A Work Project, presented as part of the requirements for the Award of a Master's degree in Business Analytics from the Nova School of Business and Economics.

Graph-Based Reasoning for Retrieval-Augmented Generation: A Study in the Portuguese Legal Domain

Maria Leonor Trindade Hermenegildo

Work project carried out under the supervision of:

Qiwei Han

10/01/2025

Abstract

This study explores RAG systems tailored to the Portuguese legal domain, highlighting challenges in underrepresented languages. Fixed-size chunking strategies, particularly *TokenTextSplitter*, were found to be most effective, while more advanced techniques like Recursive and Semantic splitting showed little benefits. Larger chunk sizes improved retrieval accuracy and answer quality, though the impact of chunk overlap remains inconclusive. Self-reflection techniques show promising results, particularly for weaker LLMs and when different techniques are paired. However, there is an increment in computational cost to consider.

Keywords

Retrieval-Augmented Generation, RAG, Large Language Models, LLM, Artificial Intelligence, AI, Hallucination, Question Answering, RAG evaluation, Vector Store, Chunking, Legal AI, Graph-Based Reasoning, Self-Assessment, Self-Reflection, Multi-Agent Systems, MAS

This work used infrastructure and resources funded by Fundação para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22209).

Definitions

- AI: Artificial Intelligence
- BERT: Bidirectional Encoder Representations from Transformers
- CoT: Chain of Thought
- DL: Deep Learning
- GenAI: Generative Artificial Intelligence
- GraphRAG: Graph Retrieval-Augmented Generation
- LLM(s): Large Language Model(s)
- MAS(s): Multi Agent System(s)
- ML: Machine Learning
- NLP: Natural Language Processing
- RAG: Retrieval-Augmented Generation
- RNN: Recurrent Neural Network
- SOTA: State-of-the-art

1 Introduction

The rapid evolution of Large Language Models (LLMs) has marked a significant turning point over the years in the application of Artificial Intelligence (AI) across various domains. In recent years, research highlights that specialization, rather than merely increasing parameter counts, can significantly enhance model performance while minimizing inference costs (Zhao et al., 2023; Almeida et al., 2024). In the legal domain, this specialization is particularly impactful, as advances in AI, such as in Machine Learning (ML) and Natural Language Processing (NLP), are reshaping how legal professionals and law firms approach their work.

Legal professionals must often manage and interpret vast amounts of complex texts, including statutes and case law, which consumes a significant amount of time that could otherwise be spent on case analysis and solution development. For this reason, AI has emerged as a transformative tool capable of efficiently analysing and synthesising information in a fraction of the time traditionally needed (Conrad et al., 2023). However, legal language is highly nuanced and varies across jurisdictions and cases. This makes it difficult for general-purpose LLMs to capture the subtle distinctions necessary for accurate interpretation (Chalkidis et al., 2021), which may also translate in an increased risk of "hallucination", a phenomenon where models “fabricate information” by producing plausible but inaccurate responses (Azamfirei, Kudchadkar, and Fackler 2023).

In this context, LLMs tailored to the legal domain offer a unique advantage in interpreting and processing challenging legal language and reducing the risk of hallucinations. Yet, despite their advantages, there is a notable scarcity of LLMs specifically designed for non-English legal systems, including the European Portuguese legal framework. To enhance the model’s understanding of the natural legal language and mitigate the risk of hallucinations, it proves essential to improve model training by incorporating more relevant domain-specific

data, implementing smarter training methods that require fewer data and resources, and adapting general-purpose models to interpret contextual nuances more accurately (Wei et al., 2023).

Nevertheless, adapting LLMs to meet specific legal tasks via fine-tuning, while less costly than training models from scratch, still presents significant financial and computational challenges, particularly for smaller firms (Xia et al., 2024). As a response to these limitations, Retrieval-Augmented Generation (RAG) systems have emerged as a cost-effective solution, requiring minimal retraining to stay current with the surge of new cases and laws (Katz et al., 2020), and consistently outperforming fine-tuning alone (Ovadia et al., 2024). RAG systems achieve this by retrieving up to date, external information to contextualize generated responses for greater accuracy, as opposed to generalised or fine-tuned models that rely solely on the information it is trained on.

Although RAG emerges as a promising solution for domain adaptation, its effectiveness depends highly on component design and optimization. There are several optimization methods across its components, with one of the most important ones being chunking (Yepes et al., 2024), which involves segmenting large documents into smaller, manageable pieces.

Chunking is especially important for legal corpora, which are characterized by intricate structures and long-context texts. While prior studies have studied chunking, Wang et al. (2024) propose further research to evaluate the specific impact of different strategies on the overall performance of the RAG system. Hence, the group component addresses this gap by focusing on experimenting with various chunking techniques to create a robust baseline for a RAG framework suited for Portuguese legal texts. It provides a comprehensive overview of chunking strategies, including fixed-sized, recursive, and contextual strategies (Yepes et al., 2024), along with advanced techniques like Sliding Window and Small-To-Big (Yang, 2023). The generated

responses from these strategies are then assessed in a three-fold approach: retrieval relevance, presence of hallucination, and overall answer quality, combined with computational efficiency in embedding and generation, thus aiming to identify configurations that can best handle the complex structure, extended length, and underrepresented language of European Portuguese legal documents.

The thesis also tackles key challenges in RAG applications, particularly the persistent issue of hallucination, even in well-structured retrieval systems. To mitigate this, it emphasizes ensuring the relevance of document selection, queries, and generated answers by integrating graph-based reasoning to enhance the model's overall performance.

By addressing these challenges, it is aimed to deliver important insights to expand the capabilities of RAG systems for legal research in the Portuguese Legal Domain, empowering professionals to navigate the complexities of Portuguese legal documents with greater precision and efficiency.

2 Literature Review

2.1 A Brief Introduction to LLM History

LLMs have significantly influenced the field of NLP, introducing new capabilities for understanding human language and producing human-like generations (Touvron et al., 2023b). Defined by their large parameter counts and extensive training on diverse datasets, these models handle tasks like translation, text completion, and summarisation fluently (Radford et al., 2018). Widely recognized LLMs, such as GPT, BERT (Bidirectional Encoder Representations from Transformers), PaLM, and more recently Llama, Claude and others, are known for their ability to generalise tasks, not requiring any sort of fine-tuning or special prompting to return useful results (Wang et al., 2024).

Early Works and NLP

The early development of language models traces back to the interwar period, with the World first observing the term “translating machine”. During World War II, the first language task was solved by a human-made machine, the development of a department led by Alan Turing that discovered a way to break the German messaging cryptography. The field was later advanced after research from Noam Chomsky, with the joint effort of IBM and Georgetown University to create a machine capable of doing machine translation, this time, capturing grammar (Johri et al., 2021).

With more advancements, came the modern models, reliant on statistical techniques, such as n-grams and Hidden Markov Models (HMMs), which used probabilistic frameworks to predict word sequences based on previous sentences. These models were the groundwork to modern NLP approaches, but struggled to capture context and long-range dependencies, limiting their capability especially when dealing with longer texts (Wang et al., 2024). With the

introduction of Neural Networks, language models gained the ability to represent words in a continuous vector space. This development enabled them to capture semantic relationships between words, improving context-awareness and performance across different NLP tasks (Touvron et al., 2023a; Touvron et al., 2023b).

The Introduction of Pre-Trained Models

The transition from these earlier models to LLMs was significantly accelerated by the development of self-supervised learning techniques, which enabled developers to train their models with more data. While earlier models often relied on manually labelled datasets, constraining their scalability due to time and monetary constraints for data annotation (Wang et al., 2024), self-supervised learning addresses this limitation by allowing the model to identify patterns without human-based labels, generating a more generalized language understanding (Bommasani et al., 2022).

Shift to Transformer Architecture

Building upon these advancements, the development of the transformer architecture, introduced by Vaswani et al. (2017), provided a powerful framework for capturing context, allowing the development of models such as GPT-1, by OpenAI. The model innovates by utilising the Transformer model with generative pre-training for language tasks and then fine-tuning it for specific goals (Radford et al., 2018), generating better results than previous models and establishing a benchmark for the future, with upcoming models, such BERT and GPT-2 following a similar strategy (Wang et al., 2024).

Different from Recurrent Neural Networks (RNNs), which process the text sequentially, the Transformer model uses a self-attention mechanism, allowing it to model relationships between words in parallel. This design not only improves the efficiency of capturing context and long-range dependencies but also significantly reduces the training time while enhancing

the overall model performance (Radford et al., 2018). It also enables LLMs to handle sentences of different lengths, more complex structures, making them more suitable for larger and less curated datasets for pre-training. This architecture serves as the cornerstone for models such as GPT-3 and PaLM, benchmarks of generative language modelling (Naveed et al., 2024).

Scaling in Data, Parameters, and Computational Power

The development of LLMs has been possible due to advancements in data availability, model size, and computational power. Larger datasets provide the linguistic and contextual diversity needed for models to generalise across different tasks (Kaplan et al., 2020). Plus, increasing model parameters have shown to improve performance significantly, with diminishing returns to scale, as demonstrated by models like GPT-3, with its 175 billion parameters (Brown et al., 2020).

These improvements, however, come with great computational demands. Scaling up parameters and data simultaneously, the best approach for increasing model performance, requires high-performance hardware and extensive processing power, making LLM training both costly and resource intensive. The development of specialised hardware, such as GPUs and TPUs, has become essential in meeting these computational needs, being the latest enabler to the development of LLMs (Radford et al., 2018).

Yet, despite these advancements that have broadened the scope of LLM applications, there are still challenges in context-specific fields such as law and medicine, where the tolerance for error is lower (Touvron et al., 2023b), and with models still struggling to deliver accurate, up-to-date information on recent developments or provide reliable outputs for knowledge-intensive tasks, as their knowledge is inherently limited to the data available during training.

2.2 LLMs to RAG: The shift from traditional LLMs to RAG model

To address these challenges, researchers have been investigating approaches to improve the efficiency and capability of AI frameworks within the legal domain (Padiu et al., 2024), focusing on better adapting models to handle domain-specific language, improving reasoning capabilities, and refining retrieval mechanisms.

Lewis et al. (2020) introduced the RAG framework as a method to improve the LLM's output factual accuracy through incorporating retrieval mechanisms externally. This approach improves pre-trained language models, which traditionally rely on internally stored information (parametric memory), by granting them access to external knowledge (non-parametric memory). This framework achieves state-of-the-art performance by combining a pre-trained retriever with a generator, allowing LLMs to excel in knowledge-intensive tasks. In this system, the internal memory is managed by a pre-trained sequence-to-sequence transformer, while external memory consists of a dense vector index - often built from sources like Wikipedia - accessed through a pre-trained neural retriever.

The Evolution of the RAG Paradigm

Since its introduction, numerous research efforts have explored different RAG frameworks and architectures to optimise retrieval accuracy and reduce hallucinations. Nevertheless, the core components of the RAG paradigm - Retrieval, Generation, and Augmentation - remain consistent. Pertaining to these main components, Gao et al. (2024) categorise the evolution of RAG research into three main stages: *Naive RAG*, *Advanced RAG*, and *Modular RAG*.

The *Naive RAG* approach is presented as a more straightforward architecture. As Gao at al. describes, this system relies on a process of indexing, retrieval, and generation to integrate external knowledge into language model outputs. During indexing, unstructured data in various

formats is pre-processed into consistent text chunks that align with the input constraints of language models. These chunks are then stored in a vector database in a vectorized form after processing using an embedding model (Chen et al., 2024). During retrieval, the system calculates and compares similarity scores between the user query and the stored chunks (both in vectorized form) and identifies the top K most relevant chunks. Finally, the user query is synthesized with the retrieved chunks and are fed as inputs to the LLM to generate an output (Zhang et al., 2023; Gao et al., 2024).

While *Naive RAG* introduced significant advancements by adding context to queries and enhancing the accuracy of generated responses, several limitations have been highlighted (Gao et al.,2024). In the retrieval phase, it deals with precision and recall issues that can result in distracting or missing pieces of context, affecting the quality of the response. During the generation phase the model's response may not align with the retrieved context, making *Naive RAG* susceptible to Hallucinations and consequent production of irrelevant or biased outputs toward pre-trained knowledge. The augmentation process often encounters challenges such as fragmented or overly repetitive responses, making it difficult to maintain logical flow, relevance to the topic, and consistency in tone or meaning. Furthermore, there is a risk of over-relying on retrieved content without adding meaningful new insights, limiting the depth and originality of the generated response (Gao et al.,2024).

Advanced RAG emerged as a framework to address the limitations of *Naive RAG* by introducing optimization strategies across the RAG pipeline components. Such strategies include refining the user query and improving the indexing structure which happens in the pre-retrieval phase. This includes transforming or re-writing the query (Ma et al., 2023 & Peng et al., 2023) to make it clearer and more suitable for retrieval, as well as enhancing the granularity of the data and incorporating metadata to better the comprehension of the indexed text by the

model. In post-retrieval, key optimization strategies focus on integrating retrieved information with the query more effectively. This includes prioritizing the most closely connected context chunks through re-ranking algorithms (Xi et al., 2023), following with the compressing of the re-ranked chunk list to emphasise essential parts. This helps prevent the LLM from being overwhelmed with irrelevant details that do not contribute to the task at hand (Ilin, 2023).

Building on this, *Modular RAG* introduces even greater adaptability through a flexible modular structure composed of interchangeable modules that can be reconfigured to address specific challenges and that enable task-specific optimizations (Gao et al., 2024). Key modules include *Search*, which extends retrieval capabilities beyond vector databases by integrating sources like search engines and knowledge graphs; *Fusion*, that enhances the search process by using multiple queries to explore different perspectives; and *Memory*, leveraging the LLM's internal memory for continuous self-improvement and better query alignment. Additionally, *Routing* optimises the pathway of the query to the different RAG components, while *Predict* minimises irrelevant information and repetition by generating more focused context directly from the LLM. The *Task Adapter* module further allows customization of RAG for specific tasks, particularly in retrieval, enabling more tailored solutions across different domains. These components are customizable, and researchers have tried to develop various frameworks and toolkits to facilitate *Modular RAG* development, including LangChain, LlamaIndex, Haystack, FlashRAG (Jin et al., 2024) etc. Yet, all these frameworks share the same principle of offering greater flexibility, allowing RAG to be fine-tuned for a wide variety of tasks and use cases.

Beyond the three main research paradigms, recent studies have also focused on *GraphRAG*, an innovative framework designed to address the limitations of traditional RAG approaches in real-world applications (Hu et al., 2024). Leveraging graph-based data, such as Knowledge Graphs (KGs), it maps relationships between information through interconnected

edges and nodes that represent relationships or semantic similarities. Unlike the sequential retrieval system used in traditional RAG frameworks, *GraphRAG* enables a more efficient and targeted retrieval process by exploring datasets based on the relationships between data points (Edge et al., 2024). This method enhances retrieval precision and generation quality, delivering more accurate and contextually relevant outcomes for knowledge-intensive tasks (Peng et al., 2024).

Tailoring LLMs to Specific Domains: RAG vs Fine-tuning

Another approach to optimising LLMs, besides RAG, is fine-tuning, which involves adapting a pre-trained language model to a specific task by further training it on a smaller, task-relevant dataset (Bergmann, 2024). This process allows the model to specialise by updating its internal weights and parameters to fulfil the specific requirements of the task, effectively internalising patterns and knowledge relevant to a particular domain. In contrast, RAG combines retrieval-based methods with generative language models. Rather than relying exclusively on the knowledge embedded in the model's parameters, it retrieves relevant external information in real time and integrates into the model's response. This suggests that this dynamic retrieval process enables RAG to handle up-to-date or domain-specific queries without requiring additional retraining. As a result, it is particularly useful in cases where the model needs to access current information or specialised knowledge that the model has not been originally trained on.

Fine-tuning and RAG are often compared to each other in the realm of their performance of injecting external knowledge into LLMs (Ovadia et al., 2023 & Alghisi et al., 2024). As Ovadia et al. explain, fine-tuning is beneficial for tasks and domains that don't necessarily require constant updates, while RAG excels in knowledge-intensive tasks where real-time access to information is crucial. Nonetheless, it is noted that RAG consistently demonstrated

superior performance over unsupervised fine-tuning in various knowledge-intensive tasks spanning multiple topics. However, these approaches are complementary rather than exclusive and can be combined to enhance the optimization of LLMs, enabling both efficient real-time retrieval and customized responses. Moreover, the efficacy of both methods often relies on the specific used case and the combination of tools employed (Alghisi et al., 2024).

Applications of the RAG Architecture

RAG's application spans a wide range of activities. Fan et al. (2023) categorise them into three main categories: NLP applications, downstream tasks and domain-specific applications.

In NLP applications, RAG helps to enhance the capabilities of Q&A systems, ChatBots, and fact verification by providing external knowledge support, enabling more interactive and contextually rich exchanges with users, and judging the reliability of retrieved information (Gao et al., 2023). In downstream tasks, including personalised recommendation systems and software engineering, RAG enhances user preferences by integrating retrieval and generation processes (Lu et al., 2021), and facilitates code generation and repair by improving accuracy and efficiency in code summarization and semantic parsing tasks.

Considering the LLM's limitations to provide reliable information in knowledge-intensive domains, RAG facilitates various automation tasks across domains like Science, Finance, and more. In Science, it has been utilized for tasks such as generating molecules (Wang et al., 2022), designing proteins, and predicting molecular properties (Ma et al., 2023). In Finance, RAG can improve the accuracy of financial sentiment analysis by retrieving financial information from external sources (Zhang et al., 2023).

As RAG is an evolving architecture, its applications are actively being developed and tested across various domains. There is a growing trend of leveraging RAG to address diverse

business challenges and automate processes. However, there are limited open-source research publications that explore the feasibility, best practices, and detailed outcomes of RAG systems specifically within the legal domain. Companies like Thomson Reuters and LexisNexis, for instance, offer legal research tools powered by RAG. They claim that these systems effectively minimise Hallucinations, with LexisNexis even asserting a 100% Hallucination-free experience (Wellen, 2024). However, as these tools are proprietary, empirically assessing these claims remains challenging.

Magesh et al. (2024) evaluated these tools with a focus on factual Hallucination and found that RAG-based legal research tools, such as those from LexisNexis and Thomson Reuters, still exhibit hallucination rates ranging from 17% to 33%. In comparison, GPT-4 demonstrated a higher hallucination rate of 43%. This goes to show that RAG is a sensible solution to reducing hallucinations, especially when compared to LLMs such as GPT; however, they are not without limitations. Rather than serving as a definitive solution, RAG-based tools should be seen as valuable for expediting initial stages of legal research, though not as an absolute source of reliable information.

Evaluation methods

Evaluating RAG systems is a significant challenge due to their different components and specific application cases. Even so, an effective evaluation framework is crucial to ensure the RAG system performs reliably, especially in high-stakes domains such as legal research.

According to Yu et al. (2024), there are different evaluation frameworks that use specific metrics for both the retrieval and generation components; there are both tailored solutions to different problems and standardised versions, aiming to adapt to a wide variety of RAG applications. Popular frameworks for developing RAG systems such as LangChain or LlamaIndex, emphasise metrics such as faithfulness, to ensure generated answers are grounded

in the retrieved context; context relevance, to evaluate how well retrieved chunks and answers align with the query; and correctness, which compares generated answers to a reference or ground truth answer. Many also leverage mean-reciprocal rank and precision metrics to evaluate retrieval, and semantic similarity metrics for assessing generation quality. Community-driven RAG specific evaluation tools such as *UpTrain*, *DeepEval*, *RAGAS*, and *RAGChecker*, available on GitHub, offer additional frameworks for evaluating AI systems and RAG pipelines.

Other evaluation measures commonly used include *BLEU*, or Bilingual Evaluation Understudy, developed by Papineni et al. in 2001, and *ROUGE*, also known as Recall-Oriented Understudy for Gisting Evaluation, developed by Santhosh in 2023. *BLEU* is widely applied in machine translation tasks, where the objective is to automatically translate text from one language to another. It assesses the precision of n-grams - contiguous sequences of n words - by comparing the model's output to human-generated reference translations (Papineni et al., 2001).

ROUGE, on the other hand, is primarily focused on recall. It calculates the proportion of true positives relative to the total number of true positives and false negatives, making it particularly valuable in situations where false negatives are more critical than false positives. Frequently used in text summarization tasks, *ROUGE* measures the similarity between automatically generated summaries and reference summaries by analysing the overlap of n-grams, including unigrams, bigrams, and higher-order sequences (Santhosh, 2023).

While these tools have proven valuable across various applications, there is no universal solution that addresses all challenges, particularly in specialized domains like law. Many of these evaluation tools are limited in their compatibility with languages other than English, which is a significant concern when applying them to legal texts in languages such as European

Portuguese. To effectively assess performance in the legal domain, they often require adaptation to account for the unique linguistic, contextual, and terminological characteristics of the application case. This may involve customizing existing evaluation frameworks or developing entirely new methodologies to ensure accurate and reliable assessments of domain-specific language processing models.

2.3 AI Applications in the Legal Domain

The legal domain is inherently rooted in language, as legal institutions, actors, and processes rely heavily on the production, consumption, and interpretation of vast amounts of text (Chalkidis et al.,2021). Over the past three decades, the volume and complexity of legal texts has expanded significantly, driven by increased regulation, digitalization of legal processes, and the globalisation of law (Katz et al., 2020; Coupette et al., 2021). As a result, traditional judicial processes, characterised by time-consuming and costly procedures led by human decision-making, have increasingly struggled to keep pace with these demands (Lai et al., 2023).

In response to these challenges, recent advancements in DL and AI are being leveraged to alleviate the workload on judicial systems, improving efficiency and enhancing the quality of decision-making (Re et al., 2019). Through automated routine tasks and legal analysis support, AI has the potential to address critical shortcomings in traditional legal workflows, like long delays, high costs, and inconsistent rulings (Ejjami, 2024). However, the legal domain remains a particularly hard case for LLM specialization due to the complicated nature of the legal corpora that composes its data. Challenging structures, domain-specific language, and varying interpretation make up some of the difficulties faced by both human and machine in legal interpretation (Padiu et al, 2024).

AI and Legal Applications

Several applications have been developed for this field and the tasks they aim to address may be categorized into case retrieval, judgement prediction, document drafting, semantic annotation, and question answering, as per Padiu et al. (2024). Recent research has heavily emphasized the adaptation of transformer models for the legal domain. Chalkidis et al. (2020) investigated three approaches to tailoring BERT for legal applications: employing the standard BERT model, further pre-training it on legal-specific datasets, and pre-training a new model exclusively on legal data. Their findings reveal that domain-specific pre-training substantially improves outcomes in tasks like legal text classification and entailment. In a similar vein, Tewari (2024) introduced LegalPro-BERT, a fine-tuned model designed specifically to classify legal provisions in contracts.

One notable development is ChatLaw (Cui et al., 2024), an open-source LLM tailored for Chinese legal applications trained on a very large dataset of statutes, regulation, and case-law. This model can perform tasks such as legal consultation, case analysis, and document drafting by incorporating a legal knowledge graph to improve retrieval accuracy and contextual understanding. Other developments in the Chinese legal domain include Lawyer LLaMA (Huang et al., 2023), also trained on a large-scale Chinese legal dataset, to provide legal advice, analyse cases, and generate legal articles.

For Portuguese legal professionals, Alpaca Law (AIChatOnline, 2024) stands out as a tailored application, trained on Portuguese court and legislative data specifically designed to provide information and guidance on Portuguese law.

Legal Datasets and Benchmarks

The development of legal datasets and benchmarks plays a crucial role in advancing AI applications in the legal domain. Given the specificity of the field and consequent needs for

evaluation methods, legal applications should favour accuracy, reasoning capability, and specific language understanding to mimic those of lawyers, and as such cannot be evaluated on standard datasets. The Case Law Evaluation and Retrieval Corpus (CLERC) (Hou et al., 2024), for example, supports the identification of relevant case citations and the integration of retrieved citations into coherent legal analyses, comprising over 1.84 million federal case documents. Although models like GPT-4 achieve high scores in analysis generation, issues such as hallucination remain a challenge. The CLERC benchmark serves to advance the development of AI systems capable of effective retrieval and reasoning over legal precedents.

Another significant contribution to the field is LegalBench (Guha et al., 2023), a comprehensive benchmark for evaluating the legal reasoning capabilities of LLMs. Developed through interdisciplinary collaboration with legal professionals, LegalBench includes over 100 tasks covering six types of legal reasoning and was evaluated on 20 open-source and commercial models. LexGLUE (Chalkidis et al., 2021), another important benchmark, evaluates language models on tasks such as case classification, judgment prediction, and legal entailment using seven datasets. This benchmark highlights the challenges of adapting general models to the legal domain and highlights the necessity of domain-specific training for optimal performance.

These advancements demonstrate the growing importance of tailoring language models and benchmarks not only to the unique requirements of the legal domain, but also to specific use cases, may they be related to the task itself or not. By addressing challenges such as domain-specific reasoning, handling long documents, and improving retrieval capabilities, these studies pave the way for the effective integration of AI into legal practice.

Real-World AI Applications in Law

Some AI applications are already being integrated into real-world applications. One notable implementation of AI in this context is the Harm Assessment Risk Tool (HART), a predictive tool employed by law enforcement in the United Kingdom in custody-related decisions that predict the likelihood that an individual will commit a grave offence within the following years (Greenstein, 2022). Similarly, in China, the use of AI has advanced with the creation of "Internet courts" (Vasdani, 2020), where legal cases are decided online, offering faster, more accessible justice. Elsewhere, the use of AI in the legal domain has sparked debate, as seen in a recent case in Colombia where a Judge used ChatGPT to support his ruling on a child's medical care (Aydin, 2023). While the ruling itself was not disputed, the use of AI in judicial decision-making triggered controversy, highlighting both the opportunities and challenges AI presents to modern legal systems.

As observed, AI is increasingly transforming legal processes, improving efficiency and access to justice through specialized models and real-world applications. Despite significant advancements, challenges in accuracy and ethical considerations remain central to the future of AI in law.

2.4 Regulatory Considerations in Legal AI

2.4.1 Introduction to Data Privacy and Security in Legal AI

In the legal field, privacy and security are crucial elements as the field handles sensitive and highly confidential information. Client/lawyer relationship hinges on confidentiality, a "fundamental principle of justice", as per the Council of Bars and Law Societies of Europe. While applications of AI in the legal system offer efficiency, they also present unique challenges, especially regarding confidentiality. RAG systems, designed to enhance LLM performance by obtaining pertinent information from outside sources prior to producing a

response, introduce potential vulnerabilities. These systems can expose sensitive legal information, making them susceptible to cyberattacks or accidental access, which could undermine both confidentiality and trust (Bruckhaus, 2024). For instance, a cybersecurity breach could expose sensitive legal information, undermining both confidentiality and trust.

2.4.2 Legal and Regulatory Frameworks Governing Privacy

The demand for strong regulatory frameworks has increased due to the quick adoption of AI technologies in high-stakes industries like law. By addressing privacy, security, and ethical issues, these frameworks make sure that the use of AI systems complies with social norms and legal requirements (European Parliamentary Research Service, 2020).

In 2018, the European Council introduced the GDPR, a framework for data protection rules. Aimed at controlling the collection and processing of personal data, it seeks to safeguard individual privacy rights within the EU. By establishing strict guidelines for the collection, processing, and storage of personal data, GDPR ensures that individuals retain control over their personal information (Council of the European Union). The flexibility of GDPR's provisions encourages innovation in AI applications. However, this adaptability also introduces ambiguities, particularly regarding compliance in AI systems like RAG, which operate at the intersection of data retrieval and processing (European Parliamentary Research Service, 2020).

Building on GDPR's foundation of data protection, the European Union has introduced the EU AI Act, the first comprehensive regulation designed specifically to govern the ethical and safe deployment of AI systems (European Parliament, 2024). The AI Act takes a risk-based approach, meaning that the level of regulatory scrutiny intensifies with the level of risk the AI system poses (Die Bundesregierung, 2024).

The GDPR and EU AI Act enhance one another. While GDPR establishes baseline privacy protections, the AI Act extends these safeguards by addressing the ethical implications

of AI deployment, particularly in high-risk sectors. National bodies like Portugal's CNPD operationalize these frameworks, ensuring local compliance through audits, enforcement actions, and guidance for legal AI implementations (Comissão Nacional de Proteção de Dados, n.d.). This combination of frameworks provides a comprehensive approach to managing the intersection of AI and privacy, forming the foundation for enforceable security measures that will be discussed in the following section.

2.4.3 Security Implications in Legal AI

Data used in the legal domain is inherently sensitive, encompassing personal information, confidential case files, and attorney-client communications, making data security of utmost importance. LLMs, which rely on large datasets and complex models, introduce new risks, such as data leakage. This complexity can lead to legally incorrect results or errors due to outdated or unverified information, presenting significant challenges to data integrity and reliability in legal applications (Jegorova, 2022). Additionally, these risks can result in the unauthorized disclosure of sensitive data and expose the system to adversarial attacks, where attackers manipulate input data to reveal confidential information (Wallace et al., 2021).

This emphasizes the importance of robust data processing protocols, encryption, and regular model checking, to safeguard customer privacy and preserve integrity of legitimate AI results.

Ensuring Accurate AI Outputs in Legal Systems

The presence of hallucinations highlights the necessity for robust validation protocols to minimize errors in high-stakes legal decisions, as they pose serious risks to legal outcomes. A 2024 Stanford study found that 1 out of 6 Legal Models hallucinate in benchmarking queries, stressing the prevalence of this risk (Magesh et al., 2024) and the need for precaution given that

LLMs rely on statistical patterns, generating text without understanding or validating information (Lewis et al., 2021).

Mitigation Strategies include human oversight and model audits. Human oversight, also called the human-in-the-loop approach, where humans are permitted to intervene at every stage of the model (Cousineau, 2024) permits those with deep domain knowledge, to detect errors in AI outputs. As discussed in this paper before, the legal domain is extremely nuanced and could be easily misinterpreted without human intervention, which can provide context, critical thinking and judgement to compensate for AI's limitations (Dienstein et al., 2024).

Transparency is also critical in maintaining trust in AI systems. Many AI models, particularly those built on DL architectures, operate as “black boxes”, making it difficult to interpret or justify their outputs (Bathaee, 2018, pp. 890-893). This lack of transparency risks undermining the legal profession's foundational principles of reasoning and accountability. Mechanisms such as audit trails and explainability requirements can address these challenges, providing clear rationales for AI-generated decisions and ensuring that legal professionals can validate outputs (Doshi-Velez, 2017; Wachter, 2017).

2.4.4 Bias in AI Models

Bias is a critical risk in AI, especially when it comes to ensuring fairness in high-stakes decisions, such as legal sentencing and hiring. The potential for bias to impact outcomes that affect individuals’ lives and rights demands careful examination and mitigation. Implementing an LLM with or without RAG systems can replicate biases from training data or introduce new ones. This issue becomes particularly concerning when AI is used in high-risk areas such as judicial decision-making, labour law procedures, and medical diagnosis, when people's lives and rights may be significantly impacted by these biases (Bogen et al., 2019). Eliminating AI

bias in the legal field is therefore not just a technical issue, but a key step toward ensuring justice and equality before the law.

Algorithmic bias, defined as systematic errors leading to discriminatory outputs based on protected characteristics, reflects the EU's understanding of bias as differential treatment (Amann et al., 2020; European Union Agency for Fundamental Rights, 2022). The 2016 State v. Loomis case illustrates algorithmic bias, with the Wisconsin Supreme Court ruling that using AI for sentencing did not violate due process, highlighting fairness and transparency challenges. This case accentuates the challenges of balancing fairness and transparency in legal AI systems, particularly when they impact judicial decisions.

Bias in legal texts, such as stereotypical language or unequal case outcomes, can have serious consequences in AI applications. A 2010 study by Boyed et al., found that male judges are statistically less likely to favour parties alleging discrimination unless they were seated with a female colleague.

Fairness in AI

Fairness is closely tied to ethical principles such as transparency and accountability. A lack of transparency, especially in “black box” models, raises ethical concerns. Decisions must be justifiable and interpretable to uphold legal reasoning. Ensuring accountability for AI outputs, whether through lawyers, developers, or organisations, is critical to preventing misuse or errors that undermine trust in legal AI systems (Doshi-Velez, 2017; Wachter, 2017).

In an ethical application of AI in the legal domain, bias and fairness are deeply intertwined. Addressing bias and ensuring fairness in AI models helps maintain public trust, uphold justice, and support equality in high-stakes legal decisions (Alvarez et al., 2024).

3. Methodology

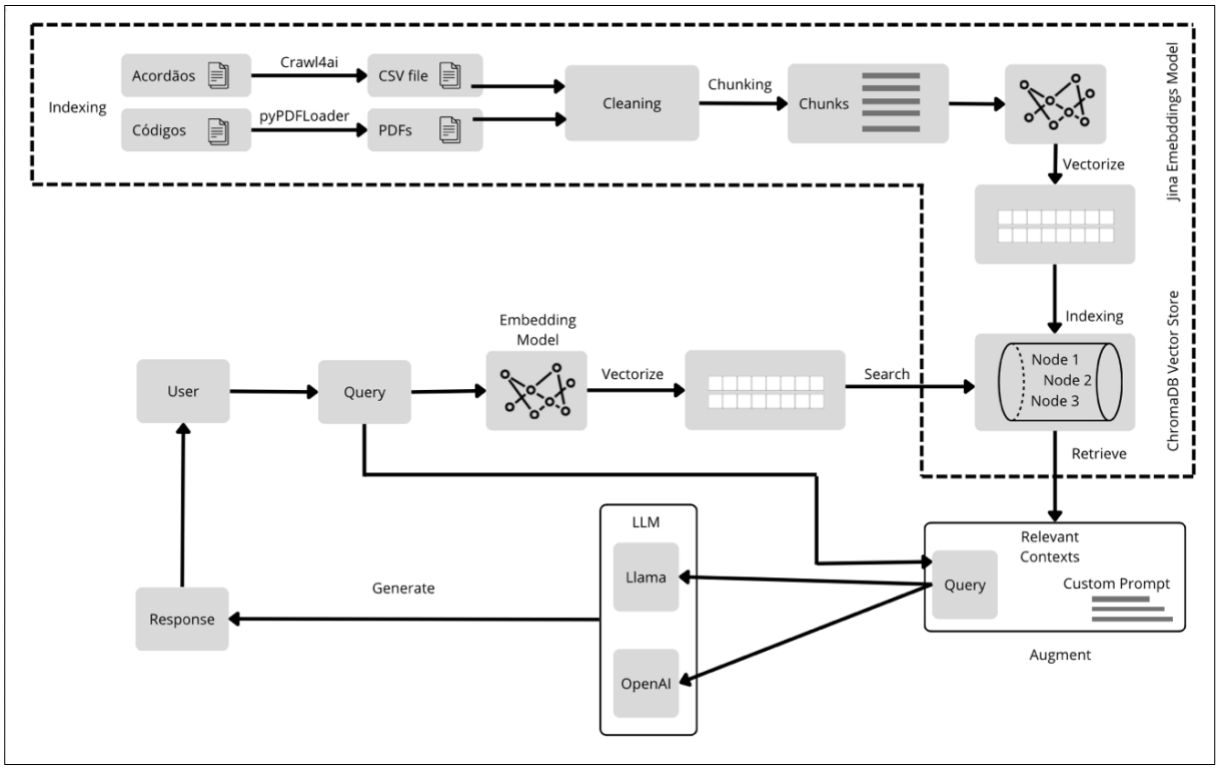

Figure 1 - *Methodology to implement the RAG infrastructure*

This section outlines the methodology employed in the development of the RAG application, with a focus on leveraging LangChain as the core framework to streamline the process. Initially, it describes the processes for data collection, including the sources and selection criteria, and details the methods for cleaning and pre-processing the data to ensure quality and consistency. Next, the exploratory strategy for chunking strategies is presented, highlighting the rationale behind the chosen approaches given the aim of this study, as well as an overview on embedding models and vector stores. In the sections that follow, the decisions regarding the construction of the RAG infrastructure are detailed, to form a baseline model to evaluate chunking techniques. Finally, the evaluation strategy employed is presented, highlighting the criteria and variations considered in assessing the system's overall performance.

3.1 Framework

The study leverages LangChain, an open-source framework that simplifies the development, production, and deployment of applications using LLMs. Specifically, it provides essential tools for building RAG applications, including functionalities for chunking, embedding, retrieval, and other advanced features (LangChain, 2024).

Throughout the development of this thesis, LangChain has changed and updated its packages and versions of the framework, also deprecating some functions in a near future. This work uses the v0.3 version that was last updated in September of 2024.

3.2 Data Gathering & Constructing a Database

3.2.1 Data Sources and Collection

With the aim of constructing a system to assist lawyers in legal research, it was essential to simulate an environment reflective of the types of documents traditionally used in this context. To achieve this, a small-scale system focusing on labour law was developed. The dataset comprises ten (10) "acórdãos" - judicial decisions that include a description of the case, the reasoning underpinning the decision, and the final ruling - all relative to Labour Legislation and selected from the "Tribunal da Relação de Lisboa", the second-instance court from the Lisbon region, thus ensuring their relevance to the Portuguese legal context.

In addition to the "acórdãos", the dataset includes the primary Portuguese legal codes frequently referenced in these decisions, specifically the "Código do Trabalho", "Código do Processo Civil", and "Código do Processo de Trabalho". These materials constitute the non-parametric knowledge base of the RAG system.

Focusing on labour law provides a cohesive thematic scope, which is instrumental in developing an evaluation framework. This approach enables cross-referencing between

documents, ensuring that the system retrieves information that is accurate, relevant, and contextually interconnected.

The “acórdãos” were collected using Crawl4AI, an open-source Python library specifically designed for web crawling which scrapes them into markdown format, and then saved into a CSV file (Crawl4AI, 2024). In contrast, the “Códigos” were obtained as PDF files from the official Portuguese source - “Diário da República”. Due to the differences in source formats, the uploading processes also differ: the “Códigos” are processed using LangChain's *PyPDFLoader*, while the “acórdãos” are uploaded from the CSV file, generated during the scraping process. This CSV includes metadata such as the court case name, date, link, responsible judge, decision, vote, descriptors (words or short sentences that resume the main themes or questions being addressed in a decision) or keywords, and a summary of the decision, whereas the PDFs only contain the source of the document, as well as the page the chunk belongs to.

3.2.2 Cleaning Techniques

Text preprocessing is widely recognized as a critical step in NLP downstream tasks, as emphasized by Haviana, Mulyono, and Badie’ah (2023). This area has seen notable advancements in recent years, particularly with the emergence of transformer models, which have ushered in a new era of text analysis (Ridoy et al., 2024).

As per Shabbir (2023), unprocessed text data can be compared to “a chaotic jigsaw puzzle”, where cleaning serves as the process of assembling the pieces, discarding those that don’t belong, and refining the rest to create a clear and coherent picture. Thus, cleaning enhances the data quality, improves readability for the model and ensures consistency to allow for easier learning (Shabbir, 2023).

The most common cleaning methods consist of lowercasing the data, removing punctuations, numbers, extra spaces, emojis, emoticons, contractions and special characters in general, and replacing the repetitions of punctuations. By installing the right libraries, one can easily go through these steps, but when it comes to domain-specific data and a less-represented language that not many models are trained on, further consideration must be given to each traditional step to ensure it is sensibly applied.

The Portuguese Language

Portuguese, the fifth most spoken language globally (Diplomatic Portal, 2024), presents unique linguistic features that demand specialized approaches for effective language processing. It is the official language of several countries, including Portugal, Brazil, and Mozambique, but significant differences exist between Brazilian and European Portuguese. These variations span pronunciation, verb conjugation, and syntax. For example, in Brazilian Portuguese, object pronouns are commonly placed before the verb, resembling Spanish structures, while in European Portuguese, they typically follow the verb (Posner et al., 2024).

Additionally, Portuguese grammar includes distinctive features such as the "infinitivo pessoal" (personal infinitive), which allows an infinitive to indicate its subject. This characteristic, along with regional dialectal differences, poses challenges for traditional linguistic processing techniques that rely on rigid grammatical structures or standardized vocabulary.

These complexities highlight the importance of developing tailored language cleaning and processing methodologies to address the unique requirements of Portuguese, particularly for tasks involving European Portuguese, where existing tools may fall short in handling its intricate grammatical and syntactic nuances.

Portuguese Language Preprocessing

For cleaning the Portuguese text, this thesis emphasises maintaining the original structure of the documents to ensure generalisation and minimising overfitting to specific cases. Lemmatization, a process that reduces words to their base or root form, was avoided due to the unique challenges of the Portuguese language. For example, in Portuguese, many words and verb conjugations share similar structures but convey entirely different meanings depending on their context, which could hinder the model's ability to fully comprehend the text.

Given the considerable variation in structure across “acórdãos” and other legal texts, even within the same court, the text cleaning was kept minimal. For “acórdãos”, the cleaning process involved simple techniques such as lowercasing, removing image links, excess spaces, newlines, and markdown artefacts which were found to not have an emphasising effect. By contrast, the “Códigos”, extracted from PDFs using LangChain, lacked markdown artefacts and required less preprocessing. This streamlined approach was chosen to maintain consistency across the dataset and facilitate future scalability, while addressing the complexities of the Portuguese language.

3.2.3 Chunking

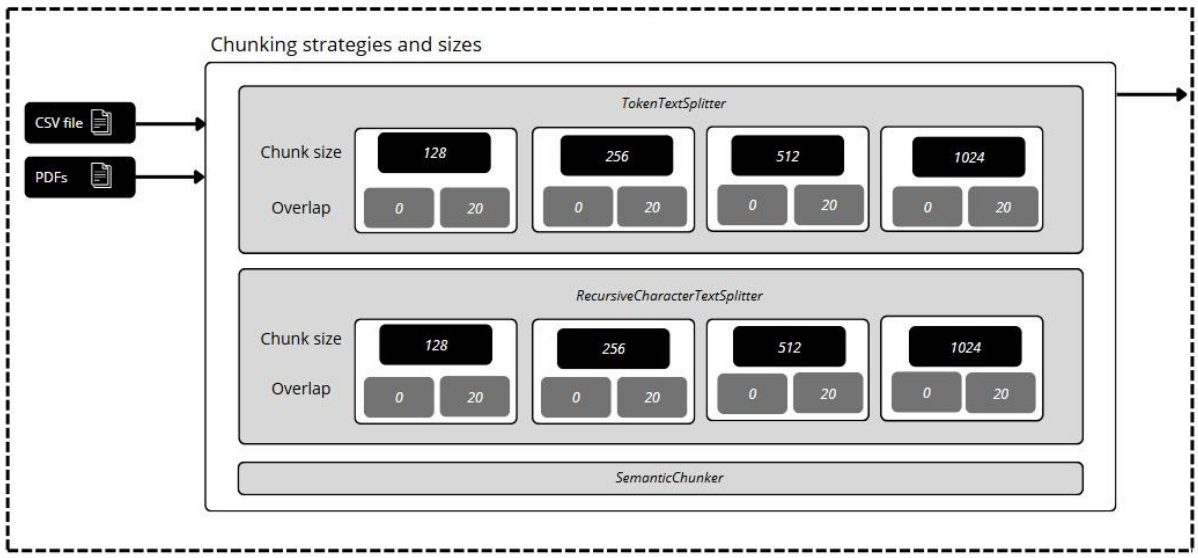

Figure 2 - *Chunking strategy pairs employed in the study.*

A notable limitation of LLMs is their restricted contextual window, which hinders their ability to process lengthy documents such as legal texts in their entirety. To address this, large texts are segmented into smaller, manageable portions, allowing LLMs to analyse each segment more effectively. This approach, known as chunking, enhances the model's ability to deliver accurate and coherent insights across the document (Yepes et al., 2024).

By grouping together information into ideally sized pieces, LLMs can efficiently process and produce the desired outcomes. Whenever the RAG system receives a question, it will search for contextually relevant chunks and ground the generated answer on the retrieved chunks. Moreover, chunking can also be applied to quickly grasp the key concepts of a document, such as a book or journal article. If the text can be clustered into semantically coherent groups and then each cluster summarised in some way, this can help speed up the time to insights (Martin-Short, 2024).

The following section introduces the chunking techniques employed in this thesis, which represent key strategies outlined by Yepes et al. (2024): fixed-size, recursive, and

contextual chunking. These methods are designed to segment text while preserving essential metadata, as described in Section 3.2.1, ensuring that each chunk remains linked to its original document properties.

Fixed-Size Strategy - *TokenTextSplitter*

TokenTextSplitter is a chunking technique from LangChain which splits a raw text string based on the number of tokens rather than characters or sentences. It first tokenizes the text into Byte-Pair Encoding (BPE) tokens, a compression-based algorithm, and then splits these tokens into chunks of a predefined size. Finally, the tokens within each chunk are reconstructed back into text, ensuring the chunks are manageable while retaining the original meaning.

Unlike character-based or semantic chunking, which may overlook the actual token count, *TokenTextSplitter* breaks down text precisely by the number of tokens. This advantageously prevents token overflow errors, which can disrupt processing when input exceeds the model's maximum token capacity. However, it may overlook the underlying context and affect the response generation (Yepes et al., 2024).

Recursive Strategy - *RecursiveCharacterTextSplitter*

The *RecursiveCharacterTextSplitter* is typically seen as the go-to first algorithm for many applications within LangChain. It divides large text into smaller, manageable chunks through a hierarchical process using a predefined or customizable set of separators. The algorithm starts with broader splits, such as paragraphs, and progressively uses finer separators, like commas, until the chunks meet the specified size.

This approach may be beneficial in legal or other structured document analysis, allowing for the preservation of contextual relevance within each chunk while maximising the readability and traceability of each chunk.

Contextual Strategy - *SemanticChunker*

SemanticChunker, from LangChain, focuses on preserving the relationships within the text by creating semantically meaningful chunks. It uses cosine similarity between embeddings of consecutive text segments to ensure each chunk captures cohesive and contextually complete information. This method may be particularly beneficial for applications requiring a nuanced understanding, such as legal or scientific document analysis, where maintaining context and logical flow is crucial.

One critical setback, however, is the increased processing time compared to the previous chunking strategy. Moreover, there is no guarantee that the chunks will be semantically meaningful or even complete sentences (Qu et al., 2024).

Additional Dimensions in Consideration

Chunk size plays the primary role in determining the contextual window for processing and understanding text. If the chunks are too small, they may lack sufficient context, while excessively large chunks risk including irrelevant information, diminishing their utility (Pipitone and Alami, 2024). To address this, particularly in the case of legal texts, it is essential to explore different chunk sizes in the strategies where it is applicable (fixed-size and recursive).

Advanced chunking methods that will be discussed include the Sliding Window approach, which introduces overlaps between chunks to better preserve context, and the Small-To-Big strategy (Yang, 2023) that involves retrieving information using smaller sized chunk embeddings and subsequently contextualising it with larger chunks for improved accuracy, assessing its potential application given the results of previous evaluations.

Therefore, *TokenTextSplitter* and *RecursiveCharacterTextSplitter* strategies will be implemented using four (4) chunk sizes: 128, 256, 512, and 1024 tokens, each size will then be tested both without overlap and with an overlap of 20 tokens. While larger overlaps would

ideally be used for larger chunk sizes to preserve context across chunks, a uniform overlap of 20 tokens will be employed for all configurations to maintain consistency, thereby implementing the Sliding Window hypothesis.

3.2.4 Embeddings & Storage

Upon chunking, the following step concerns the translation of the generated chunks into machine-consumable representations. This corresponds to transforming the chunks into embeddings, which "are representations of values or objects like text, images, and audio that are designed to be consumed by machine learning models and semantic search algorithms" (Cloudflare, 2024). As machines need assurance with how to deal with words, the Word Embedding Technique is based on converting words to mathematical, high-dimensional space, fixed-size vector representations in a way that NLP and ML models can effectively comprehend, converting semantic relationships between documents into spatial relationships between vectors (Sturua et al., 2024).

With this said, embeddings offer numerous advantages. They reduce dimensionality, make computations more efficient, and help models generalise better (Harsoor, 2024). Additionally, they also place words with similar meanings, creating together a semantic relationship, e.g. 'king' and 'queen' would be closer together than 'king' and 'apple'. Moreover, nowadays, modern embeddings like BERT or GPT also take into account the context in which the word was used, allowing the same word to have different embeddings depending on its usage in a sentence.

Embedding models face a significant dual challenge when applied to Portuguese legal documents: handling a less-represented language and accommodating the long-context windows that characterise legal texts. To address the Portuguese language, various multilingual models have been developed by extending traditional training, typically conducted in English,

with datasets in other widely spoken languages. Notable examples include *Sentence Transformers*, which offer four multilingual variations, as well as *XLM-RoBERTa* (Conneau et al., 2019), a multilingual adaptation of the RoBERTa model that supports Portuguese.

Despite their utility, these multilingual models present a critical shortcoming: the adaptation process often results in lower maximum sequence lengths. For instance, Sentence Transformer and Bert models are limited to sequence lengths of 128 tokens, while XLM-RoBERTa supports up to 512 tokens, meaning that larger sequences get truncated and therefore not embedded. This imposes constraints not only on the range of chunk sizes that can be effectively tested, but also on the chunking strategies employed. Semantic chunking cannot reliably ensure that token counts remain below these thresholds, potentially leading to the loss of meaningful context in the document.

To address these limitations and enable the study of chunking methods, *jina-embeddings-v3* was selected as the embedding model for this study since it supports larger sequence lengths, making it particularly well-suited for handling the extensive and highly structured nature of legal documents. This feature allows for exploring a broader range of chunking strategies without the risk of exceeding token limits.

Jina Embedding Model

The *jina-embeddings*-v3 is a cutting-edge multilingual text embedding model based on the XLM-RoBERTa model, with 570M parameters and 8192 token-length, outperforming the latest proprietary embeddings from OpenAI and Cohere on MTEB, ranking 2nd place on the MTEB English leaderboard for models with fewer than 1 billion parameters (Jina, 2024). It's larger than *jina-embeddings-v2*, but significantly smaller than embedding models fine-tuned from LLMs (Sturua et al., 2024).

This model achieves state-of-the-art performance on multilingual data and long-context retrieval tasks (Sturua et al., 2024), and features task-specific Low-Rank Adaptation (LoRA) adapters enabling it to generate high-quality embeddings for various tasks including query-document retrieval, clustering, classification, and text matching. It includes four key tasks and offers five adapters to cater to different use cases, including embedding queries and passages in asymmetric retrieval scenarios, classification tasks, and for tasks involving semantic similarity, such as STS or symmetric retrieval (Jina, 2024).

The large advantages of this model for this thesis are that it both supports 89 languages in total, with Portuguese being included within the thirty with best performance, and it supports sequence lengths up to 8192 tokens as well. Although this is a paid alternative, its promising claims stand out and its price offering for smaller companies may be an attractive selling point (Jina, 2024).

3.2.5 ChromaDB

A vector database is designed to store, index, and query high-dimensional vector data with efficiency. Unlike traditional relational databases that structure information into rows and columns, vector databases emphasize the geometric properties of the stored data. This enables them to perform tasks such as similarity searches and distance calculations more effectively by utilizing the mathematical characteristics of vector spaces (Nidhiworah, 2024; Ramprasad and Sivakumar, 2024).

Currently, there are many types of these structures being built, and they are also becoming more specific between themselves, from being open-source or commercial, to being dedicated vector databases or databases that support vector search. In 2024, the five best vector databases to consider are: Pinecone, MongoDB, Milvus, Chroma and Weaviate (Orr, 2024).

For the purpose of this study, the Chroma vector store was chosen as it's an open-source vector database designed for the efficient storage and retrieval of vector embeddings generated from rhetorical or textual data (Chroma, 2024), which is integrated into the LangChain framework. This way, the vector database will then contain a collection for each of the chunking strategy pair embeddings.

Although Pinecone can also connect with LangChain allowing for real-time and content-based searching, there are concerns regarding its pricing structure, particularly for large-scale deployments with significant data volumes, and advanced querying capabilities. On the other hand, Chroma allows for extensible querying, including complex range searches and combinations of vector attributes, and benefits from being open-source and having a growing community of developers contributing to its improvement and addressing issues (Woyera, 2023).

3.3 Selection of the LLM

The selection of an appropriate model for this thesis involves navigating trade-offs between computational capacity, model output quality, deployment strategy, and whether to operate the model locally or via an API. These decisions are critical in the context of the legal domain, where the sensitivity of data and the practical constraints of implementation must be carefully addressed. The selection of the model is also constrained by the reduced number of options available for Portuguese, as opposed to models suitable for the English language.

With that said, a key decision in model selection lies in determining whether the model should be run locally or accessed via an external API. Each approach offers distinct benefits and challenges that must be weighed in relation to the project's goals. Running the model locally entails processing data within a controlled environment, which offers a significant

advantage in terms of data privacy, as information remains within the user's infrastructure without being transmitted to external servers. In domains like law, where data confidentiality is a key priority, this can be a critical consideration. For the specific use case of this thesis, the legal data utilized is derived from publicly available sources, reducing the need for local processing to protect confidentiality. While local deployment ensures control over data, it imposes constraints on the size and complexity of the model that can be employed. This is due to the significant computational requirements of large-scale models, which often exceed the capabilities of standard hardware. Additionally, maintaining the infrastructure necessary to support such models can be resource-intensive, requiring both financial and technical investments that may not be justified for public data processing.

Despite these limitations, local deployment becomes highly advantageous in scenarios involving law firms or institutions that wish to integrate classified or proprietary data into their systems. In such cases, the benefits of enhanced privacy and control outweigh the challenges, making local deployment the preferred option. Conversely, deploying the model via an API provides a scalable and resource-efficient alternative. By leveraging the computational infrastructure of external providers, this approach allows access to advanced models without requiring significant local hardware investments. This can enable the use of state-of-the-art models that would otherwise be infeasible to run locally.

Nonetheless, reliance on APIs introduces concerns about data security and compliance, particularly when dealing with sensitive or jurisdiction-specific legal data. For this thesis, these concerns are mitigated by the public nature of the data, making API-based deployment a practical and effective choice. Moreover, API-based solutions can facilitate rapid prototyping and experimentation, which align well with the research objectives.

The decision between local and API-based deployment ultimately hinges on the specific requirements of the use case. For research purposes, both a local and API-sourced model were selected, and their results will be compared.**3.3.1 Llama**

According to Hugging Face, the Llama collection comprises multilingual LLMs that are pretrained and instruction-tuned generative models supporting a wide range of languages, including Portuguese. These models are optimized for multilingual dialogue use cases, such as agentic retrieval and summarization tasks, and often surpass many open-source and proprietary chat models in performance on standard industry benchmarks. It's an autoregressive language model that uses an optimized transformer architecture, often intended for use cases like commercial and research use in multiple languages. Its capabilities make it well-suited for assistant-like chat functionalities and agentic tasks such as knowledge retrieval and summarization, aligning with the objectives of this thesis.

Among the models available, the *Meta-LLaMA/LLaMA-3.1-8B* model from Hugging Face stands out for its balance between lightweight requirements and robust reasoning capabilities (Meta, 2024), as well as its capacity for the Portuguese language, and therefore is the first selection for a local model.

3.3.2 OpenAI

The *GPT-4o Mini* is OpenAI's most cost-effective small model, designed to make applications more affordable while outperforming *GPT-4* in chat preferences on Chatbot Arena (formerly LMSYS leaderboard). It is over 60% cheaper than *GPT-3.5 Turbo*, with pricing set at $0.15 per million input tokens and $0.60 per million output tokens (OpenAI, 2024).

This model is particularly suited for applications requiring multiple models calls in sequence or parallel, processing large volumes of context, or providing fast, real-time text interactions with users. Supporting both text and vision via API, it offers a context window of

up to 128K tokens and a maximum output of 16K tokens, with knowledge updated to October 2023 (OpenAI, 2024). These features made it an ideal choice for API integration in the context of this thesis.

3.4. Evaluation

This section outlines the standardized evaluation process developed to assess various RAG configurations, detailing the construction of an evaluation dataset, the evaluation criteria, and their applications. The evaluation covers the distinct collections of embeddings stored in the retriever, each representing a unique combination of chunking strategy, and, where applicable, chunk size and overlap as detailed in Section 3.2.3.

The evaluation framework is designed to systematically measure both computational efficiency and performance at different stages of the RAG process. The evaluation will pertain 2 evaluation moments: the first corresponding to the embedding of documents into the vector store for each chunking strategy pair, measuring computational efficiency; and the second to the deployment of the RAG to generate answers.

To ensure a comprehensive evaluation in the second half, each collection is tested for retrieval and generation performance, alongside computational efficiency, to determine the most effective chunking configurations for the task. To approach this, LLMs were leveraged to judge the generations and retrieval. This iterative evaluation process promotes the development of a robust, efficient, and high-performing system tailored to the unique requirements of the legal domain.

3.4.1 Question Dataset for Evaluation

This unified evaluation process begins with creating a question dataset tailored to the documents embedded in the database collections. Constructing this dataset presents challenges due to limited domain-specific expertise, such as determining what questions accurately reflect

those a lawyer might ask, identifying correct answers, and pinpointing the relevant sections of text they rely on. To address these challenges, a more advanced LLM, ChatGPT (specifically *GPT-4o*), was employed to generate questions systematically, which were subsequently proofread and validated by a lawyer to ensure domain accuracy.

Each "acórdão" that composes the database was uploaded as an attachment, with a custom prompt crafted to generate questions relevant to its legal context (Appendix 1). For legislative codes, individual articles were uploaded, and a specific prompt was created to generate targeted questions (Appendix 2). The prompts emphasized key aspects: adopting the role of a legal advisor, posing analytical questions focused on legal interpretation, and addressing the implications of the legislation. To generate a robust dataset, the prompt explicitly required the generation of two types of questions: single-hop and multi-hop. Single-hop questions - those requiring answers directly from the text or via a single reasoning step - test a model's ability to retrieve explicit information and handle straightforward reasoning. Multi-hop questions, on the other hand, involve multiple pieces of context or reasoning steps, capturing the complexity and interconnected nature of legal reasoning. This type of question is shown to evaluate a model's ability to synthesize information across documents or disparate sections of text (Khashabi et al, 2018), reflecting real-world scenarios lawyers face when interpreting laws and cases, although expected outcomes in responses are worse than those of human performance (Welbl et al., 2018).

The final dataset consists of twenty-six (26) questions in Portuguese split equally between single-hop and multi-hop types, alongside reference answers, the context necessary for the response, as well as the type of question. This structure ensures a comprehensive evaluation framework that tests both retrieval accuracy and reasoning depth. Single-hop questions provide a baseline for assessing a model's ability to handle direct queries, while multi-hop questions

challenge the model to integrate and synthesize information, reflecting the intricate reasoning often required in legal contexts. By combining these question types, the evaluation aims to benchmark RAG models effectively against the dual demands of precision and complexity inherent in legal tasks.

3.4.2 Evaluation Criteria

As discussed above, the evaluation comprises two distinct moments: embedding into the vector store, and the generation of answers to the crafted question dataset. Embedding time will become increasingly relevant when building a database more comprehensive than the one used for research in this thesis and therefore is also important in determining the best configurations. Portuguese Courts contain hundreds of thousands of legal documents, of which a small sample of the universe was selected.

In the first moment, the evaluation of embedding times will simply consider the time taken for each chunking strategy and the respective number of chunks created. In the second moment, performance of the model itself will be assessed. Based on the evaluation criteria proposed by frameworks such as LangChain, LlamaIndex, and referencing works in section 2.3, the assessment of each RAG variation follows a three-fold structured approach: retrieval quality, generation quality, and computational efficiency. These dimensions aim to provide a thorough evaluation of the effectiveness and suitability of each configuration.

Retrieval Evaluation

The evaluation of retrieval quality focuses on the ability of the RAG system to retrieve context chunks containing the necessary information to answer a given question. Following the standard practice in using the Chroma vector as a retriever, each retrieval step provides four context chunks for analysis. This assessment is performed using an LLM, specifically *gpt-4o-mini*, with a tailored prompt (Appendix 3) designed to determine whether the retrieved set

includes the chunks most relevant to the question. This results in scores of 0, if not present at all; 0.5 if some information is included; and 1, if fully present. This approach ensures that the subsequent generation step is grounded in accurate and relevant information. An average score for retrieval quality is computed across all questions for each collection, with separate evaluations conducted for single-hop and multi-hop questions. This allows for nuanced insights into how the retrieval mechanism performs under varying levels of question complexity.

Generation Evaluation

Generation quality is evaluated in two (2) distinct ways: checking the generated answers against the reference answer and checking for hallucination. To achieve this, once more prompts were crafted (Appendix 4 & 5) to guide the LLM in determining the alignment between the generated answers and reference answers, and whether the generated answer is grounded on the retrieved context, respectively. This grounding check is vital, as hallucinations undermine the reliability of the model, particularly in high-stakes domains like legal analysis. The quality of the generated answers is scored for each question from 0 to 5 by the LLM, and an average score is calculated for each collection, adapted from a 0 to 1 scale and again distinguishing between single-hop and multi-hop questions to analyse performance across question types.

Computational Efficiency

Computational efficiency of generated answers captures the time elapsed during the RAG pipeline's response generation for each question. Efficiency is essential in practical applications, where the speed of response generation can impact usability and scalability. For each RAG configuration, the average time taken to answer the questions is recorded, enabling comparisons between variations. While computational efficiency depends significantly on the

choice of LLM and the underlying system infrastructure, it provides a useful relative measure when evaluating different configurations.

Additional Criteria & Factors

Additional measures to have in consideration include considering the number of times the model fails to reply, as it may give insights into the performance of the dimensions evaluated. It is acknowledged that the measures employed, particularly those relying on LLM judgments, involve a degree of subjectivity. This subjectivity means that the evaluation results should be interpreted as relative comparisons rather than absolute metrics. This relativity demonstrates the importance of focusing on the comparative strengths and weaknesses of different configurations within the same experimental context.

In summary, the evaluation framework leverages LLMs for detailed assessments of retrieval and generation quality while integrating computational efficiency as a practical consideration. By distinguishing between single-hop and multi-hop questions, between local and API-run models, taking into consideration the different strategies, chunk sizes, and presence of overlap, and accounting for the subjective nature of some measures, the evaluation provides meaningful insights into the performance of various RAG configurations, tailored to the demands of processing long and structured legal documents.

3.5 RAG Pipeline

To run this experiment, a simple form of the RAG was used, that simply uses the core components of the RAG infrastructure - retrieval and generation - allied to a prompt. The prompt was designed to place the LLM as a Portuguese Legal Expert in providing answers (Appendix 6), and for the retriever, it uses simply the vector store with each collection of strategy pair embeddings. This standard method returns 4 documents and searches for similarities.

4. Results & Discussion

This section provides a detailed account of the experimental results. It begins by evaluating the outcomes of various pairs of chunking strategies, analyzed individually according to the different assessment criteria. Following this, a discussion of the aggregated results is presented, along with potential recommendations based on the findings. Additional tables and calculations can be found on the supplemental Excel attached to the work.

4.1 Chunking and Embedding

These results (Appendix 7) are a measure of the time taken to chunk and embed under a determined strategy pair. From a scalability standpoint, this stands as an important criteria given the increase in volume of information.

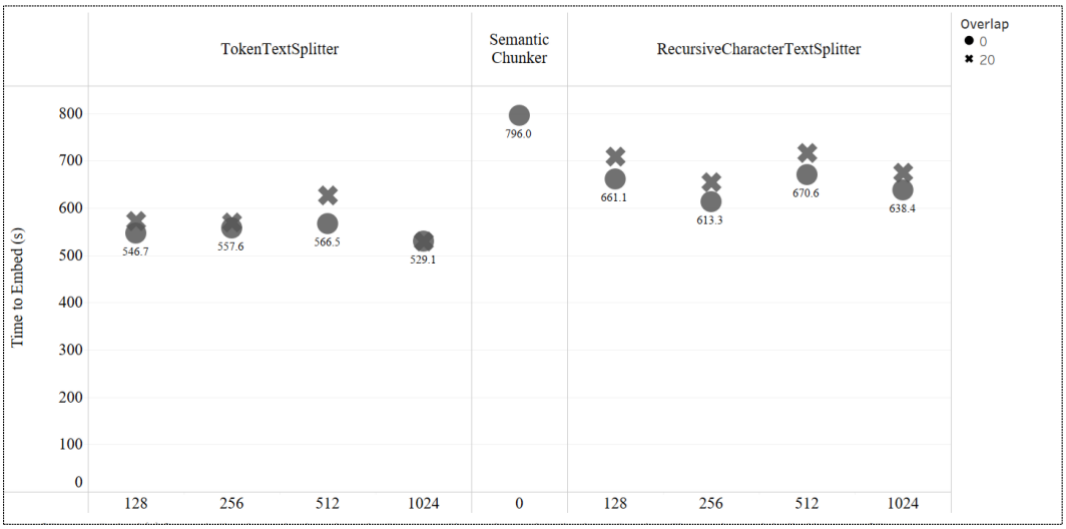

Figure 3 - *Time to embed (in seconds) per chunking strategy pair*

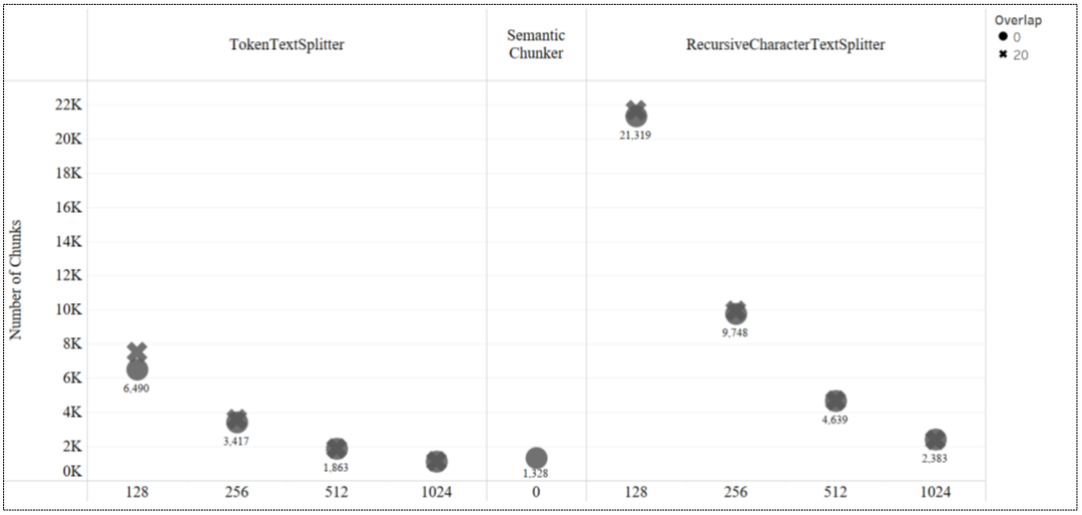

Figure 4 - *Number of chunks per chunking strategy pair*

Impact of Chunking Strategies

Empirical results demonstrate that the *TokenTextSplitter* consistently outperforms the *RecursiveCharacterTextSplitter* in terms of embedding time, regardless of chunk size or overlap. For instance, for comparable chunk sizes, the *TokenTextSplitter* achieves embedding times as low as 530 seconds, whereas the *RecursiveCharacterTextSplitter* requires over 600 seconds for each configuration. This significant performance difference highlights the efficiency of the *TokenTextSplitter* for embedding workflows where computational speed is critical.

Although the *SemanticChunker* generates substantially fewer chunks than the other strategies, it is the slowest in terms of embedding time, at around 800 seconds. This is likely due to the additional computational overhead required for semantic analysis during chunking. While its semantic accuracy may be advantageous in certain contexts, the *SemanticChunker*'s inefficiency may make it less suitable for applications where time is a primary concern.

Impact of Chunk Size

Contrary to expectations, no consistent relationship was observed between chunk size and embedding time for either the *TokenTextSplitter* or *RecursiveCharacterTextSplitter*. This

lack of a clear pattern suggests that factors such as the content or structure of the text being processed may play a more significant role than the chunk size itself. Further investigation into these factors could help clarify the interaction between chunk size and embedding time.

Impact of Chunk Overlap

The integration of chunk overlaps, which increases the total number of chunks, results in only a minor increase in embedding time for both the *TokenTextSplitter* and *RecursiveCharacterTextSplitter*. On average, embedding times rise by just 5.6%, suggesting that overlaps are computationally affordable. This makes chunk overlaps a viable strategy for preserving context between chunks without significantly compromising efficiency.

4.2 Retrieval

Retrieval is evaluated on a 0 to 1 scale (Appendix 8), where a score closer to 1 indicates the retriever has collected the information pertinent to the key context, and 0 where it fails to do so.

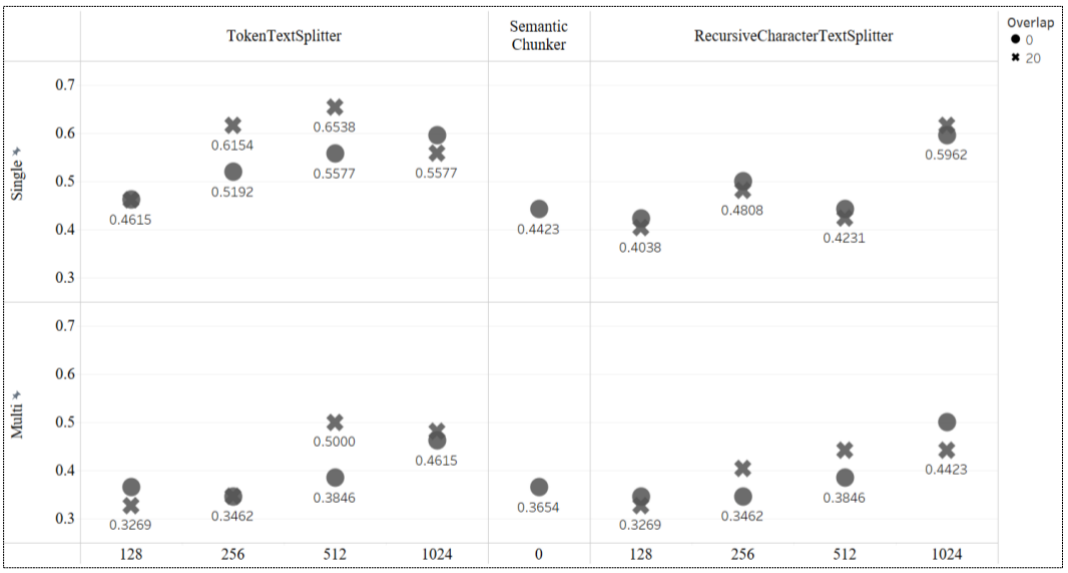

Figure 5 - *Retrieval scores for single and multi-hop questions*

Comparison of Chunking Strategies

When comparing chunking strategies, contrary to what was expected (Narimissa and Raithel, 2024), the *TokenTextSplitter* consistently outperforms other methods, achieving higher retrieval scores across all chunk sizes and configurations, with an overall average of 0.48. It is particularly sensitive to overlaps, which enhances its performance further for 256 and 512 chunk size.

The *RecursiveCharacterTextSplitter*, while achieving comparable scores for larger chunk sizes, exhibits inefficiency and inconsistent responses to overlaps, resulting in a slightly lower average performance of 0.44. Its recursive splitting logic, although effective for maintaining a hierarchical structure, may fragment context in a manner less conducive to retrieval tasks. The difference in performance between chunking strategies is more pronounced for single-hop questions, where the fixed-size strategy of the *TokenTextSplitter* is clearly superior, while for multi-hop questions, the contrast is less distinct.

Finally, the *SemanticChunker*, despite its promise of producing semantically coherent chunks, underperforms significantly in retrieval tasks, with an average retrieval score of 0.405. This indicates that semantic coherence alone does not ensure retrieval accuracy. It is possible that the *SemanticChunker* generates chunks that are too broad for matching.

Influence of Chunk Size

Retrieval performance exhibits a notable improvement as chunk sizes increase. Smaller chunks, such as those of 128 tokens, consistently underperformed across both single-hop and multi-hop questions, with average retrieval scores of 0.41 for the *TokenTextSplitter* and 0.38 for the *RecursiveCharacterTextSplitter*. In contrast, larger chunks (e.g., 1024 tokens)

demonstrated significantly higher performance, achieving averages of 0.52 and 0.54, respectively.

This trend likely stems from the increased retrieval context provided by larger chunks, which encompass more relevant information from the reference chunk. However, this improvement may also reflect subjectivity in the evaluation methodology, as larger chunks inherently overlap more with the reference chunk, thus inflating retrieval scores.

Impact of Chunk Overlap

The inclusion of chunk overlaps generally benefits retrieval performance, though the degree of impact varies across strategies. For the *TokenTextSplitter*, medium chunk sizes seemed to benefit from its inclusion, resulting in significant increases in retrieval scores for both type of questions. This suggests that overlaps may contribute to maintaining context between chunks, thus improving retrieval accuracy. It is important to note that the size selected for the overlap may be skewing these results, as for the smaller chunk size (128) the overlap may be too harsh, and for a larger chunk size, like 1024, the size may be too minute to yield an impact.

In contrast, the *RecursiveCharacterTextSplitter* showed inconsistent results, with overlaps providing limited or negligible improvements in some cases. This limited impact suggests that its recursive splitting logic may not effectively leverage overlaps to maintain context continuity. These findings indicate that while overlap inclusion is broadly beneficial, its effectiveness depends on the chunking strategy pair and potentially overlap size.

4.3 Generation

Generation is evaluated from a three-fold perspective: hallucination, overall answer quality, and generation response time. Some additional considerations on answer quality and output of response constitute a final equilibria criterion.

4.3.1 Hallucination

Hallucination is measured on a scale from 0 to 1 (Appendix 9), where scores closer to 1 suggest the answer if fully grounded on the retrieved context, whereas lower scores indicate hallucination in the generation.

Figure 6 - *Hallucination scores for single-hop questions*

Figure 7 - *Hallucination scores for multi-hop questions*

Influence of Chunking Strategies

TokenTextSplitter emerges as the most effective, achieving a marginally higher overall average hallucination score of 0.63. The *RecursiveCharacterTextSplitter*, while slightly less

effective, achieves an overall average score of 0.57, with performance comparable to the *TokenTextSplitter*. On the other hand, the *SemanticChunker* performs the worst, with an overall average hallucination score of 0.54, further emphasizing its limitations in scenarios where grounding in context is critical. Despite its semantic precision, the *SemanticChunker* fails to provide the same level of grounding as the other chunking strategies, which may render it less useful for this specific task.

Influence of Chunk Size and Overlap

The results for hallucination provide mixed insights into the role of chunk size and overlap and very little can be deduced. There is no clear trend indicating that larger or smaller chunks reduce hallucination consistently. For instance, with the *TokenTextSplitter*, scores hover around averages of 0.63 for most chunk sizes, showing minimal variation between sizes like 256, 512, and 1024 tokens. Moreover, the inclusion of overlaps results in inconsistent results that do not allow for plausible conclusions. These observations suggest that neither chunk size nor overlap alone can effectively mitigate hallucination in this setup.

Influence of LLMs

A substantial disparity exists between the performance of the two language models evaluated. OpenAI consistently outperforms Llama across all chunking strategies and configurations, with hallucination scores averaging 0.85 in single-hop and 0.84 in multi-hop evaluations for the *TokenTextSplitter*. On the other hand, Llama achieves significantly lower scores, averaging 0.46 in single-hop and 0.38 in multi-hop tasks for the *TokenTextSplitter*. This large gap indicates that model selection is a critical factor in reducing hallucination.

4.3.2 Overall Quality

Overall answer quality is measured as compared to a reference answer (Appendix 10), once again from 0 to 1, where 1 represents the highest possible score and similarity to the reference.

Figure 8 - Answer quality score for single-hop questions

Figure 10 - Answer quality scores for multi-hop questions

Influence of Chunking Strategies

The *TokenTextSplitter* consistently outperforms other chunking strategies in terms of overall answer quality, achieving an average score of 0.49. This is slightly higher than the

RecursiveCharacterTextSplitter and *SemanticChunker*, both of which average 0.46. While the differences are marginal, the *TokenTextSplitter* demonstrates a clear advantage in single-hop questions, particularly when paired with larger chunk sizes, where its grounding capabilities seem to improve.

Chunk Size & Overlap

For single-hop questions, the results suggest that increasing chunk size positively impacts answer quality, particularly when using the OpenAI model. For example, the *TokenTextSplitter* achieves progressively better scores as chunk size increases, with 1024-token chunks outperforming smaller sizes. This trend for single-hop questions is also visible for *RecursiveCharacterTextSplitter* with OpenAI's model, where larger chunks allow for greater context, resulting in improved grounding and accuracy. The same can be seen for multi-hop questions for both strategies, regarding the Llama model.

However, this pattern does not hold consistently across. While overlaps and chunk sizes occasionally yield slight improvements, the overall influence of these parameters remains inconsistent. These observations suggest that chunking parameters alone is insufficient to consistently optimize performance.

Influence of LLMs

Although there are little discernible differences in other dimensions, the answers from the *gpt-4o-mini* model consistently outperform the local Llama model, some to a great margin, for all but the *SemanticChunker* on single-hop questions.

4.3.3 Generation Response Time

Finally, the generation response time is measured as an average of the time to answer the questions that did in fact get a reply (Appendix 11).

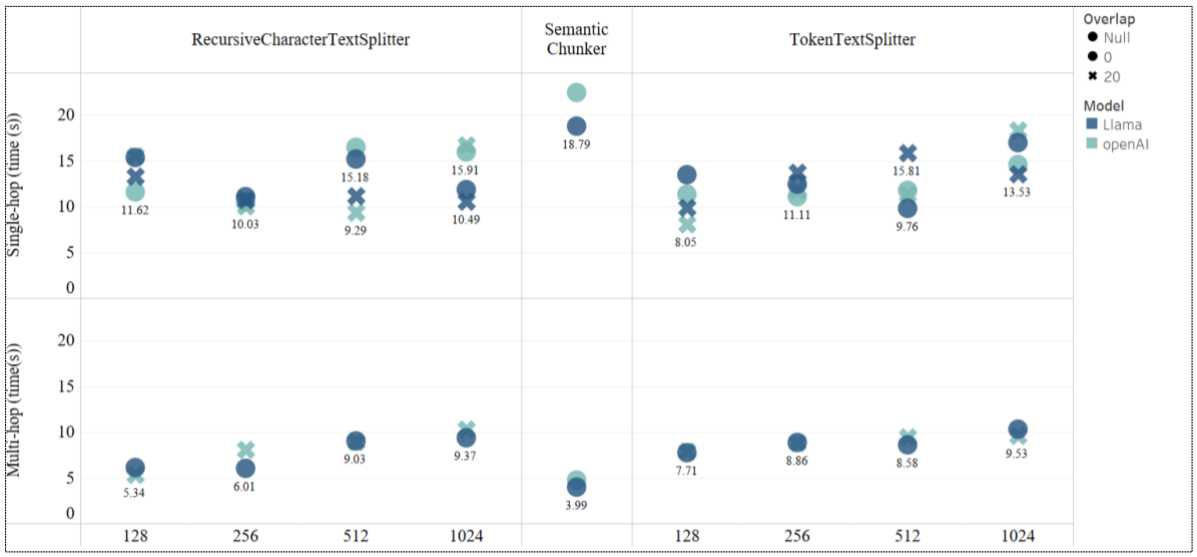

Figure 11 - Results for time (in seconds) for the generation of answers under different chunking strategy pairs

Chunking Strategy

The impact of Chunking strategies is foregoable between the *TokenTextSplitter* and *RecursiveCharacterTextSplitter*, with similar generation times for both *SemanticChunker* struggles, with a much higher generation response time.

Chunk Size & Chunk Overlap

The response generation time varies significantly depending on the chunk size and overlap configuration. Smaller chunk sizes generally lead to shorter generation times for both Llama and OpenAI models, especially in multi-hop tasks.

The addition of overlap reduces generation time for Llama in many cases. However, for OpenAI, overlap has a negligible or inconsistent impact on response times.

Influence of LLMs

Similar response times were observed for both models, thus resulting in a larger ponderation weight for other criteria, like retrieval and generation quality.

4.3.4 Additional Criteria

Final consideration was given to the prevalence of unanswered questions in each of the dataset (Appendix 12). The results are hardly significant for the Llama model generated answers, as there are very few instances of unanswered questions, which contributes to the prevalent hallucination discussed previously. For the OpenAI answers however, there is a vastly observable trend of unanswered questions for lower chunk sizes, with overlap not having distinct results. This suggests that lower chunk sizes may not retrieve the necessary context to fully ground the question and provide valuable insights.

4.3.5 Final Considerations

Given the extensive dimensions explored in this study, there is no single optimal configuration, as each setup involves a trade-off between metrics such as quality, speed, and resource efficiency. However, the findings do support some recommendations.

From the perspective of chunking strategies, *TokenTextSplitter* consistently outperformed other approaches across all evaluated dimensions, including retrieval scores, response quality, hallucination reduction, response time, as well as embedding time. Its simplicity and efficiency eliminate the need for more complex chunking strategies, such as the *RecursiveCharacterTextSplitter* or *SemanticChunker*. This aligns with previous findings, which suggest that semantic-based strategies often consume excessive computational resources without providing substantial benefits (Qu et al., 2024).

Regarding chunk size, larger chunks are recommended for some metrics, however they may not solely optimize performance of the RAG system. While smaller chunk sizes demonstrated faster response times, the difference was marginal and came at the cost of reduced retrieval scores and incomplete answers due to insufficient context. Larger chunk sizes excelled in retrieval performance and the number of complete answers generated, though their impact

on answer quality and hallucination was inconclusive. The trade-off between speed and context completeness strongly favours larger chunk sizes for use cases prioritizing accuracy and thoroughness.

Chunk overlap, while not critical, provides a slight improvement in retrieval scores. The inclusion of overlap results in a marginal increase in chunking and embedding time but has little to no impact on other metrics. Therefore, overlap can be used as a complementary configuration, but its absence will not significantly degrade performance, and its presence will not significantly enhance it.

The choice of language model proved to be the most critical factor influencing hallucination, response quality, and generation speed. OpenAI models consistently outperformed Llama in all categories, demonstrating superior grounding in retrieved context, significantly lower response times, and more complete and accurate answers. For applications where minimizing hallucination and ensuring timely responses are priorities, OpenAI's GPT models are the clear choice between the two and highlight a larger trend of more advanced models being largely more suitable.

Small-To-Big

Given the results of the experiment, the application of the Small-To-Big strategy is not advised in this context. Larger chunk sizes yield better retrieval scores, and thus retrieving from smaller chunk sizes could prove inefficient. Additionally, generation was not clearly impacted by the chunk size dimension, therefore proving further inefficiency.

4.5 Constraints and Limitations

The development and evaluation of the proposed system encountered several constrains that limited its scalability and performance.

The absence of extensive legal domain expertise posed a significant challenge throughout the development and evaluation processes. The creation of the evaluation dataset relied exclusively on LLM-generated questions, reference contexts, and answers, rather than curated legal queries derived from real-world legal practice. While the dataset was subsequently validated by a legal expert to ensure alignment with Portuguese jurisprudence, the lack of direct collaboration with multiple legal practitioners during its design limited both the depth and representativeness of the dataset. This reliance on a single point of validation restricted the ability to fully capture the complexity and variability inherent in real-world legal reasoning tasks.

Moreover, the reference contexts and answers generated during dataset creation were subsequently used to evaluate the system's outputs via another LLM. This methodology introduced a significant limitation, as it assumed the generated reference data to be entirely correct. Consequently, it constrained the evaluation process, reducing the flexibility of the LLM to produce alternative, yet still valid, responses or reasoning paths. This rigidity in evaluation could lead to underestimating the system's true performance, particularly in scenarios where legally valid interpretations differ from the reference answer.

Optimal development and implementation of LLM applications require iterative collaboration between computational experts and domain specialists, especially for tasks involving the design of evaluation datasets, contextual retrieval strategies, and prompt engineering (Szymanski et al., 2024). In this research, the synthetically generated evaluation dataset, despite being validated by a legal expert, was inherently limited in size and scope, restricting its ability to provide a comprehensive and meaningful assessment of the proposed methodology.

Additional constraints emerged from the resource-intensive requirements of the methodologies applied. The process of preparing the data - spanning scraping, cleaning, chunking, and embedding - proved particularly demanding in terms of computational resources. The cleaned database had to be processed seventeen (17) times to evaluate all the proposed techniques, significantly limiting the scale of the dataset and, by extension, the generalizability of the results.

The final critical limitation arises from the embedding models and LLMs available for use in the Portuguese legal domain. While the use of open-source LLMs like Llama-3.1-8B enabled local deployment and multilingual capabilities, their ability to reason and generate text in Portuguese legal language lagged significantly behind proprietary models like *GPT-4o-mini*. These limitations highlight the challenges of working in a less-resourced language and domain, where advancements in legal AI research have been disproportionately concentrated in English-language contexts. Addressing these gaps would require fine-tuning embedding models and LLMs with high-quality Portuguese legal corpora, alongside developing tailored architectures for legal reasoning tasks.

Despite these challenges, the research provides a foundation for advancing legal AI in the Portuguese domain. With access to advanced computational infrastructure, expanded and diversified datasets, more sophisticated language models, and iterative feedback from domain experts, the proposed methodology demonstrates significant potential for improved scalability, accuracy, and applicability. These advancements would not only enhance the system's performance but also contribute to closing the resource gap for Portuguese legal AI, enabling broader and more impactful applications.

5 Graph-Based Reasoning for Retrieval-Augmented Generation: A Study in the Portuguese Legal Domain

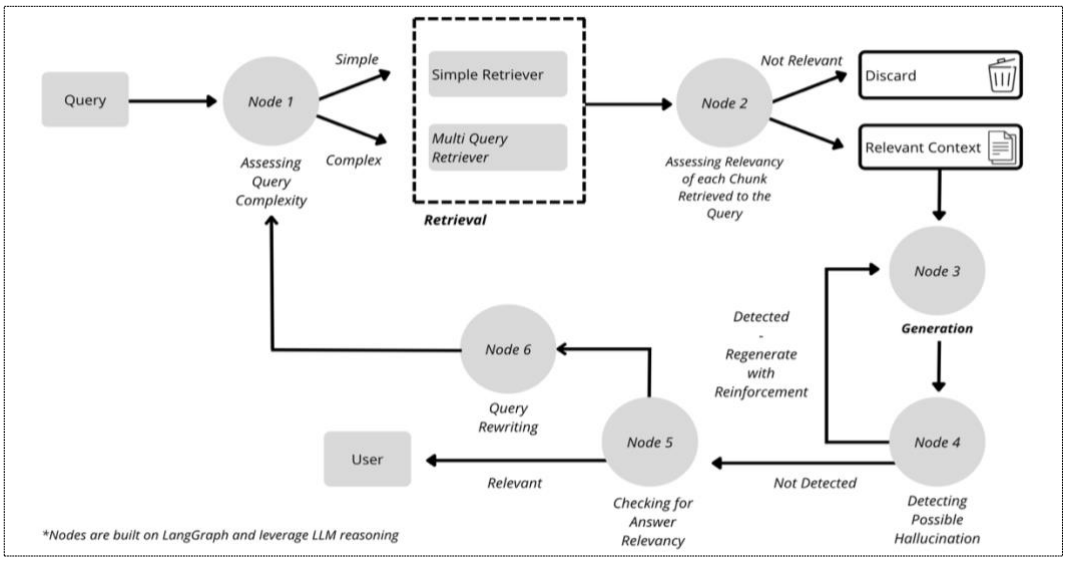

Figure 12 –Schema of Multi-Agent RAG Pipeline

5.1 Introduction

In 2023, Yao et al. referred to the unique human ability to seamlessly integrate verbal reasoning with task-oriented actions. They used the analogy of cooks, who continuously reason about their next steps after preparing ingredients, adjust their plans when they discover a missing item, or seek additional information to refine their approach while following a recipe. This interplay between reasoning and acting creates the synergy necessary for learning new tasks and making effective decisions even under uncertain conditions.

Drawing inspiration from this natural process, Yao et al. introduced the ReAct framework, the first to combine reasoning traces and task-specific actions using LLMs. By intertwining these elements, ReAct empowers machines to mimic human adaptability, enabling them to refine strategies and handle complex tasks in real time. These principles of reasoning

and acting have become increasingly relevant in generative AI, as mechanisms like self-reflection enhance the performance of AI frameworks (Renze ang Guven, 2024).

While transformative, traditional RAG architectures - critical for domain-specific tasks such as those in the Portuguese legal system - remain constrained by their static nature. Challenges such as fixed retrieval configurations, lack of self-verification mechanisms, and inefficient handling of query complexity limit their adaptability and full potential, despite their strengths in enhancing traceability and reducing hallucination (Gao et al., 2024).

This section explores the implementation of Multi-Agent Systems (MASs) through graph-based flows in the RAG infrastructure targeting flexibility within the retrieval and generation components. By referencing significant prior work on RAG LLM reasoning and self-assessment, this study adopts an ablation-oriented approach to evaluate the contributions of individual components and their combination. The objective is to identify the elements of the pipeline that derive the greatest benefit from adaptive nodes, demonstrating their potential to enhance the efficiency and accuracy of RAG systems.

5.2 The Static Nature of RAG Architectures

A key limitation of the RAG framework lies in its static nature, a result of its one-size-fits-all architecture, affecting both the retrieval and generation components.

From a retrieval perspective, the rigidity of RAG systems creates two significant issues. First, retrieval configurations, such as the number of documents retrieved per query or similarity thresholds, are applied uniformly across all queries. This inflexibility leads to trade-offs: thresholds set too high risk missing essential information, while those set too low result in an overflow of irrelevant data that overwhelms the system (Kratzwald and Feuerriegel, 2018).

Second, while the group component shows that larger chunk sizes tend to improve retrieval scores, this also comes with significant trade-offs. Larger chunks often introduce

excessive or irrelevant context, increasing computational overhead, as seen in the longer times required to generate answers. Despite higher retrieval scores, the quality of the generated responses does not improve proportionally. This occurs because the LLM processes all retrieved information during generation, including irrelevant content, which it struggles to filter effectively, resulting in degraded outputs (Shi et al., 2023).

In addition to retrieval challenges, RAG systems lack flexibility in query management, rigidly requiring retrieval for every query - even when unnecessary. In many cases, the parametric knowledge embedded within the LLM may be sufficient to generate accurate and relevant responses without external retrieval (Zhang et al., 2024). When retrieval is redundant, it introduces unnecessary computational overhead, increases inference time, and reduces overall efficiency. Although the evaluation dataset assumes retrieval for every query, this limitation is critical in real-world applications where users may ask clarifying or follow-up questions. Applying the same retrieval approach uniformly to all queries - whether simple or complex - leads to inefficiencies, emphasizing the need for adaptive mechanisms tailored to query complexity.

Post-retrieval, the static nature of RAG also affects the quality of generated responses. While RAG frameworks are agreed to be more reliable than parametric-only models for reducing hallucinations and producing faithful, grounded answers, both existing research from Shuster et al. (2021) and empirical findings from the group study indicate that this reliability is not guaranteed. The absence of verification mechanisms in generation means the system cannot systematically evaluate or correct hallucinations, particularly when processing large volumes of retrieved data.

5.3 Reference Works in LLM Reasoning in RAG Architectures

Recent research has highlighted the static nature of RAG systems, through the development of novel frameworks designed to address the limitations discussed above. This section details several of these frameworks, focusing on their principles and methodologies, as they form the foundation for this study in enhancing retrieval and generation processes within the RAG system.

One of the most significant advancements in this area is the Self-Reflective RAG framework (*Self-RAG*), introduced by Asai et al. (2024), which achieves state-of-the-art outputs by addressing challenges inherent to the retrieval and generation processes. To achieve this, the framework trains an arbitrary language model in an end-to-end manner, enabling it to generate and leverage special tokens known as reflection tokens. These tokens act at critical stages of the RAG workflow, first determining whether to retrieve information, then assessing the relevance of retrieved chunks. Finally, they evaluate generated responses for hallucination and alignment with the query, ensuring the selection of only the best results.

Other token-based methodologies include QUARK (Lu et al., 2022), which applies reinforcement learning to guide generation through a reward-based mechanism to align outputs with user expectations and unlearn bad behaviors. Similarly, the CTRL framework (Keskar et al., 2019) uses control tokens to allow precise guidance of generation attributes, such as style and content, offering a mechanism for tailoring outputs to specific needs.

Focusing on query complexity, Adaptive-RAG (Jeong et al., 2024) trains a classifier - a smaller language model - to predict the complexity of incoming queries, which the system categorizes into three levels: those requiring no retrieval, simple retrieval, or iterative retrieval.

This method allows for reducing computational overhead for simple queries and avoiding inadequate handling of complex ones. The CRAG framework from Yan et al. (2024), extends this adaptability to query complexity by introducing corrective mechanisms that evaluate the quality of retrieved information. It adopts a "decomposition and recomposition" technique that partitions the chunks into "knowledge strips" and maintains only the relevant pieces of context to reduce Hallucinations. Similarly, Yoran et al. (2023) emphasize robustness in retrieval-augmented models by employing natural language inference filtering on retrieved passages and fine-tuning the model with mixed-content datasets containing both relevant and irrelevant material, which mitigate the impact of irrelevant retrieved passages and of cascading-errors, particularly in multi-hop questions. Further advancements include frameworks like SAIL (Luo et al., 2023), which incorporate external knowledge sources, such as internet-based searches, to complement non-parametric data.

5.4 Multi-Agent Systems

5.4.1 Introduction to Multi-Agent Systems

Multi-Agent Systems (MASs) (Weiss et al., 2000) are composed of multiple interacting computing entities, or agents, capable of reasoning and acting autonomously while collaborating to achieve individual or shared objectives. According to Wooldridge (2002), agents exhibit two core characteristics: independence in decision-making and adaptability to their environment. These properties enable MASs to address complex, distributed problems beyond the scope of centralized approaches.

MASs emerged from the field of Distributed Artificial Intelligence, which explored how autonomous entities could collaborate to address complex problems beyond the capabilities of individual systems. This framework has proven valuable in addressing distributed problems across a variety of domains, from optimizing industrial processes (Yahouni et al., 2021) to

improving decision-making in healthcare systems (Shakshuki & Reid, 2015). Organizationally, MASs can adopt a variety of structures from hierarchical to peer-to-peer organizations, among others (Abbas et al., 2015), allowing them to adapt to system complexity and requirements.

The integration of LLMs into agent decision-making processes has introduced promising opportunities for advancing MASs (de Zarza et al., 2023). Previous studies demonstrate that LLMs can significantly enhance agent performance by providing strategic recommendations and facilitating informed decision-making in various tasks such as negotiation (Muglich et al., 2022) and coordination (Yang et al., 2022). This synergy between MASs and LLMs highlights the capacity of recent advancements in GenAI to augment the autonomy and collaborative capabilities of agents, opening new avenues for addressing increasingly complex and dynamic challenges.

5.4.2 MASs in RAG Architectures

MASs have been increasingly utilized in RAG systems to enhance efficiency, adaptability, and scalability in domain-specific tasks. For instance, Gamage et al. (2024) describe a supervisory MAS in a net-zero emissions energy system, where agents manage anomaly detection, behavior analysis, and visualization, optimizing workflows by processing structured and unstructured data. Similarly, in business information extraction (Arslan et al., 2024), MASs automate the retrieval, enrichment, and classification of events from diverse sources, working collaboratively in a pipeline.

Beyond task delegation, MASs enable dynamic decision-making across retrieval and generation steps, improving adaptability in RAG pipelines (Ghosh, 2024). The *Modular RAG* paradigm introduced by Gao et al. (2024) aligns with this approach by decomposing RAG systems into autonomous but collaborative task-specific modules for retrieval, re-ranking, and generation, allowing for fine-tuning and task-specific optimizations.

5.5 A Multi-Agent RAG through LangGraph

LangGraph, part of the LangChain ecosystem, is a library meant for developing stateful, multi-agent applications leveraging LLMs. It translates workflows into graphs, where nodes correspond to agents and edges define the flow of information. This approach enables flexibility in designing complex systems, supporting the diverse MASs discussed in section 8.4., and tracks and updates a central state, ensuring all agents share a common understanding of the task's progress (LangChain, 2024). Therefore, it is a valuable tool for exploring graph-based RAG architectures, particularly in tasks requiring dynamic decision-making and adaptive workflows.

This study builds upon insights from methodologies presented in section 8.3 that aim to overcome the static nature of traditional architectures by incorporating self-assessment and self-reflection mechanisms under a more resource-efficient and modular approach, overcoming resource and restrictions on domain knowledge, involved in training and fine-tuning models.

The principles of these methodologies are adapted into a graph-based framework to systematically examine their impact on RAG systems through ablation experiments. The focus of this exploration is on three key components: query processing, retrieval validation, and generation quality. The evaluation of these components follows the common framework detailed in earlier sections, ensuring consistency and comparability of results. Central to the system's design is the use of LLM reasoning, which underpins all decision-making processes (represented as nodes) and ensures adaptability across the pipeline.

5.5.1 Query Processing

In this study, all queries in the testing dataset require retrieval, rendering threshold-based decisions to retrieve, such as in *Self-RAG*, unnecessary. Instead, a classification process

distinguishes between simple and complex queries (Appendix 13), a distinction influenced by the Adaptive-RAG and CRAG frameworks. Simple queries proceed through the standard RAG pipeline framed in section 3.5, while complex queries are processed using LangChain's *MultiQueryRetriever*. This retriever generates multiple queries from diverse perspectives through LLM reasoning, retrieves relevant documents for each query, and combines the results into a deduplicated, unified set. By diversifying retrieval perspectives, this approach mitigates limitations inherent in distance-based methods and ensures a richer context for subsequent stages.

5.5.2 Retrieval Validation

To ensure the quality of the retrieved documents, the retrieval validation process iterates through each document to evaluate its relevance to the original query. Building on principles from *Self-RAG*, this iterative process determines the inclusion or exclusion of documents in the final context used for generating answers, which reduces the risks associated with excessive or irrelevant context (Appendix 14).

5.5.3 Generation Quality

The final stage in the RAG pipeline focuses on ensuring that generated answers (Appendix 6) meet the standards of accuracy, relevance, and alignment with the retrieved context. To achieve this, two mechanisms are implemented. The first involves hallucination detection, where the generated answer is compared against the retrieved chunks to verify its grounding (Appendix 15). If the answer is found to diverge significantly from the context, the system triggers a regeneration process using the same retrieval set, already approved by previous stages, but with a prompt focusing on its previous fault (Appendix 16). The second mechanism addresses relevancy validation, assessing whether the generated answer directly addresses the user query (Appendix 17). In cases where the answer fails to meet the relevance

criteria, the system employs LLM reasoning to refine the query or suggest alternative formulations (Appendix 18), enabling an adaptive and iterative approach to improve alignment with user intent.

5.5.4 Application

The different nodes were applied independently, with experiments conducted for query complexity assessment, document relevance selection, hallucination detection, and query rewriting. These were further combined into more comprehensive structures: a *Self-RAG*-like setup integrating document grading, hallucination and relevancy checks, and query rewriting for non-relevant results; a CRAG structure combining complexity assessment with document validation; and a complete infrastructure incorporating all nodes (Figure 12). All configurations are detailed in Appendix 19, based on the nodes from Figure 14.

5.6 Results

Each experimental configuration was evaluated on the single-hop and multi-hop question datasets with 2 of the best performing chunking strategy pairs, *TokenTextSplitter*, of chunk size 512, with both no overlap and overlap of 20. The results were averaged across these two configurations for each type of question, with the baseline for comparison derived from the previous experiments (Appendix 20). The baseline employed the simple RAG architecture without additional reasoning layers, while the evaluated methods incorporated advanced components to assess their impact on performance.

Evaluations were conducted using both OpenAI *gpt-4o-mini* and extended to MistralAI's NeMo, an open 12B parameter model that supports Portuguese. Mistral was selected over Llama, which had been utilized in the group component, due to compatibility constraints; LangChain and LangGraph methods are not fully adapted to Llama, whereas Mistral offers comparable capabilities to *Llama 3.1-8B* (*Mistral*) alongside the necessary

compatibility. Additional results for baseline were ran for Mistral, under the same framework in section 3.5 (Appendix 21).

The results will be compared in a 2-fold method: from an LLM perspective, and differentiating single-hop from multi-hop questions, to try to derive conclusions on the application as well as what configurations may be most beneficial. The results discussed represent the percentual difference between the baseline and the applications, highlighting only significant differences. Time was not taken into consideration, due to time increments applied for MistralAI to safeguard the maximum of 1 request per second.

5.6.1 LLM Comparison (Appendices 22 & 23)

A key observation in the comparison of LLMs is the varying degree of improvement observed for each model. MistralAI's NeMo, as an approximation of Llama and therefore a weaker model, displays significantly larger enhancements from the integration of self-reflection techniques compared to the more advanced GPT model. For example, Mistral shows substantial improvements in answer quality metric with *Self-RAG* (+20%) and Hallucination checking (+17%), whereas GPT achieves comparatively more modest gains, such as +5% for *Self-RAG* and +10% for Hallucination Checking. These results suggest that a stronger baseline reasoning ability, such as that of GPT, may reduce the benefits of additional reasoning nodes.

For GPT, most setups provided limited gains in answer quality (hovering around +3% to +6%) and retrieval assessment, with all configurations negatively impacting the Hallucination scores. On the other hand, for Mistral, Hallucination Checking and Document Grading proved particularly effective, with gains of +20% and +16%, respectively, in retrieval scores. Still, both models benefited from ensemble setups such as *Self-RAG* and the complete architecture, with these configurations amplifying gains across multiple metrics. Notably, *Self-*

RAG for Mistral saw improvements of +20% in answer quality and +15% in retrieval, highlighting the synergy of combining hallucination control and document grading.

However, certain setups, such as Query Rewriting and Complexity Assessment, failed to produce significant standalone improvements. For example, Query Rewriting negatively impacted Mistral's retrieval (-13%) and hallucination (-29%) scores, while Complexity Assessment showed only minor gains (+5% for Answer in Mistral, +4% in GPT) but substantial degradation in hallucination (-17% for Mistral, -7% for GPT). The CRAG ensemble also performed poorly, particularly for Mistral, where it showed no improvement in Answer scores (0.00%) and degraded Hallucination (-25%). These results highlight the need for thoughtful integration of reasoning components to avoid redundancy or performance degradation.

5.6.2 Question Type Comparison (Appendices 24 & 25)

Single-hop questions demonstrated more pronounced improvements in answer quality compared to multi-hop questions, reflecting their simpler nature. Gains in retrieval scores for single-hop were modest across most setups, with the highest coming from Hallucination Checking (+3%) and the Complete ensemble (+2%), suggesting that basic retrieval mechanisms suffice for these tasks.

Multi-hop questions, in contrast, showed their greatest benefits in enhancing retrieval scores, which in turn were correlated to higher answer quality score impacts, emphasizing the importance of robust mechanisms to handle their complexity. Document Grading and *Self-RAG* delivered strong performance for multi-hop tasks, with retrieval score improvements of +16% and +17%, and subsequent answer scores of +8% and 10%, respectively. The Complete ensemble also performed well, yielding +12% in retrieval and +11% in answer quality scores. Despite these retrieval-focused improvements and their corresponding answer quality

improvements, multi-hop tasks experienced smaller gains in answer quality, reflecting the increased difficulty of generating coherent answers across complex queries.

Hallucination remained a persistent challenge across both question types. Multi-hop questions proved particularly prone to hallucination-related degradation, with most configurations showing negative impacts. However, some configurations, such as Hallucination Checking, *Self-RAG*, and the Complete ensemble, demonstrated some mitigation of this issue, with multi-hop hallucination scores degrading slightly less, suggesting potential in the hallucination check node, present in all.

Overall, the comparative analysis highlights the differing demands of single-hop and multi-hop questions. Single-hop queries favour straightforward reasoning approaches, while multi-hop questions rely more heavily on retrieval-focused strategies to achieve balanced outcomes. Ensemble approaches such as *Self-RAG* and Complete ensemble consistently stand out as the top performers across both question types, amplifying the benefits of stand-out individual nodes like Hallucination checking and Document Grading.

5.7 Discussion

The results presented in this study highlight significant variability across models and question types. Techniques like Hallucination Checking, Document Grading, and their combination in the *Self-RAG*-like and Complete ensembles presented promising results across. However, the study's exploratory nature, constrained by the use of prompting frameworks instead of task-specific fine-tuning, led to time-intensive answer generation (Appendix 26) and increased complexity from combining multiple techniques, underscoring the need for optimization. Fine-tuning models to the best configurations could streamline processes and reduce latency.

While the prompts guiding LLM reasoning were refined using AI tools, further tailoring could improve results significantly. Testing these prompts on larger, more diverse datasets would better evaluate their generalization and effectiveness. Future work should focus on optimizing prompts, incorporating task-specific tuning, and scaling experiments to more complex datasets. These steps could address challenges like hallucination control and enhance the reasoning capabilities of weaker LLMs, advancing RAG systems.

6 Conclusion

This study explored RAG systems tailored for the Portuguese legal domain, addressing the challenges posed by underrepresented languages in legal and AI research. By implementing and evaluating advanced methodologies such as chunking strategies and graph-based reasoning, this work sheds light on the opportunities and obstacles in enhancing legal tool for Portuguese-speaking legal professionals.

Key findings, in the group component, revealed that the *TokenTextSplitter* approach yielded the most effective collections. While Semantic and Recursive chunking techniques faced computational limitations, larger chunk sizes were found to improve retrieval accuracy and answer quality significantly, emphasizing the importance of fine-tuning chunking parameters for legal tasks. However, as European Portuguese is underrepresented in the training data of cutting-edge LLMs compared to English, performance discrepancies were observed. Larger multilingual models helped bridge this gap, aligning with trends noted in OpenAI (2024) and other studies, highlighting the scalability of larger parameter counts for multilingual tasks.

The integration of graph-based reasoning showed significant variability across models and question types, with techniques like Hallucination Checking and Document Grading performing well on their own, and amplifying results when combined in ensembles. However, the use of prompting frameworks, rather than task-specific fine-tuning, led to slower answer generation and increased complexity. While fine-tuning models could improve efficiency and reduce latency, further refining prompts and testing them on larger, more diverse datasets could improve generalization. Future work should focus on optimizing prompts, task-specific tuning, and scaling to more complex datasets, addressing challenges like hallucination control and enhancing the reasoning capabilities of weaker LLMs.

Despite its promising findings, this study faced several limitations. Computational constraints restricted the size of the dataset and the scope of the knowledge base, potentially affecting retrieval performance. Furthermore, the absence of domain expertise within the team limited the contextual evaluations of legal nuances. These limitations highlight the importance of expanding resources and expertise in future iterations.

Future Directions

To further improve the performance and applicability of RAG systems, future work should focus on several areas. Fine-tuning with larger, more diverse datasets, including cross-jurisdictional legal documents, would broaden applicability and enhance retrieval accuracy. Exploring alternative embeddings methods, retrieval strategies, and chunking configurations could yield more computationally efficient solutions. Additionally, the inclusion of domain expertise from legal professionals would ensure more accurate and contextually relevant outputs, as well as a more faithful evaluation framework. Extending this work to other underrepresented languages would further generalize the findings and address the linguistic inequities in AI research.

Portuguese, despite being the 5th most spoken language globally, has significantly worse performance in LLMs when compared to Mandarin or English, the 2 most spoken languages worldwide. This is partly due to limited funding and research contributions from Portuguese-speaking countries, as well as the already mentioned bias in sources coming to LLMs. By addressing these gaps and fostering collaboration across linguistic domains, this work highlights the potential to bridge these disparities and improve AI support for underrepresented languages.

Closing remarks

This study contributes to the growing body of research aimed at reducing linguistic and domain-based inequities in AI. By focusing on European Portuguese legal analysis, this work sets the groundwork for extending RAG systems to other underrepresented languages and specialized domains, ultimately promoting inclusivity and advancing the utility of AI-driven tools in legal contexts.

References

Abbas, Hosny Ahmed. 2015. “Organization of Multi-Agent Systems: An Overview.” *International Journal of Intelligent Information Systems* 4 (3): 46. https://doi.org/10.11648/j.ijiis.20150403.11.

AIChatOnline.org. n.d. “Alpaca Law: O Assistente Legal Português-Free Portuguese legal assistance.” https://aichatonline.org/gpts-2OToEk6EXI-alpaca-law-o-assistente-legal-portugu%C3%AAs.

Alghisi, Simone, Massimo Rizzoli, Gabriel Roccabruna, Seyed Mahed Mousavi, and Giuseppe Riccardi. 2024. “Should We Fine-Tune or RAG? Evaluating Different Techniques to Adapt LLMs for Dialogue.” *arXiv (Cornell University)*, June. https://doi.org/10.48550/arxiv.2406.06399.

Almeida, Thales Sales, Hugo Abonizio, Rodrigo Nogueira, and Ramon Pires. 2024a. “Sabi\’a-2: A New Generation of Portuguese Large Language Models.” arXiv.Org. March 14, 2024. https://arxiv.org/abs/2403.09887.

Alvarez, Jose M., Alejandra Bringas Colmenarejo, Alaa Elobaid, Simone Fabbrizzi, Miriam Fahimi, Antonio Ferrara, Siamak Ghodsi, Carlos Mougan, Ioanna Papageorgiou, Paula Reyero, Mayra Russo, Kristen M. Scott, Laura State, Xuan Zhao, and Salvatore Ruggieri. 2024. "Policy Advice and Best Practices on Bias and Fairness in AI." *Ethics and Information Technology* 26, article 31. Published April 29, 2024. https://doi.org/10.1007/s10676-024-09746-w.

Amann, Julia, Alessandro Blasimme, Effy Vayena, Dietmar Frey, and Vince I. Madai. 2020. “Explainability for artificial intelligence in healthcare: a multidisciplinary perspective.”

BMC Medical Informatics and Decision Making 20 (1). https://doi.org/10.1186/s12911-020-01332-6.

Arslan, Muhammad, Saba Munawar, and Christophe Cruz. 2024. “Sustainable Digitalization of Business with Multi-Agent RAG and LLM.” *Procedia Computer Science* 246 (January): 4722–31. https://doi.org/10.1016/j.procs.2024.09.337.

Asai, Akari, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. “Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection.” arXiv.Org. October 17, 2023. https://arxiv.org/abs/2310.11511.

Aydın, Ömer, and Enis Karaarslan. 2023a. “Is ChatGPT Leading Generative AI? What is Beyond Expectations?” *SSRN Electronic Journal*, January. https://doi.org/10.2139/ssrn.4341500.

Azamfirei, Razvan, Sapna R. Kudchadkar, and James Fackler. 2023a. “Large language models and the perils of their hallucinations.” *Critical Care* 27 (1). https://doi.org/10.1186/s13054-023-04393-x.

Bathaee, Yavar. 2018. "The Artificial Intelligence Black Box and the Failure of Intent and Causation." *Harvard Journal of Law & Technology* 31 (2): 890–893. https://jolt.law.harvard.edu/assets/articlePDFs/v31/The-Artificial-Intelligence-Black-Box-and-the-Failure-of-Intent-and-Causation-Yavar-Bathaee.pdf.

Bergmann IBM (blog). March 15, 2024. https://www.ibm.com/topics/fine-tuning.

Bogen, Miranda. "All the Ways Hiring Algorithms Can Introduce Bias." Harvard Business Review, May 6, 2019. Accessed October 31, 2024. https://hbr.org/2019/05/all-the-ways-hiring-algorithms-can-introduce-bias.

Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Von Arx Sydney, Michael S. Bernstein, et al. 2021. “On the Opportunities and Risks of Foundation Models.” arXiv.Org. August 16, 2021. https://arxiv.org/abs/2108.07258.

Boyd, Christina L., Lee Epstein, and Andrew D. Martin. "Untangling the Causal Effects of Sex on Judging." American Journal of Political Science 54, no. 2 (2010): 389–411. Accessed October 31, 2024. https://doi.org/10.1111/j.1540-5907.2010.00437.x.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. “Language Models are Few-Shot Learners.” arXiv.Org. May 28, 2020. https://arxiv.org/abs/2005.14165.

Bruckhaus, Tilmann. 2024. “RAG Does Not Work for Enterprises.” arXiv.Org. May 31, 2024. https://arxiv.org/abs/2406.04369.

Chaerul Haviana, Sam Farisa, Sri Mulyono, and Badie’Ah. 2023. “The Effects of Stopwords, Stemming, and Lemmatization on Pre-trained Language Models for Text Classification: A Technical Study.” IEEE Conference Publication | IEEE Xplore. September 20, 2023. https://ieeexplore.ieee.org/document/10295797.

Chalkidis, Ilias, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2021. “LexGLUE: A Benchmark Dataset for Legal Language Understanding in English.” arXiv.Org. October 3, 2021. https://arxiv.org/abs/2110.00976.

Chalkidis, Ilias, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. “LEGAL-BERT: The Muppets straight out of Law School.” *Findings of the Association for Computational Linguistics: EMNLP 2020, Pages 2898–2904, Online. Association for Computational Linguistics.*, January. https://doi.org/10.18653/v1/2020.findings-emnlp.261.

Chatbot Arena. n.d. "Homepage." Accessed December 5, 2024. https://lmarena.ai.

Chen, Kezhen, Linda He, Ben Athiwaratkun, Jue Wang, Maurice Weber, Heejin Jeong, Yonatan Oren, and Michael Poli. 2024. “Building a personalized code assistant with open-source LLMs using RAG Fine-tuning.” 2024. https://www.together.ai/blog/rag-fine-tuning.

Comissão Nacional de Proteção de Dados (CNPD). "O Que Somos e Quem Somos." Accessed October 31, 2024. https://www.cnpd.pt/cnpd/o-que-somos-e-quem-somos/.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. “Unsupervised Cross-lingual Representation Learning at Scale.” arXiv.Org. November 5, 2019. https://arxiv.org/abs/1911.02116.

Conrad, Jack G., Shirsha Ray Chaudhuri, Shounak Paul, and Saptarshi Ghosh. 2023. “AI & Law: Formative Developments, State-of-the-Art Approaches, Challenges & Opportunities.” *Proceedings of the 6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD) (CODS-COMAD ’23)*, January. https://doi.org/10.1145/3570991.3571050.

Council of Bars and Law Societies of Europe. Model Code of Conduct for European Lawyers. Brussels: CCBE, 2021. Accessed October 21, 2024. https://www.ccbe.eu/fileadmin/speciality_distribution/public/documents/DEONTOLOGY/DEON_CoC/EN_DEONTO_2021_Model_Code.pdf

Council of the European Union. "The General Data Protection Regulation." European Council. Accessed October 21, 2024. https://www.consilium.europa.eu/en/policies/data-protection/data-protection-

regulation/#:~:text=The%20GDPR%20establishes%20the%20general,data%20processing%20operations%20they%20perform.

Coupette, Corinna, Janis Beckedorf, Dirk Hartung, Michael Bommarito, and Daniel Martin Katz. 2021. “Measuring Law Over Time: A Network Analytical Framework with an Application to Statutes and Regulations in the United States and Germany.” *Frontiers in Physics* 9 (May). https://doi.org/10.3389/fphy.2021.658463.

Cousineau, Carter. "Responsible AI Implementation Starts with Human-in-the-Loop Oversight." Thomson Reuters, August 28, 2024. Accessed October 31, 2024. https://www.thomsonreuters.com/en-us/posts/innovation/responsible-ai-implementation-starts-with-human-in-the-loop-oversight/.

Cui, Jiaxi, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. 2023. “Chatlaw: A Multi-Agent Collaborative Legal Assistant with Knowledge Graph Enhanced Mixture-of-Experts Large Language Model.” arXiv.Org. June 28, 2023. https://arxiv.org/abs/2306.16092.

De Zarzà, I., J. De Curtò, Gemma Roig, Pietro Manzoni, and Carlos T. Calafate. 2023. “Emergent Cooperation and Strategy Adaptation in Multi-Agent Systems: An Extended Coevolutionary Theory with LLMs.” *Electronics* 12 (12): 2722. https://doi.org/10.3390/electronics12122722.

Desai, Akash A. 2023. “Hybrid Search: Combining BM25 and Semantic Search for Better Results with LangChain.” *Medium*, December 9, 2023. https://medium.com/etoai/hybrid-search-combining-bm25-and-semantic-search-for-better-results-with-lan-1358038fe7e6.

Die Bundesregierung. "AI Act: Europäisches Gesetz zur Künstlichen Intelligenz." Accessed October 21, 2024. https://www.bundesregierung.de/breg-de/themen/digitalisierung/kuenstliche-intelligenz/ai-act-2285944.

Dinstein, Orrie, and Jaymin Kim. "'Human in the Loop' in AI Risk Management — Not a Cure-All Approach." IAPP News, August 21, 2024. Accessed October 31, 2024. https://iapp.org/news/a/-human-in-the-loop-in-ai-risk-management-not-a-cure-all-approach.

Doshi-Velez, Finale, and Been Kim. "Towards A Rigorous Science of Interpretable Machine Learning." arXiv preprint arXiv:1702.08608 (2017): 3-8. Accessed October 31, 2024. https://arxiv.org/abs/1702.08608.

Edge, Darren, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. “From Local to Global: A Graph RAG Approach to Query-Focused Summarization.” *arXiv (Cornell University)*, April. https://doi.org/10.48550/arxiv.2404.16130.

Ejjami, Rachid. 2024. “AI-driven Justice: Evaluating the Impact of Artificial Intelligence on Legal Systems.” *International Journal for Multidisciplinary Research* 6 (3). https://doi.org/10.36948/ijfmr.2024.v06i03.23969.

“Embedding API.” n.d. https://jina.ai/embeddings/.

European Parliament. "EU AI Act: First Regulation on Artificial Intelligence." Last updated June 18, 2024. Accessed October 21, 2024. https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence.

European Parliamentary Research Service. The Impact of the General Data Protection Regulation (GDPR) on Artificial Intelligence. PE 641.530. Brussels: European

Parliament, June 2020. Accessed October 21, 2024. https://www.europarl.europa.eu/RegData/etudes/STUD/2020/641530/EPRS_STU(2020)641530_EN.pdf.

European Union Agency for Fundamental Rights. Bias in Algorithms: Artificial Intelligence and Discrimination. Vienna, 2022. Accessed October 31, 2024. https://fra.europa.eu/sites/default/files/fra_uploads/fra-2022-bias-in-algorithms_en.pdf

Fan, Wenqi, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models." arXiv.Org. May 10, 2024. https://arxiv.org/abs/2405.06211.

Gamage, Gihan, Nishan Mills, Daswin De Silva, Milos Manic, Harsha Moraliyage, and Andrew Jennings. 2024. "Multi-Agent RAG Chatbot Architecture for Decision Support in Net-Zero Emission Energy Systems." IEEE Conference Publication | IEEE Xplore. March 25, 2024. https://ieeexplore.ieee.org/document/10540920/.

Ganesan, Kavita. 2018. "ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks." *arXiv (Cornell University)*, January. https://doi.org/10.48550/arxiv.1803.01937.

Gao, Yunfan, Yun Xiong, Meng Wang, and Haofen Wang. 2024. "Modular RAG: Transforming RAG Systems into LEGO-like Reconfigurable Frameworks." arXiv.Org. July 26, 2024. https://arxiv.org/abs/2407.21059v1.

Gao, Yunfan, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2023. "Retrieval-Augmented Generation for Large Language Models: A Survey." arXiv.Org. December 18, 2023. https://arxiv.org/abs/2312.10997.

GeeksforGeeks. 2024. “Understanding BLEU and ROUGE score for NLP evaluation.” GeeksforGeeks. October 4, 2024. https://www.geeksforgeeks.org/understanding-bleu-and-rouge-score-for-nlp-evaluation/.

Ghosh, Bijit. 2024. “Agentic RAG - Bijit Ghosh - Medium.” *Medium*, November 28, 2024. https://medium.com/@bijit211987/agentic-rag-81ed8527212b.

Greenstein, Stanley. 2021. “Preserving the rule of law in the era of artificial intelligence (AI).” *Artificial Intelligence and Law* 30 (3): 291–323. https://doi.org/10.1007/s10506-021-09294-4.

Guha, Neel, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Aditya K, Alex Chohlas-Wood, et al. 2023. “LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models.” December 15, 2023. https://proceedings.neurips.cc/paper_files/paper/2023/hash/89e44582fd28ddfea1ea4dcb0ebbf4b0-Abstract-Datasets_and_Benchmarks.html.

Harrison Chase. LangChain, October 2022.

Harsoor, Sharan. 2024. “Embeddings: A Deep Dive from Basics to Advanced Concepts.” *Medium*, December 3, 2024. https://medium.com/@sharanharsoor/embeddings-a-deep-dive-from-basics-to-advanced-concepts-f092765476fc.

“Home - Crawl4AI Documentation.” n.d. https://crawl4ai.com/mkdocs/.

Hou, Abe Bohan, Orion Weller, Guanghui Qin, Eugene Yang, Dawn Lawrie, Nils Holzenberger, Andrew Blair-Stanek, and Van Durme Benjamin. 2024. “CLERC: A Dataset for Legal Case Retrieval and Retrieval-Augmented Analysis Generation.” arXiv.Org. June 24, 2024. https://arxiv.org/abs/2406.17186.

Hu, Yuntong, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2024a. “GRAG: Graph Retrieval-Augmented Generation.” *arXiv (Cornell University)*, May. https://doi.org/10.48550/arxiv.2405.16506.

Huang, Quzhe, Mingxu Tao, Chen Zhang, Zhenwei An, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023. “Lawyer LLaMA Technical Report.” arXiv.Org. May 24, 2023. https://arxiv.org/abs/2305.15062.

Huang, Yizheng, and Jimmy Huang. 2024. “A Survey on Retrieval-Augmented Text Generation for Large Language Models.” arXiv.Org. April 17, 2024. https://arxiv.org/abs/2404.10981v1.

Ilin, Ivan. 2024. “Advanced RAG Techniques: an Illustrated Overview - Towards AI.” *Medium*, February 27, 2024. https://pub.towardsai.net/advanced-rag-techniques-an-illustrated-overview-04d193d8fec6.

Jegorova, Marija, Chaitanya Kaul, Charlie Mayor, Alison Q. O'Neil, Alexander Weir, Roderick Murray-Smith, and Sotirios A. Tsaftaris. "Survey: Leakage and Privacy at Inference Time." arXiv preprint arXiv:2107.01614 (2022). Accessed October 31, 2024. https://arxiv.org/pdf/2107.01614.

Jeong, Soyeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. “Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity.” arXiv.Org. March 21, 2024. https://arxiv.org/abs/2403.14403.

Jerry Liu. LlamaIndex, November 2022.

Jin, Jiajie, Yutao Zhu, Xinyu Yang, Chenghao Zhang, and Zhicheng Dou. 2024. “FlashRAG: A Modular Toolkit for Efficient Retrieval-Augmented Generation Research.” *arXiv (Cornell University)*, May. https://doi.org/10.48550/arxiv.2405.13576.

Jobanputra, Sara Zanzottera, Silvano Cerza, Vladimir Blagojevic, Thomas Stadelmann, Tanay Soni, and Sebastian Lee. Haystack: the end-to-end NLP framework for pragmatic builders,November 2019.

Johri, Prashant, Khatri, Sunil K., Al-Taani, Ahmad T., Sabharwal, Munish, Suvanov, Shakhzod, and Kumar, Avneesh. *Lecture Notes in Networks and Systems*. ICCIN 2020. Springer, 2021.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. "Scaling Laws for Neural Language Models." arXiv.Org. January 23, 2020. https://arxiv.org/abs/2001.08361.

Katz, Daniel Martin, Corinna Coupette, Janis Beckedorf, and Dirk Hartung. 2020. "Complex societies and the growth of the law." *Scientific Reports* 10 (1). https://doi.org/10.1038/s41598-020-73623-x.

Keskar, Nitish Shirish, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. "CTRL: A Conditional Transformer Language Model for Controllable Generation." arXiv.Org. September 11, 2019. https://arxiv.org/abs/1909.05858.

Khashabi, Daniel, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. "Looking Beyond the Surface: A Challenge Set for Reading Comprehension over Multiple Sentences." *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.*, January. https://doi.org/10.18653/v1/n18-1023.

Kratzwald, Bernhard, and Stefan Feuerriegel. 2018. “Adaptive Document Retrieval for Deep Question Answering.” arXiv.Org. August 20, 2018. https://arxiv.org/abs/1808.06528.

Lai, Jinqi, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and Philip S. Yu. 2024. “Large language models in law: A survey.” *AI Open,* October. https://doi.org/10.1016/j.aiopen.2024.09.002.

“LangChain.” n.d. https://www.langchain.com/.

Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, et al. 2020. “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.” arXiv.Org. May 22, 2020. https://arxiv.org/abs/2005.11401.

“Lexis+ AI Delivers Hallucination-Free Linked Citations.” n.d. Community. https://www.lexisnexis.com/community/insights/legal/b/product-features/posts/how-lexis-ai-delivers-hallucination-free-linked-legal-citations.

Llama.com. n.d. "Homepage." Accessed December 4, 2024. https://www.llama.com.

Lu, Ximing, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. “QUARK: Controllable Text Generation with Reinforced Unlearning.” December 6, 2022. https://proceedings.neurips.cc/paper_files/paper/2022/hash/b125999bde7e80910cbdbd323087df8f-Abstract-Conference.html.

Lu, Yu, Junwei Bao, Yan Song, Zichen Ma, Shuguang Cui, Youzheng Wu, and Xiaodong He. 2021. “RevCore: Review-augmented Conversational Recommendation.” arXiv.Org. June 2, 2021. https://arxiv.org/abs/2106.00957.

Luo, Hongyin, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. “SAIL: Search-Augmented Instruction Learning.” arXiv.Org. May 24, 2023. https://arxiv.org/abs/2305.15225.

Ma, Chang, Haiteng Zhao, Lin Zheng, Jiayi Xin, Qintong Li, Lijun Wu, Zhihong Deng, et al. 2024. “Retrieved Sequence Augmentation for Protein Representation Learning.” *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, January, 1738–67. https://doi.org/10.18653/v1/2024.emnlp-main.104.

Ma, Xinbei, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. “Query Rewriting in Retrieval-Augmented Large Language Models.” *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, January. https://doi.org/10.18653/v1/2023.emnlp-main.322.

Magesh, Varun, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D. Manning, and Daniel E. Ho. 2024. “Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools.” arXiv.Org. May 30, 2024. https://arxiv.org/abs/2405.20362.

Malte Pietsch, Timo Möller, Bogdan Kostic, Julian Risch, Massimiliano Pippi, Mayank

Martin-Short, Robert. 2024. “A Visual Exploration of Semantic Text Chunking - Towards Data Science.” *Medium*, November 16, 2024. https://towardsdatascience.com/a-visual-exploration-of-semantic-text-chunking-6bb46f728e30.

Mistral AI. 2024. “Mistral NeMo.” Mistral AI | Frontier AI in Your Hands. December 5, 2024. https://mistral.ai/news/mistral-nemo/.

Muglich, Darius, De Witt Christian Schroeder, Van Der Pol Elise, Shimon Whiteson, and Jakob Foerster. 2022. “Equivariant Networks for Zero-Shot Coordination.” arXiv.Org. October 21, 2022. https://arxiv.org/abs/2210.12124.

Naveed, Humza, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2023. “A Comprehensive Overview of Large Language Models.” arXiv.Org. July 12, 2023. https://arxiv.org/abs/2307.06435.

Nidhiworah. 2024a. “Chroma DB- Introduction - Nidhiworah - Medium.” *Medium*, November 24, 2024. https://medium.com/@nidhiworah02/chroma-db-introduction-25718915bae6.

Orr, Einat, PhD. 2024. “Best 16 Vector Databases for 2024 [Top Picks].” Git For Data - lakeFS. April 26, 2024. https://lakefs.io/blog/12-vector-databases-2023/.

Ovadia, Oded, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2023a. “Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLMs.” arXiv.Org. December 10, 2023. https://arxiv.org/abs/2312.05934.

Padiu, Bogdan, Radu Iacob, Traian Rebedea, and Mihai Dascalu. 2024. “To What Extent Have LLMs Reshaped the Legal Domain So Far? A Scoping Literature Review.” *Information* 15 (11): 662. https://doi.org/10.3390/info15110662.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. “BLEU.” *Association for Computing Memory Digital Library*, January, 311. https://doi.org/10.3115/1073083.1073135.

Parthasarathy, Venkatesh Balavadhani, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. 2024. “The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities.” arXiv.Org. August 23, 2024. https://arxiv.org/abs/2408.13296v1.

Peng, Boci, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. “Graph Retrieval-Augmented Generation: A Survey.” arXiv.Org. August 15, 2024. https://arxiv.org/abs/2408.08921.

Pipitone, Nicholas, and Ghita Houir Alami. 2024a. “LegalBench-RAG: A Benchmark for Retrieval-Augmented Generation in the Legal Domain.” arXiv.Org. August 19, 2024. https://arxiv.org/abs/2408.10343.

Pires, Telmo, Eva Schlinger, and Dan Garrette. 2019. “How multilingual is Multilingual BERT?” arXiv.Org. June 4, 2019. https://arxiv.org/abs/1906.01502.

Portuguese Connection Language School. n.d. “How hard is it to learn portuguese?” https://www.learnportugueseinlisbon.com/blog/is-portuguese-a-hard-language-to-learn.

“Portuguese Culture and Language.” n.d. Embassy of Portugal to the United States of America. https://washingtondc.embaixadaportugal.mne.gov.pt/en/about-portugal/portuguese-culture-and-language#:~:text=Portuguese%20is%20currently%20the%20fifth,until%201999%2C%20and%20in%20Goa.

Posner, Rebecca, and Marius Sala. 2024. “Portuguese language | Origin, History, Grammar, & Speakers.” Encyclopedia Britannica. November 19, 2024. https://www.britannica.com/topic/Portuguese-language.

Qu, Renyi, Ruixuan Tu, and Forrest Bao. 2024a. “Is Semantic Chunking Worth the Computational Cost?” arXiv.Org. October 16, 2024. https://arxiv.org/abs/2410.13070.

Radford, Alec, Karthik Narasimhan, Tim Salimans, Ilya Sutskever. 2018. “Improving Language Understanding by Generative Pre-Training.” *OpenAI*. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

Ramprasad, Akshara, and P. Sivakumar. 2024. “Context-Aware Summarization for PDF Documents using Large Language Models.” *2024 International Conference on Expert*

Clouds and Applications (ICOECA) 1 (April): 186–91. https://doi.org/10.1109/icoeca62351.2024.00044.

Re, Richard M., and Alicia Solow-Niederman. 2019. "Developing Artificially Intelligent Justice." *Stanford Technology Law Review*, May.

Renze, Matthew, and Erhan Guven. 2024. "Self-Reflection in LLM Agents: Effects on Problem-Solving Performance." arXiv.Org. May 5, 2024. https://arxiv.org/abs/2405.06682.

Ridoy, Shahriyar Zaman, Jannat Sultana, Zinnat Fowzia Ria, Mohammed Arif Uddin, Md Hasibur Rahman, and Rashedur M. Rahman. 2024. "An Efficient Text Cleaning Pipeline for Clinical Text for Transformer Encoder Models." IEEE Conference Publication | IEEE Xplore. August 29, 2024. https://ieeexplore.ieee.org/document/10705199.

Robertson, Stephen, and Hugo Zaragoza. 2009. "The Probabilistic Relevance Framework: BM25 and Beyond." *Foundations and Trends® in Information Retrieval* 3 (4): 333–89. https://doi.org/10.1561/1500000019.

Santhosh, Sthanikam. 2023. "Understanding BLEU and ROUGE score for NLP evaluation." *Medium*, April 17, 2023. https://medium.com/@sthanikamsanthosh1994/understanding-bleu-and-rouge-score-for-nlp-evaluation-1ab334ecadcb.

Shabbir, Sheeza. 2023. "Text Cleaning in NLP: Libraries, Techniques, and How to Get Started." *Medium*, October 25, 2023. https://medium.com/@datascientist_SheezaShabbir/text-cleaning-in-nlp-libraries-techniques-and-how-to-get-started-8c7c7e8ba7cf.

Shakshuki, Elhadi, and Malcolm Reid. 2015. “Multi-Agent System Applications in Healthcare: Current Technology and Future Roadmap.” *Procedia Computer Science* 52 (January): 252–61. https://doi.org/10.1016/j.procs.2015.05.071.

Shuster, Kurt, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. “Retrieval Augmentation Reduces Hallucination in Conversation.” arXiv.Org. April 15, 2021. https://arxiv.org/abs/2104.07567.

State v. Loomis: Wisconsin Supreme Court Requires Warning Before Use of Algorithmic Risk Assessments in Sentencing." Harvard Law Review 130, no. 5 (2017): 1530. Accessed October 31, 2024. https://harvardlawreview.org/print/vol-130/state-v-loomis/.

Sturua, Saba, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, et al. 2024. “jina-embeddings-v3: Multilingual Embeddings With Task LoRA.” arXiv.Org. September 16, 2024. https://arxiv.org/abs/2409.10173.

Tewari, Amit. 2024. “LegalPro-BERT: Classification of Legal Provisions by Fine-tuning BERT Large Language Model.” arXiv.Org. April 15, 2024. https://arxiv.org/abs/2404.10097.

Thirunavukarasu, Arun James, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. “Large language models in medicine.” *Nature Medicine* 29 (8): 1930–40. https://doi.org/10.1038/s41591-023-02448-8.

Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, et al. 2023b. “Llama 2: Open Foundation and Fine-Tuned Chat Models.” arXiv.Org. July 18, 2023. https://arxiv.org/abs/2307.09288.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, et al. 2023a. “LLaMA: Open and Efficient

Foundation Language Models." arXiv.Org. February 27, 2023. https://arxiv.org/abs/2302.13971.

Upadhyay, Prashant, Rishabh Agarwal, Sumeet Dhiman, Abhinav Sarkar, and Saumya Chaturvedi. 2024. "A comprehensive survey on answer generation methods using NLP." *Natural Language Processing Journal* 8 (July): 100088. https://doi.org/10.1016/j.nlp.2024.100088.

Uptrain-Ai. n.d. "GitHub - uptrain-ai/uptrain: UpTrain is an open-source unified platform to evaluate and improve Generative AI applications. We provide grades for 20+ preconfigured checks (covering language, code, embedding use-cases), perform root cause analysis on failure cases and give insights on how to resolve them." GitHub. https://github.com/uptrain-ai/uptrain.

Vasdani, Tara. 2020. "Robot justice: China's use of Internet courts | Lexisnexis Canada." n.d. https://www.lexisnexis.ca/en-ca/ihc/2020-02/robot-justice-chinas-use-of-internet-courts.page.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention is All you Need." *arXiv (Cornell University)* 30 (June): 5998–6008. https://arxiv.org/pdf/1706.03762v5.

Wachter, Sandra, Brent Mittelstadt, and Luciano Floridi. "Why a Right to Explanation of Automated Decision-Making Does Not Exist in the General Data Protection Regulation." International Data Privacy Law 7, no. 2 (2017): 76-99. Accessed October 31, 2024. https://doi.org/10.1093/idpl/ipx005.

Wallace, Eric, Tony Z. Zhao, Shi Feng, and Sameer Singh. "Concealed Data Poisoning Attacks on NLP Models." In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language

Technologies, 139-150. Association for Computational Linguistics, 2021. Accessed October 21, 2024. https://aclanthology.org/2021.naacl-main.13.pdf.

Wang, Xiaohua, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, et al. 2024. “Searching for Best Practices in Retrieval-Augmented Generation.” *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, January, 17716–36. https://doi.org/10.18653/v1/2024.emnlp-main.981.

Wang, Zichao, Weili Nie, Zhuoran Qiao, Chaowei Xiao, Richard Baraniuk, and Anima Anandkumar. 2022. “Retrieval-based Controllable Molecule Generation.” arXiv.Org. August 23, 2022. https://arxiv.org/abs/2208.11126.

Wang, Zichong, Zhibo Chu, Thang Viet Doan, Shiwen Ni, Min Yang, and Wenbin Zhang. 2024. “History, development, and principles of large language models: an introductory survey.” *AI And Ethics*, October. https://doi.org/10.1007/s43681-024-00583-7.

Wei, Fusheng, Robert Keeling, Nathaniel Huber-Fliflet, Jianping Zhang, Adam Dabrowski, and Jingchao Yang. 2023. “Empirical Study of LLM Fine-Tuning for Text Classification in Legal Document Review.” IEEE Conference Publication | IEEE Xplore. December 15, 2023. https://ieeexplore.ieee.org/document/10386911.

Weiss, Gerhard. 2000. *Multiagent Systems : A Modern Approach to Distributed Artificial Intelligence*. http://ci.nii.ac.jp/ncid/BA40989172.

Welbl, Johannes, Pontus Stenetorp, and Sebastian Riedel. 2018. “Constructing Datasets for Multi-hop Reading Comprehension Across Documents.” *Transactions of the Association for Computational Linguistics* 6 (December): 287–302. https://doi.org/10.1162/tacl_a_00021.

Wellen, Serena. 2024. “Hallucination-Free Linked Legal Citations.” Https://Www.Lexisnexis.Com.Au/. April 30, 2024.

https://www.lexisnexis.com.au/en/insights-and-analysis/practice-intelligence/2024/hallucination-free-linked-legal-citations.

Wiggins, Walter F., and Ali S. Tejani. 2022. “On the Opportunities and Risks of Foundation Models for Natural Language Processing in Radiology.” *Radiology Artificial Intelligence* 4 (4). https://doi.org/10.1148/ryai.220119.

Wooldridge, Michael. 2002. *An Introduction to MultiAgent Systems*. http://www.gbv.de/dms/hebis-darmstadt/toc/98534017.pdf.

Woyera. 2023a. “Pinecone vs. Chroma: The Pros and Cons - Woyera - Medium.” *Medium*, July 22, 2023. https://medium.com/@woyera/pinecone-vs-chroma-the-pros-and-cons-2b0b7628f48f.

Xi, Yunjia, Jianghao Lin, Weiwen Liu, Xinyi Dai, Weinan Zhang, Rui Zhang, Ruiming Tang, and Yong Yu. 2022. “A Bird’s-eye View of Reranking: from List Level to Page Level.” arXiv.Org. November 17, 2022. https://arxiv.org/abs/2211.09303.

Xia, Yuchen, Jiho Kim, Yuhan Chen, Haojie Ye, Souvik Kundu, and Cong Callie Hao. 2024. “Understanding the Performance and Estimating the Cost of LLM Fine-Tuning.” IEEE Conference Publication | IEEE Xplore. September 15, 2024. https://ieeexplore.ieee.org/abstract/document/10763668.

Yahouni, Z., A. Ladj, F. Belkadi, O. Meski, and M. Ritou. 2021. “A smart reporting framework as an application of multi-agent system in machining industry.” *International Journal of Computer Integrated Manufacturing* 34 (5): 470–86. https://doi.org/10.1080/0951192x.2021.1901312.

Yan, Shi-Qi, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. “Corrective Retrieval Augmented Generation.” arXiv.Org. January 29, 2024. https://arxiv.org/abs/2401.15884.

Yang, Mengjiao, Dale Schuurmans, Pieter Abbeel, and Ofir Nachum. 2022. “Chain of Thought Imitation with Procedure Cloning.” arXiv.Org. May 22, 2022. https://arxiv.org/abs/2205.10816.

Yang, Sophia, PhD. 2023. “Advanced RAG 01: Small-to-Big Retrieval - Towards Data Science.” *Medium*, November 5, 2023. https://towardsdatascience.com/advanced-rag-01-small-to-big-retrieval-172181b396d4.

Yao, Shunyu, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022a. “ReAct: Synergizing Reasoning and Acting in Language Models.” arXiv.Org. October 6, 2022. https://arxiv.org/abs/2210.03629.

Yepes, Antonio Jimenoo, Yao You, Jan Milczek, Sebastian Laverde, and Renyu Li. 2024. “Financial Report Chunking for Effective Retrieval Augmented Generation.” arXiv.Org. February 5, 2024. https://arxiv.org/abs/2402.05131.

Yoran, Ori, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. “Making Retrieval-Augmented Language Models Robust to Irrelevant Context.” arXiv.Org. October 2, 2023. https://arxiv.org/abs/2310.01558.

Yu, Hao, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2024. “Evaluation of Retrieval-Augmented Generation: A Survey.” arXiv.Org. May 13, 2024. https://arxiv.org/abs/2405.07437.

Zhang, Boyu, Hongyang Yang, Tianyu Zhou, Muhammad Ali Babar, and Xiao-Yang Liu. 2023. “Enhancing Financial Sentiment Analysis via Retrieval Augmented Large Language Models.” *Proceedings of the Fourth ACM International Conference on AI in Finance, 349–356.*, November. https://doi.org/10.1145/3604237.3626866.

Zhao, Wayne Xin, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, et al. 2023. “A Survey of Large Language Models.” arXiv.Org. March 31, 2023. https://arxiv.org/abs/2303.18223.

Appendices

You are a Portuguese legal expert specializing in the analysis of Acórdãos (Court Decisions). Based on the link {link}, your task is to generate up to 3 open questions per Acórdão..

Questions should:

1. Be relevant to legal professionals, analyzing the case, the legal basis and the possible implications.
2. Avoid superficial interpretations; Instead, they should focus on aspects such as the court's interpretation of the law, the main arguments presented, and the broader legal impact.
3. Be written in European Portuguese.
4. Each question should include:

• A detailed question derived from the text of the 'Texto Integral'.
• A concise answer, directly related to the question.
• The specific excerpt from the text of the 'Texto Integral' used to create the question and answer.
• Context metadata: URL: The link to the Judgment, Proceedings: The number of the proceeding corresponding to the Judgment.
• The output must be in CSV format with the following columns: URL: The link to the Judgment, Case: The case number, Question: The question asked, Answer: The answer to the question, Reference: The excerpt from the text of the 'Integral Decision' used as basis for question and answer.

Additional rules:
• Make sure questions are directed to substantive legal issues that are useful for legal professionals to analyze.

Appendix 1 - Prompt template for generating questions from ‘‘acórdãos’’ (Adapted from Portuguese).

Read the following articles:
{article}

Create, for each article, a specific essay question about the main concept or idea addressed, without directly mentioning the article and WITHOUT DEPARTING FROM THE SCOPE OF THE ARTICLE.

Provide an objective answer to the essay question and the reference context.

Example entry: Text: "Shift work is considered to be any organization of teamwork in which workers successively occupy the same workstations, at a certain pace, including rotational, continuous or discontinuous, and may perform the work at different times. in a given period of days or weeks."

Example output:

Essay question: Explain how shift work is organized and its possible impacts on the worker.
Essay answer: Shift work is organized with alternating roles and schedules within a team, generating flexibility, but also possible adaptation and health challenges.

Appendix 2 - Prompt template for generating questions from "Artigos" (Adapted from Portuguese).

You will compare two pieces of text: the RETRIEVED CONTEXT and the GROUND TRUTH CONTEXT.
Your task is to evaluate how well the RETRIEVED CONTEXT aligns with the GROUND TRUTH CONTEXT and assign a score between 0 and 1 based on the criteria below:

Evaluation Criteria:
1. Assess whether the RETRIEVED CONTEXT includes the GROUND TRUTH CONTEXT entirely or partially.
2. Focus solely on the inclusion of concepts and factual alignment. Do not penalize for additional unrelated content unless it disrupts understanding.

Scoring Guide:
- 1: The RETRIEVED CONTEXT fully includes the GROUND TRUTH CONTEXT.
- 0.5: Some concepts from the GROUND TRUTH CONTEXT are included, but it is incomplete.
- 0: The GROUND TRUTH CONTEXT is not included in the RETRIEVED CONTEXT at all.

Appendix 3 - Prompt template for evaluating retrieval.

You are a teacher grading a quiz. You will be given a QUESTION, a GROUND TRUTH (correct) ANSWER, and a STUDENT ANSWER. Your task is to assign a score between 0 and 5 based on the criteria below:

Language Context:
The QUESTION and GROUND TRUTH ANSWER are in European Portuguese, but the STUDENT ANSWER may be in a different language. Translate as needed before evaluation.

Evaluation Criteria:
*1. Grade based *only on the factual accuracy* of the STUDENT ANSWER relative to the GROUND TRUTH ANSWER.*
2. Penalize any conflicting or contradictory statements in the STUDENT ANSWER.
*3. Additional information in the STUDENT ANSWER is acceptable *only if it does not include unverified references* or refer to legal articles ("artigos") unless explicitly mentioned in the GROUND TRUTH ANSWER.*

Scoring Guide:
- 5: Fully accurate, no contradictions, and aligns completely with the criteria.
- 4: Mostly accurate but includes minor unverified references (-1 penalty per unverified reference).
- 3: Moderately accurate, with noticeable gaps or slight contradictions.
- 2: Partially accurate, significant omissions, or contradictions.
- 1: Barely accurate, mostly incorrect with significant issues.
- 0: Completely inaccurate or fails to meet the criteria.

Appendix 4 - Prompt Template for evaluating overall answer quality.

You are a teacher grading a quiz. You will be provided with a RETRIEVED CONTEXT and a STUDENT ANSWER. Your task is to assess how well the STUDENT ANSWER adheres to the RETRIEVED CONTEXT and assign a score between 0 and 5 based on the criteria below:

Language Context:
- The RETRIEVED CONTEXT is always in Portuguese, but the STUDENT ANSWER might be in English or Portuguese. Translate as needed and do not penalize for language differences.

Evaluation Criteria:
*1. Verify that the STUDENT ANSWER is *grounded in the RETRIEVED CONTEXT* and does not introduce significant unrelated content.*
2. Focus solely on adherence to the RETRIEVED CONTEXT when grading.
3. Penalize any significant unverified or unrelated references slightly.

Scoring Guide:
- 5: Fully grounded in the RETRIEVED CONTEXT with no issues.
- 4: Minor deviations or unverified references (-1 penalty per instance).
- 3: Moderate alignment but includes noticeable unrelated content or minor contradictions.
- 2: Limited alignment with the RETRIEVED CONTEXT, with significant issues.
- 1: Barely aligned, mostly incorrect, with severe issues.
- 0: Completely unaligned or fails to meet the criteria.

Appendix 5 - Prompt template for assessing hallucination in generated answers.

ONLY ANSWER IN PORTUGUESE FROM PORTUGAL
You are a legislative counsel of Portugal, specialized in Portuguese cases of the Constitutional Court of Portugal, you will get context and a question to ground your answer on.
If you do not know, STATE CLEARLY YOU DO NOT KNOW.

{context}

Question: {question}
Generated Answer:

Appendix 6 - Prompt template for LLM in answer generation.

Strategy	Chunk size	Overlap	Number of Chunks	Time to Embed (s)
TokenTextSplitter	128	0	6490	547
		20	7507	573
	256	0	3417	558
		20	3598	570
	512	0	1863	566
		20	1897	626
	1024	0	1122	529
		20	1126	529
RecursiveCharacterTextSplitter	128	0	21319	661
		20	21671	708
	256	0	9748	613
		20	9936	655
	512	0	4639	671
		20	4668	717
	1024	0	2383	638
		20	2392	674
Semantic Chunker	-	-	1328	796

Appendix 7- Results for time (in seconds) for the generation of chunks and embeddings under different strategies.

Strategy	Chunk size	Overlap	Single		Multi	
			Llama	*Openai*	*Llama*	*Openai*
TokenTextSplitter	128	0	0,46	0,46	0,38	0,35
		20	0,46	0,46	0,31	0,35
	256	0	0,54	0,50	0,35	0,35
		20	0,65	0,58	0,35	0,35
	512	0	0,58	0,54	0,38	0,38
		20	0,65	0,65	0,46	0,54
	1024	0	0,62	0,58	0,42	0,50
		20	0,58	0,54	0,42	0,54
RecursiveCharacterTextSplitter	128	0	0,42	0,42	0,38	0,31
		20	0,42	0,38	0,35	0,31
	256	0	0,50	0,50	0,35	0,35
		20	0,46	0,50	0,42	0,38
	512	0	0,46	0,42	0,38	0,38
		20	0,42	0,42	0,42	0,46
	1024	0	0,62	0,58	0,50	0,50
		20	0,65	0,58	0,46	0,42
SemanticChunker	-	-	0,42	0,46	0,38	0,35

Appendix 8 - Retrieval scores under different chunking strategy pairs.

Strategy	Chunk size	Overlap	Single		Multi	
			Llama	*Openai*	*Llama*	*Openai*
TokenTextSplitter	128	0	0,46	0,83	0,29	0,88
		20	0,55	0,83	0,45	0,88
	256	0	0,40	0,86	0,38	0,77
		20	0,29	0,83	0,37	0,88
	512	0	0,51	0,88	0,35	0,80
		20	0,52	0,82	0,45	0,82
	1024	0	0,43	0,89	0,29	0,85
		20	0,49	0,85	0,42	0,85
RecursiveCharacterTextSplitter	128	0	0,38	0,60	0,42	0,74
		20	0,43	0,63	0,32	0,43
	256	0	0,42	0,66	0,29	0,77
		20	0,52	0,74	0,34	0,72
	512	0	0,48	0,91	0,35	0,86
		20	0,43	0,71	0,51	0,80
	1024	0	0,51	0,74	0,34	0,72
		20	0,49	0,83	0,35	0,91
SemanticChunker	-	-	0,38	0,72	0,26	0,80

Appendix 9 - Hallucination scores for the different chunking strategy pairs.

Strategy	Chunk size	Overlap	Single		Multi	
			Llama	*Openai*	*Llama*	*Openai*
TokenTextSplitter	128	0	0,42	0,45	0,34	0,60
		20	0,52	0,52	0,38	0,62
	256	0	0,42	0,54	0,40	0,58
		20	0,42	0,58	0,43	0,63
	512	0	0,40	0,62	0,38	0,55
		20	0,40	0,52	0,42	0,65
	1024	0	0,37	0,62	0,38	0,60
		20	0,42	0,58	0,43	0,58
RecursiveCharacterTextSplitter	128	0	0,43	0,46	0,35	0,49
		20	0,37	0,42	0,35	0,35
	256	0	0,38	0,46	0,31	0,65
		20	0,46	0,48	0,37	0,63
	512	0	0,48	0,49	0,32	0,62
		20	0,32	0,51	0,38	0,60
	1024	0	0,35	0,52	0,46	0,58
		20	0,42	0,55	0,40	0,58
SemanticChunker	-	-	0,46	0,45	0,32	0,60

Appendix 10 - Overall answer quality scores for the different chunking strategy pairs.

Strategy	Chunk size	Overlap	Single		Multi	
			Llama	*Openai*	*Llama*	*Openai*
TokenTextSplitter	128	0	13,49	11,36	3,60	4,27
		20	9,81	8,05	4,12	3,60
	256	0	12,45	11,11	4,39	3,99
		20	13,71	11,67	4,47	4,82
	512	0	9,76	11,80	4,64	4,84
		20	15,81	11,30	3,94	4,61
	1024	0	16,94	14,59	5,47	4,66
		20	13,53	18,31	4,83	4,88
RecursiveCharacter TextSplitter	128	0	15,32	11,62	3,16	3,16
		20	13,22	15,45	2,95	2,18
	256	0	11,10	10,45	2,63	4,44
		20	10,70	10,03	3,38	3,63
	512	0	15,18	16,43	5,03	4,40
		20	11,13	9,29	4,00	4,47
	1024	0	11,82	15,91	4,47	5,00
		20	10,49	16,65	4,91	5,25
SemanticChunker			18,79	22,41	3,99	4,78
Average			13,13	13,32	4,12	4,29

Appendix 11 - Results for time (in seconds) for the generation of answers under different chunking strategy pairs.

Strategy	Chunk size	Overlap	Single		Multi	
			Llama	*Openai*	*Llama*	*Openai*
TokenTextSplitter	128	0	0	2	0	0
		20	0	1	0	0
	256	0	0	1	0	0
		20	0	1	0	0
	512	0	1	0	0	0
		20	0	0	0	0
	1024	0	0	0	0	0
		20	0	0	0	0
RecursiveCharacterTextSplitter	128	0	0	4	1	2
		20	0	4	0	6
	256	0	0	4	0	0
		20	0	2	0	0
	512	0	0	1	0	0
		20	0	2	0	0
	1024	0	1	2	0	0
		20	0	1	0	0
SemanticChunker	-	-	0	2	0	0

Appendix 12 - Number of unanswered questions per chunking strategy pair.

You are an expert grader in the legal exploratory domain, tasked with evaluating the complexity of a given question. Classify the question as either 'simple' or 'complex' based on the following criteria:

- 'simple': The question requires a direct answer with a single reasoning step, often involving straightforward retrieval of information from a given text. Examples include defining legal terms, summarizing a case's fundaments, or explaining a specific aspect without requiring further inference or interpretation.

- 'complex': The question requires multiple reasoning steps, integration of information, or interpretative analysis. This includes questions that involve reasoning court decisions, resolving ambiguities, or synthesizing interpretations.

Return only the classification as 'simple' or 'complex' in your response.

Question: {question}

Appendix 13 - Prompt for assessing query complexity.

You are a grader assessing the relevance of a retrieved document to a user question in the legal context. It does not need to be a stringent test. The goal is to filter out erroneous retrievals.

If the document contains keyword(s) or semantic meaning related to the user question, grade it as relevant. Give a binary score 'yes' or 'no' score to indicate whether the document is relevant to the question.

The documents and questions presented to you are in portuguese.

Retrieved document: {document}

Question: {question}

Appendix 14- Prompt for assessing document relevance.

You are a grader tasked with evaluating whether a generated answer by an LLM is grounded in or supported by a provided set of retrieved facts.

- 'Yes': The answer is fully supported by and aligned with the information in the retrieved facts. There are no contradictions or unsupported claims.

- 'No': The answer includes information not found in the retrieved facts, contradicts the facts, or lacks sufficient grounding.

Provide only the binary score: 'yes' or 'no'.

Set of facts: {documents}

LLM Generation: {generation}

Appendix 15- Prompt for Hallucination checking.

ONLY ANSWER IN PORTUGUESE FROM PORTUGAL

You are a legislative counsel of Portugal, specialized in Portuguese cases of the Constitutional Court of Portugal. The previously provided answer may contain hallucination. Be sure to reformulate to address the question and context properly.

Question: {question}

Previous answer: {generation}

Context: {documents}

Answer:

Appendix 16- Prompt for answer regeneration.

You are a grader tasked with evaluating whether an answer sufficiently addresses or resolves a given question.

- 'Yes': The answer provides a response that directly or adequately addresses the question, even if not perfectly comprehensive.

- 'No': The answer fails to address the question in any meaningful way or is entirely irrelevant.

Provide only the binary score: 'yes' or 'no'.

Question: {question}

LLM Generation: {generation}

Appendix 17 - Prompt for answer relevancy check.

You are a question re-writer tasked with optimizing input questions for effective vectorstore retrieval.

- Analyze the input question to understand its underlying semantic intent and meaning.

- Rewrite the question to be more specific, unambiguous, and aligned with the context of vectorstore retrieval.

- Ensure the rewritten question captures the core intent while removing vagueness or irrelevant details.

- Return only the optimized question.

Initial question: {question}

Improved version:

Appendix 18 - Prompt for query rewriting.

Configuration	Nodes
complexity	1,3
grade_docs	2,3
hallucination	3,4 + Regeneration Prompt
query_rewrite	6,3
self_rag	2,3,4,5,6 + Regeneration Prompt
crag	1,2,3
complete	All nodes + Regeneration Prompt

Appendix 19- Configuration setup for experimentation (nodes are exemplified in figure 14).

Single OpenAI

Collection	Answer	Retrieval	Hallucination
512_0	0,62	0,54	0,88
512_20	0,52	0,65	0,82
Average	0,57	0,60	0,85

Multi OpenAI

Collection	Answer	Retrieval	Hallucination
512_0	0,55	0,38	0,80
512_20	0,65	0,54	0,82
Average	0,60	0,46	0,81

Appendix 20- Baseline OpenAI single & multi questions (scores from 0 (worse) to 1 (best)).

Single MistralAI

Collection	Answer	Retrieval	Hallucination
512_0	0,49	0,54	0,82
512_20	0,55	0,65	0,91
Average	0,52	0,60	0,86

Multi MistralAI

Collection	Answer	Retrieval	Hallucination
512_0	0,55	0,38	0,80
512_20	0,35	0,35	0,54
Average	0,45	0,37	0,67

Appendix 21- Baseline MistralAI - single & multi questions (scores from 0 (worse) to 1 (best)).

OpenAI

Configuration	Answer	Retrieval	Hallucination
complexity	0,60	0,57	0,77
grade_docs	0,58	0,51	0,67
hallucination	0,64	0,55	0,80
query_rewrite	0,60	0,54	0,70
self_rag	0,62	0,56	0,76
crag	0,59	0,58	0,75
complete	0,62	0,57	0,80

MistralAI

Configuration	Answer	Retrieval	Hallucination
complexity	0,51	0,49	0,64
grade_docs	0,55	0,54	0,68
hallucination	0,57	0,56	0,67
query_rewrite	0,48	0,39	0,55
self_rag	0,59	0,54	0,70
crag	0,48	0,51	0,58
complete	0,55	0,50	0,67

Appendix 22- LLM comparison - OpenAI & MistralAI (average of evaluation results).

OpenAI (% change to baseline)

Configuration	Answer	Retrieval	Hallucination
complexity	0,04	**0,08**	-0,07
grade_docs	0,00	-0,03	-0,19
hallucination	**0,10**	0,04	-0,03
query_rewrite	0,03	0,02	-0,15
self_rag	0,05	0,06	-0,08
crag	0,01	**0,09**	-0,10
complete	**0,06**	**0,08**	-0,03

MistralAI (% change to baseline)

Configuration	Answer	Retrieval	Hallucination
complexity	0,05	0,04	-0,17
grade_docs	**0,14**	**0,16**	-0,10
hallucination	**0,17**	**0,20**	-0,13
query_rewrite	-0,02	-0,13	-0,29
self_rag	**0,20**	**0,15**	-0,09
crag	0,00	0,07	-0,25
complete	**0,13**	0,05	-0,12

Appendix 23- LLM comparison - OpenAI & MistralAI (in % change to baseline).

Single-hop

average	Answer	Retrieval	Hallucination
complexity	0,58	0,59	0,78
grade_docs	0,58	0,58	0,71
hallucination	0,64	0,62	0,79
query_rewrite	0,56	0,50	0,70
self_rag	0,63	0,62	0,80
crag	0,52	0,63	0,73
complete	0,59	0,61	0,77

Multi-hop

average	Answer	Retrieval	Hallucination
complexity	0,53	0,47	0,63
grade_docs	0,56	0,47	0,64
hallucination	0,58	0,49	0,68
query_rewrite	0,52	0,43	0,56
self_rag	0,58	0,48	0,66
crag	0,56	0,45	0,59
complete	0,58	0,46	0,70

Appendix 24- Question type comparison - single & multi-hop (average of evaluation results).

Single-hop (% change to baseline)

average	Answer	Retrieval	Hallucination
complexity	0,06	-0,02	-0,08
grade_docs	0,06	-0,03	-0,17
hallucination	**0,17**	0,03	-0,08
query_rewrite	0,02	-0,16	-0,18
self_rag	**0,15**	0,03	-0,06
crag	-0,05	**0,06**	-0,14
complete	**0,08**	0,02	-0,10

Multi-hop (% change to baseline)

average	Answer	Retrieval	Hallucination
complexity	0,03	**0,14**	-0,16
grade_docs	**0,08**	**0,16**	-0,12
hallucination	**0,11**	**0,21**	-0,08
query_rewrite	-0,01	0,05	-0,25
self_rag	**0,10**	**0,17**	-0,11
crag	0,06	0,09	-0,21
complete	**0,11**	**0,12**	-0,05

Appendix 25- Question type comparison - single & multi-hop (in % change to baseline).

MistralAI Single-hop

Configuration	Average Time (s)
baseline	4,80
complexity	11,06
grade_docs	18,22
hallucination	14,70
query_rewrite	8,71
self_rag	28,78
crag	58,70
complete	72,30

Appendix 26- Average time taken for each configuration for single-hop questions with MistralAI.