivo o protocolo em 19/12/2024.
Que é necessária a notificação prévia para regularização da suposta irregularidade	O fiscal do trabalho poderia, ao invés de lavrar a autuação, ter firmado Termo de Compromisso com a empresa, sendo necessária notificação prévia para regularização conforme art. 627-A da CLT.
Que a auditoria poderia ter apenas firmado termo de compromisso	A ausência do critério da dupla visita torna o auto de infração nulo de pleno direito, nos termos do art. 627 da CLT.

Fonte: Elaborado pelo autor

Com base no Quadro 2, podemos ver os argumentos "gabaritos", sendo esses fornecidos pelo especialista, e os argumentos extraídos por um modelo de exemplo. Com base nisso, calculamos as métricas utilizando os dois textos de forma integral para o cálculo. Com isso, as avaliações foram feitas usando dois bancos de dados diferentes, um que contemplava argumentos confirmados e rejeitados, e

outro que considera apenas argumentos confirmados (que estão acima do limiar definido pelo usuário).

Tabela 2 - Médias dos Modelos com BertScore, considerando aceitos e rejeitados

Métrica	deepseek-r1	gemma2-9b-it	llama-3.3-70b	llama-4-scout-17b	qwen3-32b	gpt-4o-mini
Precision	0.821	0.820	0.852	0.841	0.811	**0.862**
Recall	0.826	0.831	0.858	0.862	0.807	**0.880**
F1-Score	0.824	0.825	0.855	0.851	0.809	**0.871**

Fonte: Elaborado pelo autor

Nota: Valores em negrito indicam o melhor desempenho por métrica. Escala de 0 a 1, onde 1 representa similaridade perfeita.

Tabela 3 - Médias dos Modelos com Similaridade do cosseno, considerando aceitos e rejeitados

Embedding	deepseek-r1	gemma2-9b-it	llama-3.3-70b	llama-4-scout-17b	qwen3-32b	gpt-4o-mini
STJ IRIS	0.798	0.850	**0.894**	0.853	0.627	0.838
Mini LM V6	0.710	0.780	**0.808**	0.765	0.497	0.776
Nomic V2	0.703	0.766	0.750	**0.777**	0.591	0.739
Labse - Google	0.756	0.811	**0.812**	0.755	0.748	0.781

Fonte: Elaborado pelo autor

Nota: Valores em negrito indicam o melhor desempenho por métrica. Escala de 0 a 1, onde 1 representa similaridade perfeita.

As Tabelas 2 e 3 apresentam a avaliação que considera o conjunto de dados completo, nela observa-se que todos os modelos alcançaram pontuações de BERTScore (Precision, Recall e F1) significativamente altas, acima de 0.81, indicando uma forte capacidade de gerar textos semanticamente alinhados com os argumentos de referência. O modelo gpt-40-mini obteve o maior F1-score (0.871), demonstrando o melhor equilíbrio entre precisão e recall. O llama-3.3-70b também se destacou, ficando em segundo lugar em F1-score. O modelo llama-3.3-70b se destacou nas métricas de similaridade do cosseno, especialmente com o *embedding* STJ IRIS (0.893), superando até um modelo pago como o gpt-4o-mini, o que sugere que suas respostas são semanticamente muito próximas do padrão de referência quando analisadas por esses modelos de *embedding*. Em contrapartida, o modelo qwen3-32b apresentou o desempenho mais baixo neste cenário mais complexo.

Tabela 4 - Médias dos Modelos com BertScore, considerando apenas aceitos

Métrica	deepseek-r1	gemma2-9b-it	llama-3.3-70b	llama-4-scout-17b	qwen3-32b	gpt-4o-mini
Precision	0.871	0.872	**0.915**	0.913	0.868	0.911
Recall	0.878	0.888	0.935	**0.937**	0.876	0.929
F1-Score	0.875	0.880	**0.925**	0.925	0.872	0.920

Fonte: Elaborado pelo autor

Nota: Valores em negrito indicam o melhor desempenho por métrica. Escala de 0 a 1, onde 1 representa similaridade perfeita.

Tabela 5 - Médias dos Modelos com Similaridade do cosseno, considerando apenas aceitos

Embedding	deepseek-r1	gemma2-9b-it	llama-3.3-70b	llama-4-scout-17b	qwen3-32b	gpt-4o-mini
STJ IRIS	0.819	0.860	**0.893**	0.854	0.790	0.838
Mini LM V6	0.710	0.801	**0.808**	0.765	0.628	0.776
Nomic V2	0.708	0.767	0.754	**0.777**	0.725	0.739
Labse - Google	0.746	0.813	0.812	0.755	**0.840**	0.781

Fonte: Elaborado pelo autor

Nota: Valores em negrito indicam o melhor desempenho por métrica. Escala de 0 a 1, onde 1 representa similaridade perfeita.

Nas tabelas 4 e 5, a tendência de alto desempenho se mantém e, em geral, as pontuações melhoram, como esperado em um cenário que não considera as respostas abaixo do limiar. Os modelos da família Llama continuam na liderança, com o llama-3.3-70b e o llama-4-scout-17b empatando no F1-score (0.925). Um resultado interessante surge na métrica de similaridade com o modelo LaBSE, onde o qwen3-32b salta para a primeira posição com uma pontuação expressiva de 0.840. Isso sugere que, embora o qwen3-32b possa ter dificuldades em cenários mais complexos como com os dados completos, quando ele trabalha somente na avaliação de argumentos válidos, ele se sai melhor, como é possível ver em todas as similaridades que aumentaram significativamente.

5.4 Validação com base em métricas qualitativas

A avaliação qualitativa é essencial para determinar quão bem o sistema está extraindo argumentos de forma que um usuário final consiga avaliar a qualidade e a fidelidade do conteúdo gerado. Para validar os argumentos produzidos pelo sistema, utilizamos duas estratégias complementares:

- **DeepEval**: uma biblioteca que funciona como "LLM as a judger"que utiliza modelos de linguagem (neste caso, gpt-4o-mini) para comparar os argumentos gerados com os esperados, levando em consideração a corretude e completude dos argumentos extraídos.
- **Avaliação de especialista**: análise manual realizada por um profissional da área jurídica, que examina a coerência e consistência dos argumentos, oferecendo um julgamento qualitativo e fundamentado sobre a qualidade da extração.

5.4.1 Configuração da avaliação e resultados com DeepEval

Para avaliar os argumentos extraídos com a biblioteca DeepEval que usa um modelo de linguagem para avaliar as respostas (neste caso, gpt-4o-mini), iremos nos basear na mesma estrutura da Tabela 2, para isso com ela precisamos seguir um passo a passo, pois é necessário explicar para o modelo como a avaliação deve ser feita, com isso definimos as chamadas métricas:

5.4.1.1 Corretude

Para a métrica de avaliação de corretude, foram definidos os seguintes passos, considerando o critério e as "rubricas", como a biblioteca define a nota com base em faixas de pontuação estipuladas pelo usuário.

O critério de avaliação definido na biblioteca foi:

- Avalie se todos os argumentos apresentados na saída do modelo (`actual_output`) estão corretos em relação aos argumentos esperados (`expected_output`).
- Um argumento é considerado correto se ele estiver presente no conjunto de argumentos esperados, com o mesmo sentido e contexto.
- Argumentos que não aparecem na referência, distorcem o conteúdo original ou foram claramente inventados devem ser considerados incorretos.
- Ignore os argumentos omitidos; avalie apenas os que foram de fato incluídos pelo modelo.
- Sua tarefa é identificar e penalizar a presença de informações incorretas, irrelevantes ou não autorizadas.

E a nota foi calculada da seguinte maneira:

- **Pontuação de 0 a 2:** Resposta contém muitos argumentos incorretos ou inventados, comprome-

tendo a validade da extração.

- **Pontuação de 3 a 5:** Alguns argumentos estão corretos, mas há várias imprecisões ou alucinações que afetam a confiabilidade da resposta.
- **Pontuação de 6 a 9:** A maioria dos argumentos está correta, com poucas imprecisões ou desvios semânticos.
- **Pontuação 10:** Todos os argumentos apresentados estão corretos e refletem fielmente os argumentos esperados, tanto em sentido quanto em contexto.

5.4.1.2 Completude

Para a métrica de avaliação de completude, foram definidos os seguintes passos, considerando o critério e as "rubricas", como a biblioteca define a nota com base em faixas de pontuação estipuladas pelo usuário.

O critério de avaliação definido na biblioteca foi:

- Avalie se a saída do modelo (`actual_output`) cobre adequadamente os argumentos presentes na referência (`expected_output`).
- Um argumento é considerado coberto se ele estiver presente, mesmo com palavras diferentes, desde que mantenha o mesmo sentido e contexto.
- Não penalize a presença de argumentos extras; ignore-os completamente para esta avaliação.
- O foco é verificar se os principais argumentos esperados foram incluídos na resposta do modelo.
- Argumentos omitidos devem ser penalizados.
- Sua tarefa é avaliar o grau de cobertura dos argumentos esperados.

E a nota foi calculada da seguinte maneira:

- **Pontuação de 0 a 2:** Resposta omite a maioria dos argumentos esperados.
- **Pontuação de 3 a 5:** Poucos argumentos esperados foram recuperados.
- **Pontuação de 6 a 9:** A maioria dos argumentos esperados foi incluída.
- **Pontuação 10:** Quase todos ou todos os argumentos esperados foram extraídos corretamente.

5.4.1.3 Resultados da avaliação com DeepEval

Após a definição do ambiente utilizado para executar a biblioteca e avaliar os argumentos extraídos pelo modelo, conseguimos avaliar os argumentos extraídos. Com base nisso, em uma defesa considerada complexa, conseguimos a seguinte nota com justificativa, baseada na avaliação com o *gpt-4o-mini*:

Quadro 3 - Resultados considerando argumentos rejeitados e confirmados com métricas de Corretude e Completude

Métrica	**Score**	**Limiar**	**Justificativa**
Corretude	0.716	0.8	A resposta apresentou a maioria dos argumentos centrais corretamente alinhados com o esperado, como a tempestividade da defesa, ausência de irregularidades e a falha na orientação. No entanto, omitiu pontos importantes como a jurisprudência favorável e a necessidade de oportunidade para correção.
Completude	0.812	0.8	A saída abrangeu a maioria dos argumentos esperados, como a natureza educativa da fiscalização. Faltaram menções específicas, como o erro na quantificação e a jurisprudência favorável, o que impediu a pontuação máxima.

Quadro 4 - Resultados considerando apenas argumentos confirmados com métricas de Corretude e Completude

Métrica	**Score**	**Limiar**	**Justificativa**
Corretude	0.799	0.8	Os principais argumentos foram identificados corretamente. No entanto, a omissão do princípio da dupla visita e do erro na quantificação comprometeu levemente a nota final. A estrutura e a relevância foram bem mantidas.
Completude	0.871	0.8	A resposta foi abrangente e contemplou os principais argumentos, como a natureza educativa da fiscalização e o esforço da empresa. Faltaram apenas alguns pontos específicos, como o erro na quantificação e a jurisprudência favorável.

Os Quadros 3 e 4 explicam a estrutura do retorno da biblioteca após uma avaliação de argumentos, indicando a nota e a razão dessa nota. Após essa explicação, conseguimos agora então definir a média baseada nos scores que o modelo deu para cada modelo e métrica.

No cenário mais complexo e com mais erros, como podemos ver nas Tabelas 6, o gpt-4o-mini se destaca como o modelo mais correto e completo, indicando que é o mais confiável para não gerar informações imprecisas e ainda garantir que nenhum argumento importante seja esquecido, mesmo que ao custo de pagar por *token*. Como modelo gratuito nesse aspecto, temos como destaque o DeepSeek como

Tabela 6 - Médias dos Modelos, considerando aceitos e rejeitados

Métrica	deepseek-r1	gemma2-9b-it	llama-3.3-70b	llama-4-scout-17b	qwen3-32b	gpt-4o-mini
Corretude	0.636	0.693	0.712	0.743	0.608	**0.757**
Completude	0,840	0.764	0.819	0.801	0.781	**0.856**

Fonte: Elaborado pelo autor

Nota: Valores em negrito indicam o melhor desempenho por métrica. Escala de 0 a 1, onde 1 representa similaridade perfeita.

Tabela 7 - Médias dos Modelos, considerando apenas aceitos

Métrica	deepseek-r1	gemma2-9b-it	llama-3.3-70b	llama-4-scout-17b	qwen3-32b	gpt-4o-mini
Corretude	0.650	0.677	**0.767**	0.729	0.688	0.757
Completude	0,811	0.776	**0.823**	0.722	0.700	0.823

Fonte: Elaborado pelo autor

Nota: Valores em negrito indicam o melhor desempenho por métrica. Escala de 0 a 1, onde 1 representa similaridade perfeita.

modelo mais eficiente na completude chegando ao *score* de 0.840, enquanto o modelo llama-4-scout-17b se destaca por sua corretude de 0.743.

Quando passamos para o cenário de dados mais limpos, como na Tabela 7, o llama-3.3-70b é o modelo mais equilibrado, liderando tanto em corretude quanto em completude, com scores 0.767 e 0.823. Isso indica que, com dados de entrada de maior qualidade, este modelo oferece a melhor combinação de precisão e cobertura, o interessante é que o modelo gpt-4o-mini não apresenta grandes mudanças entre os dois tipos de avaliação, o que sugere que o modelo tende a extrair menos argumentos incorretos.

5.4.2 Avaliação Qualitativa pelo especialista

Esta sessão engloba a avaliação da qualidade dos argumentos extraídos com o apoio do sistema por um usuário especialista na área jurídica. Essa avaliação foi feita observando pontos já citados como na sessão 4.5.2.2, observando nesse caso, fraquezas dos argumentos extraídos pelo sistema.

Após o uso do sistema pelo usuário, foram detectados alguns problemas relacionados aos modelos ao extrair argumentos:

- **Modelos retornam muito contexto:** Um dos problemas relatados foi que modelos de forma geral acabam retornando respostas com muito mais contexto ao invés de uma resposta mais direta, o que indica que os modelos não tiveram uma boa aderência ao prompt, por mais que tenha argumentado corretamente.
- **Modelos repetem argumentos:** Os modelos, de forma geral, repetem argumentos. Um argumento poderia aparecer no começo do texto e também no final, o que indica que poderia haver no prompt uma informação mais explicita para realizar essa tratativa, considerando-o somente como um argumento único.

- **Modelos usando cadeia de pensamento:** Não foi realizada a tratativa de modelos que usavam cadeia de pensamento, como DeepSeek, retornando assim trechos do seu pensamento. Como o sistema considerava como correto somente o que estava acima do limiar definido, isso não foi um problema na avaliação dos argumentos corretos, mas na avaliação completa provavelmente houve influência.

Após uma análise do sistema pelo especialista, chega-se à conclusão de que, por mais que o sistema forneça uma forma de avaliação segura e baseada em trechos similares que realmente constam no documento original, observa-se que alguns modelos são bem superiores aos outros qualitativamente, como, por exemplo, o llama-4-scout-17b, que no uso do usuário se destacou tanto em corretude quanto em completude, indicando que até mesmo um modelo de código aberto e uso gratuito pode, sim, chegar a ser uma opção viável para esta tarefa.

6 CONCLUSÕES E TRABALHOS FUTUROS

Nas seções a seguir, finalizamos nosso trabalho com a conclusão de ideias para passos futuros que complementem o presente trabalho.

6.1 Conclusões

Os grandes modelos de linguagem trabalham de forma convincente na tarefa de extração de argumentos no contexto jurídico, como podemos ver na seção 5.1, contudo é necessária uma curadoria por parte do usuário para garantir que os argumentos extraídos tenham fundamentação baseada no documento original. Com isso, podemos chegar à conclusão que, ao fornecer esse apoio ao usuário de forma que os argumentos sejam exibidos juntamente com os trechos relacionados, é uma estratégia funcional e de fácil aplicação.

Com essa estratégia, até mesmo modelos gratuitos podem alcançar um patamar de modelos pagos, possibilitando assim o uso de tais modelos para extração convincente, como o exemplo do *llama3.3-70b* e *llama-4-scout-17*b, que se saíram muito bem nos testes do sistema.

A eficácia dessa abordagem reside na sua capacidade de transformar a saída de um modelo de linguagem, em um conjunto de argumentos verificáveis. Ao fornecer rastreabilidade, permitindo que o usuário visualize os trechos específicos do documento que fundamentam cada argumento, o sistema diminui o risco de alucinações e aumenta a confiança do profissional na ferramenta. O objetivo principal deste trabalho é aprimorar a confiabilidade e precisão da extração, alcançado não apenas pelo desempenho do modelo, mas pela arquitetura de validação que ele usa.

Adicionalmente, a avaliação aprofundada revelou um importante equilíbrio entre corretude e completude na extração dos argumentos. Os experimentos demonstraram que, dependendo do modelo, pode haver uma tendência a priorizar a completude em detrimento da corretude, e vice-versa. O sistema proposto capacita o especialista a navegar nesse ponto, utilizando os scores de similaridade como um guia para focar sua atenção nos argumentos que necessitam de uma revisão mais criteriosa.

Outra contribuição significativa deste trabalho foi a validação da superioridade de ferramentas de domínio específico. A utilização do modelo de *embedding* STJ IRIS demonstrou-se mais adequada para capturar as nuances semânticas do vocabulário jurídico brasileiro, um achado que reforça a necessidade de especialização em aplicações de inteligência artificial para o Direito.

Em suma, este trabalho conclui que a integração eficaz de modelos de linguagem no setor jurídico depende menos da busca por um modelo perfeito e mais da construção de sistemas híbridos que aliem a capacidade de processamento da máquina à necessidade humana de verificação, contexto e controle. A abordagem proposta, aliada a uma interface de curadoria, prova ser um caminho promissor

para o desenvolvimento de ferramentas de inteligência artificial que sejam verdadeiramente úteis e confiáveis para os profissionais.

6.2 Trabalhos futuros

Com o desenvolvimento desse trabalho, podemos encontrar algumas oportunidades para o desenvolvimento e ampliação de novos trabalhos. Podemos verificar essas sugestões nas seções a seguir.

6.2.1 Melhoria na construção do prompt

Um dos problemas identificados no sistema foi que muitas vezes os modelos repetiam argumentos já citados por eles próprios anteriormente, o que pode ser causado por uma falta de aderência ao prompt. Investigar o impacto de diferentes técnicas de engenharia de prompt, como a few-shot prompting (fornecendo exemplos de extração correta no próprio prompt) ou a Chain-of-Thought (instruindo o modelo a 'pensar passo a passo' antes de extrair o argumento), para reduzir a repetição e aumentar a aderência às instruções. Melhorias no prompt podem resultar em respostas melhores futuramente.

6.2.2 Estudo de novos modelos de embedding

O modelo de *embedding* STJ IRIS se demonstrou convincente para o cálculo de similaridade nesse estudo, porém para um trabalho futuro poderiam ser usados novos modelos de *embedding*, ou até mesmo o treinamento de um modelo baseado em defesas e argumentos, melhorando assim a resposta da transformação de textos em vetores de alta dimensionalidade.

6.2.3 Sistema de extração baseado em agentes

Atualmente, o sistema se comporta como se fosse somente um agente trabalhando na tarefa de extração e escrita desses mesmos argumentos, porém, futuramente, pode ser aplicada uma estratégia de usar agentes, onde diferentes modelos assumem papéis especializados: um agente pesquisador que identifica as seções mais relevantes do documento, um agente extrator que extrai os argumentos dessas seções, e um agente revisor que verifica a coerência e a não repetição do resultado. Essa abordagem poderia simular um fluxo de trabalho colaborativo e aumentar a robustez da extração.

6.2.4 Construção de um sistema de RAG

O sistema atual utiliza uma arquitetura RAG para a validação de argumentos. Um trabalho futuro promissor seria expandir essa aplicação para a geração de novos textos. Os argumentos extraídos e

validados pelo especialista formariam uma base de conhecimento de alta qualidade. Utilizando esta base, um sistema RAG poderia, por exemplo, gerar um resumo coeso de uma defesa ou até mesmo elaborar um rascunho de um parecer. Isso permitiria que o modelo gerasse respostas fundamentadas nos melhores argumentos de casos anteriores, aumentando a consistência e a qualidade do texto final.

REFERÊNCIAS

BACHINGER, S. T.; FEDDOUL, L.; MAUCH, M. J.; KöNIG-RIES, B. Extracting legal norm analysis categories from german law texts with large language models. In: **Proceedings of the 25th Annual International Conference on Digital Government Research**. New York, NY, USA: Association for Computing Machinery, 2024. (dg.o '24), p. 481–493. ISBN 9798400709883. Disponível em: https://doi.org/10.1145/3657054.3657277. Acesso em: 06 ago. 2025.

BARAKAT, B.; HUANG, Q. Improving reliability of fine-tuning with block-wise optimisation. 2023. Disponível em: https://doi.org/10.48550/arXiv.2301.06133. Acesso em: 06 ago. 2025.

BRUNO, A.; MAZZEO, P. L.; CHETOUANI, A.; TLIBA, M.; KERKOURI, M. A. **Insights into Classifying and Mitigating LLMs' Hallucinations**. 2023. Disponível em: https://arxiv.org/abs/2311.08117. Acesso em: 06 ago. 2025.

CHANG, Y.; WANG, X.; WANG, J.; WU, Y.; YANG, L.; ZHU, K.; CHEN, H.; YI, X.; WANG, C.; WANG, Y. *et al.* A survey on evaluation of large language models. **ACM Transactions on Intelligent Systems and Technology**, ACM New York, NY, v. 15, n. 3, p. 1–45, 2024. Disponível em: https://doi.org/10.1145/3641289. Acesso em: 06 ago. 2025.

COELHO, A. Z. A transformação digital e o uso de técnicas inteligência artificial (ia) no sistema de justiça do brasil. **e-publica**, 2024. Disponível em: https://doi.org/10.47345/v11n1art5. Acesso em: 06 ago. 2025.

DAHL, M.; MAGESH, V.; SUZGUN, M.; HO, D. E. Large legal fictions: Profiling legal hallucinations in large language models. **Journal of Legal Analysis**, Oxford University Press (OUP), v. 16, n. 1, p. 64–93, jan. 2024. ISSN 1946-5319. Disponível em: http://dx.doi.org/10.1093/jla/laae003. Acesso em: 06 ago. 2025.

FABRE, e. a. A. La transformación de la práctica legal con la inteligencia artificial. **Interconectando Saberes**, 2024. Disponível em: https://doi.org/10.25009/is.v0i17.2837. Acesso em: 06 ago. 2025.

FARIA, J. R. de; XIE, H.; STEFFEK, F. **Automatic Information Extraction From Employment Tribunal Judgements Using Large Language Models**. 2024. Disponível em: https://arxiv.org/abs/2403.12936. Acesso em: 06 ago. 2025.

FEDUS, W.; ZOPH, B.; SHAZEER, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. **Journal of Machine Learning Research**, v. 23, n. 120, p. 1–39, 2022. Disponível em: http://jmlr.org/papers/v23/21-0998.html. Acesso em: 06 ago. 2025.

GAO, Y.; XIONG, Y.; GAO, X.; JIA, K.; PAN, J.; BI, Y.; DAI, Y.; SUN, J.; WANG, M.; WANG, H. **Retrieval-Augmented Generation for Large Language Models**: A survey. 2024. Disponível em: https://arxiv.org/abs/2312.10997. Acesso em: 06 ago. 2025.

GUO, M.-H.; XU, T.-X.; LIU, J.-J.; LIU, Z.-N.; JIANG, P.-T.; MU, T.-J.; ZHANG, S.-H.; MARTIN, R. R.; CHENG, M.-M.; HU, S.-M. Attention mechanisms in computer vision: A survey. **Computational Visual Media**, v. 8, n. 3, p. 331–368, 2022.

GUO, T.; CHEN, X.; WANG, Y.; CHANG, R.; PEI, S.; CHAWLA, N. V.; WIEST, O.; ZHANG, X. Large language model based multi-agents: A survey of progress and challenges. 2024. Disponível em: https://doi.org/10.48550/arXiv.2402.01680. Acesso em: 06 ago. 2025.

HAN, Y.; LIU, C.; WANG, P. **A Comprehensive Survey on Vector Database**: Storage and retrieval technique, challenge. 2023. Disponível em: https://arxiv.org/abs/2310.11703. Acesso em: 06 ago. 2025.

HAN, Z.; GAO, C.; LIU, J.; ZHANG, J.; ZHANG, S. Q. Parameter-efficient fine-tuning for large models: A comprehensive survey. **Transactions on Machine Learning Research**, 2024. ISSN 2835-8856. Disponível em: https://openreview.net/forum?id=lIsCS8b6zj. Acesso em: 06 ago. 2025.

HUANG, L.; YU, W.; MA, W.; ZHONG, W.; FENG, Z.; WANG, H.; CHEN, Q.; PENG, W.; FENG, X.; QIN, B.; LIU, T. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. **ACM Transactions on Information Systems**, Association for Computing Machinery (ACM), v. 43, n. 2, p. 1–55, jan. 2025. ISSN 1558-2868. Disponível em: http://dx.doi.org/10.1145/3703155. Acesso em: 06 ago. 2025.

HUSSAIN, A. S.; THOMAS, A. **Large Language Models for Judicial Entity Extraction**: A comparative study. 2024. Disponível em: https://arxiv.org/abs/2407.05786. Acesso em: 06 ago. 2025.

JUVEKAR, K.; PURWAR, A. **COS-Mix**: Cosine similarity and distance fusion for improved information retrieval. 2024. Disponível em: https://arxiv.org/abs/2406.00638. Acesso em: 06 ago. 2025.

KSHIRSAGAR, A. **Enhancing RAG Performance Through Chunking and Text Splitting Techniques**. 2024. Disponível em: https://doi.org/10.32628/CSEIT2410593. Acesso em: 06 ago. 2025.

LEWIS, P.; PEREZ, E.; PIKTUS, A.; PETRONI, F.; KARPUKHIN, V.; GOYAL, N.; KüTTLER, H.; LEWIS, M.; YIH, W. tau; ROCKTäSCHEL, T.; RIEDEL, S.; KIELA, D. Retrieval-augmented generation for knowledge-intensive nlp tasks. 2021. Disponível em: https://doi.org/10.48550/arXiv.2005.11401. Acesso em: 06 ago. 2025.

LIN, T.; WANG, Y.; LIU, X.; QIU, X. A survey of transformers. **AI Open**, v. 3, p. 111–132, 2022. ISSN 2666-6510. Disponível em: https://www.sciencedirect.com/science/article/pii/S2666651022000146. Acesso em: 06 ago. 2025.

LU, Z. Image feature selection based on attention mechanism. **Academic Journal of Science and Technology**, v. 11, n. 3, p. 85–88, Jul. 2024. Disponível em: https://drpress.org/ojs/index.php/ajst/article/view/23543.

MA, S.; CHU, Q.; MAO, J.; JIANG, X.; DUAN, H.; CHEN, C. **Leveraging Large Language Models for Relevance Judgments in Legal Case Retrieval**. 2025. Disponível em: https://arxiv.org/abs/2403.18405. Acesso em: 06 ago. 2025.

MA, Y.; SHAO, Y.; WU, Y.; LIU, Y.; ZHANG, R.; ZHANG, M.; MA, S. Lecard: A legal case retrieval dataset for chinese law system. In: **Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: Association for Computing Machinery, 2021. (SIGIR '21), p. 2342–2348. ISBN 9781450380379. Disponível em: https://doi.org/10.1145/3404835.3463250. Acesso em: 06 ago. 2025.

ORGAD, H.; TOKER, M.; GEKHMAN, Z.; REICHART, R.; SZPEKTOR, I.; KOTEK, H.; BELINKOV, Y. **LLMs Know More Than They Show**: On the intrinsic representation of llm hallucinations. 2024. Disponível em: https://arxiv.org/abs/2410.02707. Acesso em: 06 ago. 2025.

PAN, J. J.; WANG, J.; LI, G. **Survey of vector database management systems**. 2024. Disponível em: https://doi.org/10.1007/s00778-024-00864-x. Acesso em: 06 ago. 2025.

PLATCHIAS, M. e. **Hallucination: Philosophy and psychology**. [*S. l.*]: MIT Press, 2013.

QU, R.; TU, R.; BAO, F. **Is Semantic Chunking Worth the Computational Cost?** 2024. Disponível em: https://arxiv.org/abs/2410.13070. Acesso em: 06 ago. 2025.

RAU, D.; WANG, S.; DéJEAN, H.; CLINCHANT, S. **Context Embeddings for Efficient Answer Generation in RAG**. 2024. Disponível em: https://arxiv.org/abs/2407.09252. Acesso em: 06 ago. 2025.

RAWTE, V.; SHETH, A.; DAS, A. **A Survey of Hallucination in Large Foundation Models**. 2023. Disponível em: https://arxiv.org/abs/2309.05922. Acesso em: 06 ago. 2025.

THIRUNAVUKARASU, A. J.; TING, D. S. J.; ELANGOVAN, K.; GUTIERREZ, L.; TAN, T. F.; TING, D. S. W. Large language models in medicine. **Nature Medicine**, 2023. Disponível em: https://doi.org/10.1038/s41591-023-02448-8. Acesso em: 06 ago. 2025.

TONMOY, S. M. T. I.; ZAMAN, S. M. M.; JAIN, V.; RANI, A.; RAWTE, V.; CHADHA, A.; DAS, A. **A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models**. 2024. Disponível em: https://arxiv.org/abs/2401.01313. Acesso em: 06 ago. 2025.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; POLOSUKHIN, I. **Attention Is All You Need**. 2023. Disponível em: https://arxiv.org/abs/1706.03762. Acesso em: 06 ago. 2025.

VATSAL, S.; DUBEY, H. **A Survey of Prompt Engineering Methods in Large Language Models for Different NLP Tasks**. 2024. Disponível em: https://doi.org/10.48550/arXiv.2407.12994. Acesso em: 06 ago. 2025.

WANG, X.; WANG, Z.; GAO, X.; ZHANG, F.; WU, Y.; XU, Z.; SHI, T.; WANG, Z.; LI, S.; QIAN, Q.; YIN, R.; LV, C.; ZHENG, X.; HUANG, X. Searching for best practices in retrieval-augmented generation. 2024. Disponível em: https://doi.org/10.48550/arXiv.2407.01219. Acesso em: 06 ago. 2025.

XIAO, Y.; JIN, Y.; BAI, Y.; WU, Y.; YANG, X.; LUO, X.; YU, W.; ZHAO, X.; LIU, Y.; GU, Q.; CHEN, H.; WANG, W.; CHENG, W. Privacymind: Large language models can be contextual privacy protection learners. 2024. Disponível em: https://arxiv.org/abs/2310.02469. Acesso em: 06 ago. 2025.

YANG, H.; LIU, X.-Y.; WANG, C. D. Fingpt: Open-source financial large language models. 2023. Disponível em: https://doi.org/10.48550/arXiv.2306.06031. Acesso em: 06 ago. 2025.

YEPES, A. J.; YOU, Y.; MILCZEK, J.; LAVERDE, S.; LI, R. **Financial Report Chunking for Effective Retrieval Augmented Generation**. 2024. Disponível em: https://arxiv.org/abs/2402.05131. Acesso em: 06 ago. 2025.

ZHANG, T.; KISHORE, V.; WU, F.; WEINBERGER, K. Q.; ARTZI, Y. **BERTScore**: Evaluating text generation with bert. 2020. Disponível em: https://arxiv.org/abs/1904.09675. Acesso em: 06 ago. 2025.

ZHAO, P.; ZHANG, H.; YU, Q.; WANG, Z.; GENG, Y.; FU, F.; YANG, L.; ZHANG, W.; JIANG, J.; CUI, B. **Retrieval-Augmented Generation for AI-Generated Content**: A survey. 2024. Disponível em: https://arxiv.org/abs/2402.19473. Acesso em: 06 ago. 2025.

ZHAO, W. X.; ZHOU, K.; LI, J.; TANG, T.; WANG, X.; HOU, Y.; MIN, Y.; ZHANG, B.; ZHANG, J.; DONG, Z.; DU, Y.; YANG, C.; CHEN, Y.; CHEN, Z.; JIANG, J.; REN, R.; LI, Y.; TANG, X.; LIU, Z.; LIU, P.; NIE, J.-Y.; WEN, J.-R. **A Survey of Large Language Models**. 2024. Disponível em: https://arxiv.org/abs/2303.18223. Acesso em: 06 ago. 2025.

ZHENG, S.; ZHANG, Y.; ZHU, Y.; XI, C.; GAO, P.; ZHOU, X.; CHANG, K. C.-C. **GPT-Fathom**: Benchmarking large language models to decipher the evolutionary path towards gpt-4 and beyond. 2024. Disponível em: https://arxiv.org/abs/2309.16583. Acesso em: 06 ago. 2025.

..