Avaliação do Impacto de Diferentes Padrões Arquiteturais RAG em Domínios Jurídicos

Salvador Ludovico Paranhos[1], **Jonatas Novais Tomazini**[2], **Celso Gonçalves Camilo Junior**[3] **Sávio Salvarino Teles de Oliveira**[4]

[1]Instituto de Informática – Universidade Federal de Goiás (UFG)
Goiânia – GO – Brazil

{[1]tomazini, [2]salvadorludovico}@discente.ufg.br,

[3]celso@inf.ufg.br, [4]savioteles@ufg.br

Abstract. *This study evaluates the impact of different Retrieval-Augmented Generation (RAG) architectures in the legal context, focusing on the accuracy and relevance of responses in question-answering (Q&A) systems. Variations in query manipulation strategies, document retrieval, and relevance checks were investigated to analyze how these factors influence the quality of responses for legal queries. Multiple RAG architectures were implemented, along with a synthesizer module and an evaluator module to compare the efficiency of these patterns. The results indicate that RAG architecture performance varies significantly depending on the type of legal query, highlighting that understanding these dynamics is essential for optimizing Q&A systems within the legal domain.*

Resumo. *Este estudo avalia o impacto de diferentes arquiteturas de Retrieval-Augmented Generation (RAG) no contexto jurídico, com foco na precisão e relevância das respostas em sistemas de perguntas e respostas (Q&A). Foram investigadas variações em estratégias de manipulação de consultas, recuperação de documentos e verificações de relevância, analisando como essas influenciam a qualidade das respostas para consultas jurídicas. Diversas arquiteturas RAG foram implementadas, junto a um módulo sintetizador e um módulo avaliador para comparar a eficiência dos padrões. Os resultados indicam que o desempenho das arquiteturas RAG varia significativamente de acordo com o tipo de consulta jurídica e a compreensão dessas dinâmicas é essencial para otimizações em sistemas de Q&A no domínio jurídico.*

1. Introdução

A complexidade do sistema jurídico e sua importância para a sociedade demandam soluções tecnológicas eficientes que garantam acesso à informação e justiça. Grandes Modelos de Linguagem (LLMs) oferecem um potencial transformador, mas enfrentam a limitação do acesso direto a bancos de dados jurídicos específicos [Krasadakis et al. 2024]. Sistemas de Geração Aumentada por Recuperação (RAG) surgem como uma ferramenta para superar essa limitação, permitindo a integração destes dados internos ao processo de geração de texto pelas LLMs [Fan et al. 2024]. Advogados, juízes e pesquisadores podem se beneficiar da capacidade do RAG de sintetizar informações relevantes a partir de jurisprudência e legislação, agilizando a pesquisa jurídica e permitindo a construção de argumentos fundamentados.

Entretanto, a implementação de RAG no contexto jurídico apresenta desafios inerente às grandes bases de dados jurídicas, repletas de documentos longos e com linguagem altamente específica e complexa, o que torna desafiador a recuperação dos trechos relevantes para responder a uma consulta específica [Wiratunga et al. 2024]. Diversas arquiteturas RAG foram criadas e podem ser utilizadas para o contexto jurídico. Uma das primeiras abordagens, Naive RAG, busca integrar a recuperação de documentos a geração de respostas, oferecendo uma base para abordagens subsequentes que buscam refinar a precisão e relevância em sistemas de Q&A complexos [Lewis et al. 2020].

Uma abordagem que introduz a geração de hipóteses de resposta antes da recuperação pode aumentar a relevância dos documentos recuperados [Gao et al. 2022]. O *Corrective RAG* (CRAG) combina avaliação de contexto com reescrita de consultas e busca de contexto adicional [Yan et al. 2024, Ma et al. 2023]. Além disso, o Self-RAG verifica alucinações e busca checar a relevância da resposta para a consulta do usuário [Asai et al. 2023].

Recentemente, uma técnica expandiu a área ao utilizar expansão de consultas para gerar múltiplas consultas que exploram vários ângulos de um mesmo tema [Jagerman et al. 2023]. Utilizando-se desse conceito, a abordagem do RAG-Fusion introduziu o sistema de recuperação paralela de documentos a partir de múltiplas consultas e aplicação de reranking nos documentos recuperados para seleção dos contextos mais relevantes [Rackauckas 2024]. No contexto jurídico, o CBR-RAG [Wiratunga et al. 2024] apresenta uma solução que utiliza o raciocínio baseado em casos jurídicos para estruturar a recuperação na fase inicial do ciclo RAG, integrando o vocabulário de indexação para enriquecer as consultas com casos jurídicos contextualmente relevantes.

Apesar da variedade de arquiteturas RAG, a literatura ainda não apresenta estudos comparativos focados nas particularidades dos dados jurídicos. Este trabalho investiga o desempenho de diferentes arquiteturas RAG no domínio de documentos jurídicos do Tribunal de Contas do Estado de Goiás (TCE-GO), buscando identificar as abordagens mais eficazes para tarefas de perguntas e respostas (Q&A). Utilizamos um sistema de avaliação para comparar as arquiteturas e analisar sua adequabilidade e limitações em diferentes tipos de interação usuário-sistema.

2. Padrões Arquiteturais de Geração Aumentada por Recuperação

Este trabalho avalia cinco padrões arquiteturais de RAG: Naive RAG, Naive-HyDE, HyDE-CRAG, RAG-Fusion e Self-RAG. Cada arquitetura foi selecionada por suas características específicas e potencial para aprimorar a qualidade das respostas em sistemas de perguntas e respostas (Q&A) aplicados ao contexto jurídico.

2.1. Naive RAG

O padrão Naive RAG integra a recuperação de informações e geração de respostas em tarefas de NLP. Nesta abordagem, uma consulta inserida pelo usuário é transformada em uma representação vetorial por meio de modelos de *embedding*, permitindo a realização de uma busca por similaridade semântica na base de conhecimento, com o objetivo de recuperar informações contextualmente relevantes que aprimoram a geração de respostas. [Lewis et al. 2020]. É considerada a arquitetura mais simples por conter apenas as etapas de indexação, recuperação e geração do resultado final.

2.2. Naive RAG with HyDE (Hypothetical Document Embedding)

Este padrão segue o princípio do Naive RAG, porém adicionando a estratégia *Hypothetical Document Embeddings* (HyDE) *pré-retrieval*. A técnica *HyDE* é apresentada como uma estratégia para melhorar a recuperação de informações, criando uma ”resposta hipotética” gerada a partir da *consulta* do usuário antes da fase de recuperação. Essa resposta hipotética proporciona uma representação semântica mais detalhada e mais próxima dos documentos que contêm a resposta, aumentando a precisão na correspondência de documentos relevantes quando comparada ao uso direto da *consulta* original do usuário [Gao et al. 2022].

2.3. CRAG (Corrective RAG) with HyDE

O *CRAG* utiliza um avaliador para classificar documentos como “relevantes” ou “irrelevantes” [Yan et al. 2024]. Apenas os relevantes seguem para a geração, aumentando a precisão ao excluir dados irrelevantes. Caso nenhum documento seja relevante, realiza-se nova busca com a inclusão de HyDE como entrada para o *retriever*.

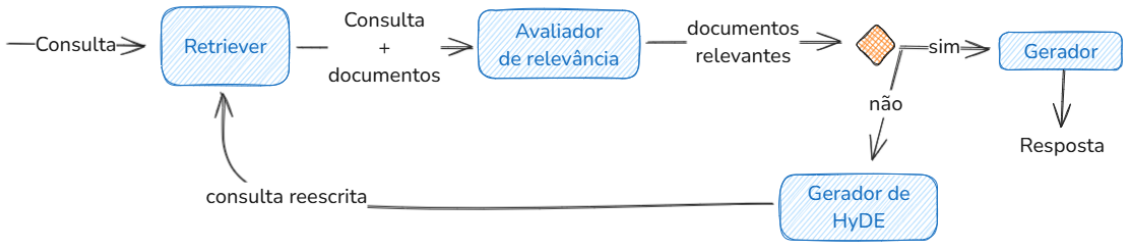

Figure 1. Padrão CRAG com técnica HyDE

2.4. Self-RAG

O Self-RAG (*Self-Reflective Retrieval-Augmented Generation*) utiliza ”tokens de reflexão” para iterar entre recuperação e avaliação, verificando alucinações e adequação da resposta à pergunta [Asai et al. 2023]. Inclui um analisador de relevância, um de fundamentação para evitar alucinações e um módulo de reescrita para ampliar consultas quando documentos relevantes não são encontrados.

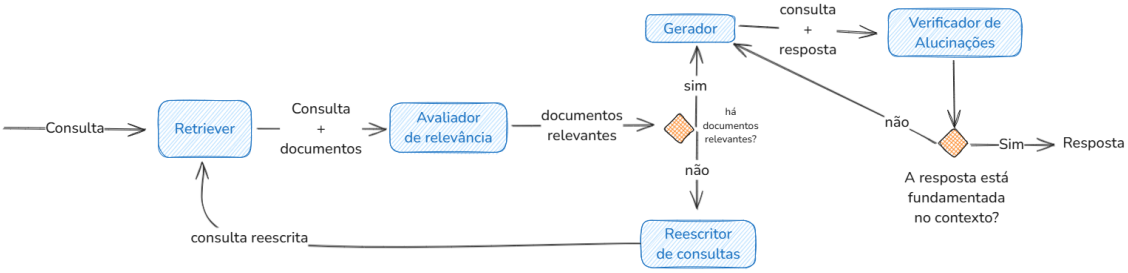

Figure 2. Descrição do padrão Self-RAG

2.5. RAG-Fusion

O padrão **RAG-Fusion** apresenta a técnica *Multiple Query Generation*, criando múltiplas consultas similares à original para gerar diversos ângulos de uma consulta, enriquecendo o contexto semântico antes da recuperação (figura 3). Após a recuperação, executa um ranqueamento nos documentos, otimizando a precisão semântica na resposta [Rackauckas 2024]. Essa técnica possibilita a seleção de documentos mais alinhados à intenção de consulta.

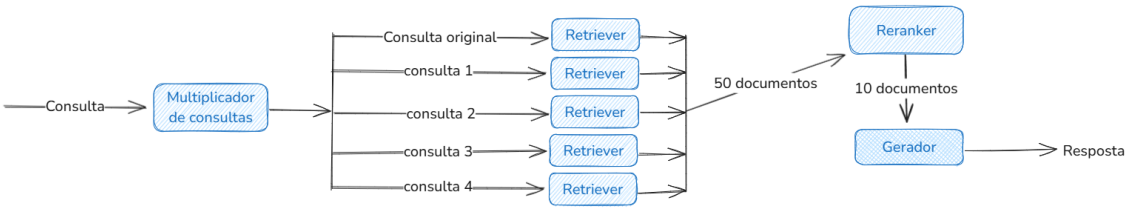

Figure 3. Fluxo do padrão RAG-Fusion

3. Metodologia

Neste estudo, trabalhamos com quatro componentes principais: padrões de arquiteturas RAG, tipos de consultas, um sintetizador de pares extraídas de documentos (consulta e resposta) e um avaliador. As arquiteturas RAG fornecem os padrões para recuperação e geração de respostas, enquanto os tipos de consultas estruturam as perguntas conforme suas características e complexidade, permitindo associar a adequabilidade de padrões aos tipos de consulta. O sintetizador organiza as informações em pares, facilitando a geração de métricas que avaliam a qualidade das respostas e do contexto recuperado. O avaliador utiliza essas métricas para mensurar a relevância e a precisão das respostas, orientando o aperfeiçoamento do sistema de Q&A no contexto jurídico.

3.1. Tipos de Consultas em Sistemas Q&A

Buscamos tipificar as principais consultas feitas a um sistema de Q&A para avaliar a qualidade das respostas com relação aos diferentes padrões. Uma consulta simples é uma pergunta direta que busca uma resposta objetiva e específica, sem necessidade de integração de múltiplas informações, tal como: ”Quem é o presidente do STF?”.

Uma consulta complexa exige uma resposta mais detalhada, que pode envolver a integração de informações de várias partes de um documento ou de diferentes documentos. O exemplo a seguir deve realizar uma busca à base de portarias do TCE para trazer a portaria utilizada na resposta: ”Considerando a Portaria nº 462/2017, quais os impactos da implementação do Sistema de Frequência online no TCE-GO?”. Uma consulta de comparação solicita uma análise entre duas ou mais entidades, conceitos ou abordagens, como no exemplo a seguir: ”Compare e contraste as portarias 1090/2017 e 81/2018 do Tribunal de Contas do Estado de Goiás”.

Uma consulta *multi-hop* envolve perguntas que requerem múltiplos passos de raciocínio ou a conexão de informações de diferentes fontes para construir uma resposta completa. O exemplo a seguir requer um raciocínio mais complexo: ”Se um servidor do TCE-GO foi admitido em 1º de abril de 2001 e, em 1º de setembro de 2010, completou

seu quarto decênio de efetivo serviço público, ele teria direito a receber a Medalha do Mérito Funcional Conselheiro Henrique Antônio Santillo em 2015?"

Uma consulta aberta representa uma questão ampla, permitindo uma resposta exploratória ou detalhada, sem necessariamente ter uma única resposta correta. Esse tipo de consulta incentiva o sistema RAG a gerar respostas que variam em conteúdo e profundidade, tal como no exemplo a seguir: "Como a implementação do processo eletrônico impactou a eficiência e a transparência do Tribunal de Contas do Estado de Goiás, considerando os desafios e as medidas adotadas para garantir a acessibilidade e a segurança das informações processuais?"

4. Experimentos

4.1. Códigos

Os códigos utilizados estão disponíveis para reprodução e consulta. A implementação dos padrões arquiteturais foi disponibilizada no repositório `JURIDIC-RAG-1F70` [1] e do método de avaliação no `evaluator-juridic` [2].

4.2. Bases de dados

Para a realização dos experimentos, utilizamos uma base de dados fornecida pelo Tribunal de Contas do Estado de Goiás (TCE-GO). A base é um conjunto de 3284 decisões do ano de 2024, que são interpretações e entendimentos sobre diversos casos julgados pelo tribunal, formando jurisprudências. Conselheiros podem se basear nessa base de decisões para julgar futuros casos análogos. A base completa pode ser acessada no site de Decisões do TCE-GO [3].

Utilizamos os métodos `RecursiveCharacterTextSplitter` do langchain com tamanhos de chunk = 1000 e overlap = 200. Para a vetorização dos documentos jurídicos, utilizamos o modelo `text-multilingual-embedding-002` do Google, para gerar *embeddings* das bases, permitindo a recuperação de informações com base em similaridade semântica. O modelo de linguagem `gemini-1.5-pro` [4] foi utilizado para geração das respostas no pipeline RAG.

Criamos uma base de dados com 66 perguntas sintéticas, sendo 11 específicas para cada tipo de consulta, que podem ser consultadas dentro do repositório `JURIDIC-RAG-1F70` no diretório `queries_responses_and_references`. Cada uma delas foi aplicada em cinco diferentes padrões arquiteturais RAG com o objetivo de avaliar a qualidade do contexto recuperado e das respostas fornecidas pelo *pipeline*. Devido a natureza não-determinística dos modelos de linguagem, cada execução foi repetida três vezes, e calculada a média aritmética das métricas de avaliação.

4.3. Avaliação

Para realizar uma análise das respostas geradas e dos contextos recuperados em cada pipeline utilizamos o RAGAS [Es et al. 2023] como *framework* de avaliação, utilizando

[1] https://anonymous.4open.science/r/JURIDIC-RAG-1F70

[2] https://anonymous.4open.science/r/evaluator-juridic

[3] https://decisoes.tce.go.gov.br/

[4] https://gemini.google.com/

o modelo `gemini-1.5-pro` para geração da avaliação automática. A avaliação contemplou as seguintes métricas: i) **fidelidade**: mede se a resposta está de acordo com o contexto recuperado, ii) **revocação de contexto**: avalia a proporção de documentos relevantes recuperados em relação ao total de documentos relevantes existentes, iii) **precisão de contexto**: mede a proporção de documentos relevantes entre todos os documentos recuperados, iv) **relevância de resposta**: avalia o quão bem a resposta atende à pergunta do usuário.

A avaliação das diferentes arquiteturas RAG revela variações significativas na qualidade das respostas no contexto jurídico, especialmente ao considerar métricas como fidelidade, revocação de contexto, precisão de contexto e relevância da resposta. Com base nesses parâmetros, observamos que as arquiteturas apresentam distintos desempenhos em relação à adaptação aos diferentes tipos de consulta.

A figura 4 apresenta o resultado da avaliação das arquiteturas em relação à métrica de fidelidade. A arquitetura Self-RAG apresentou os melhores resultados, principalmente em consultas simples, complexas e abertas. Este comportamento pode ser explicado porque essa arquitetura demonstra maior capacidade de manter as respostas dentro dos limites do contexto relevante. O gráfico demonstra a dificuldade de quase todos os padrões em manter fidelidade em consultas de comparação e consultas multi-etapas, com exceção do RAG-Fusion com multi-etapas, o que sugere a necessidade de um tratamento específico para esses tipos de consulta.

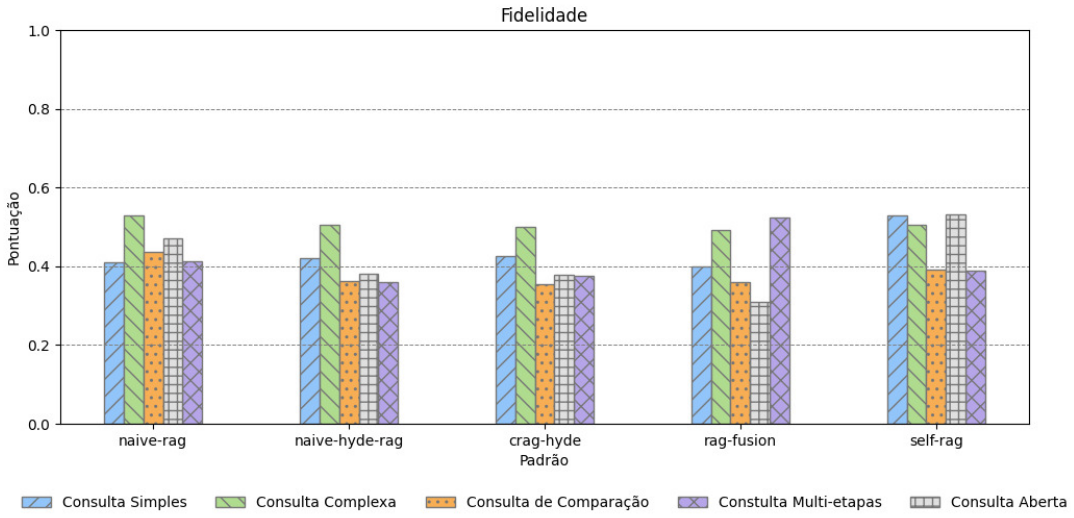

Figure 4. Fidelidade

A figura 5 apresenta os resultados da métrica de revocação de contexto. A arquitetura Self-RAG se destacou particularmente em consultas comparativas e abertas. Esse desempenho indica que esse padrão é eficaz em capturar as características mais relevantes dos documentos jurídicos. Arquiteturas como RAG-Fusion, no entanto, demonstram menores índices de revocação, especialmente em consultas complexas, o que limita sua eficácia em contextos que exigem ampla recuperação de informações.

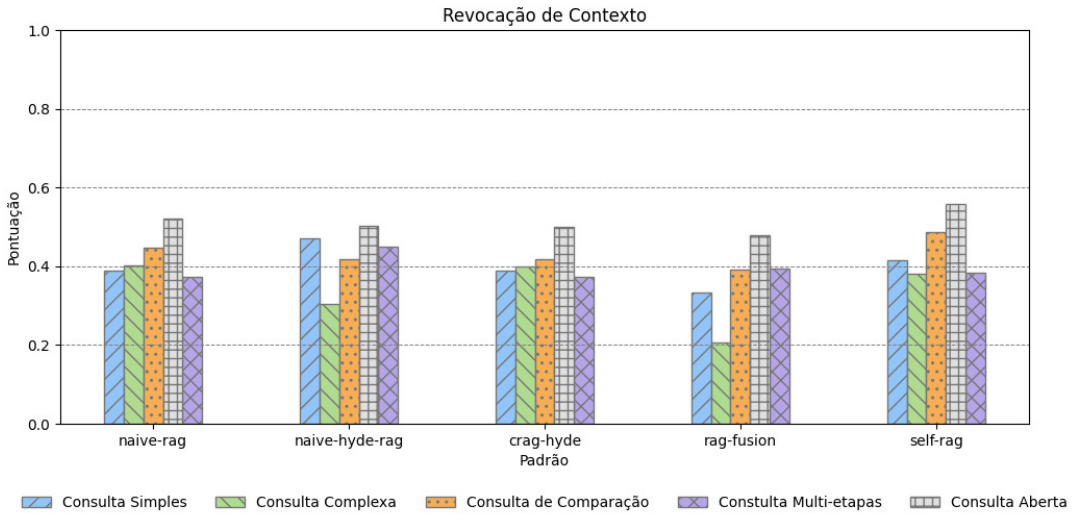

Figure 5. Revocação de contexto

Em relação à precisão de contexto, a figura 6 mostra que todos os padrões apresentaram baixo desempenho, principalmente para consultas complexas, de comparação e abertas. Os padrões conseguiram lidar melhor com consultas simples e multi-etapas, com destaque para Naive, Naive-HyDE e Self-RAG.

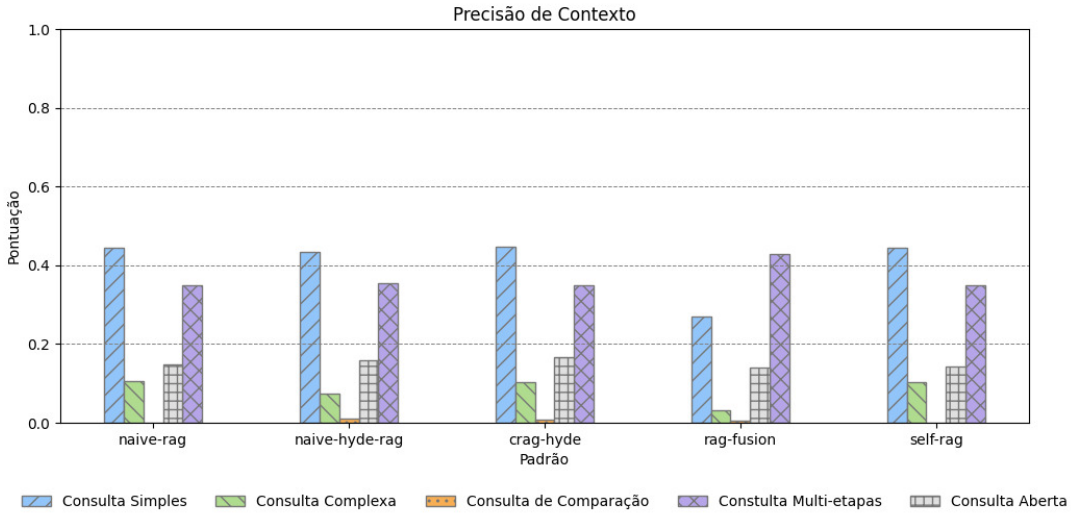

Figure 6. Precisão de contexto

Em relação à relevância da resposta, a figura 7 mostra que o Self-RAG se destacou por sua versatilidade, apresentando bom desempenho em todos os tipos de consulta, especialmente em contextos que exigem análise detalhada ou comparação. Por outro lado, o RAG-Fusion obteve o melhor desempenho em consultas multi-etapas.

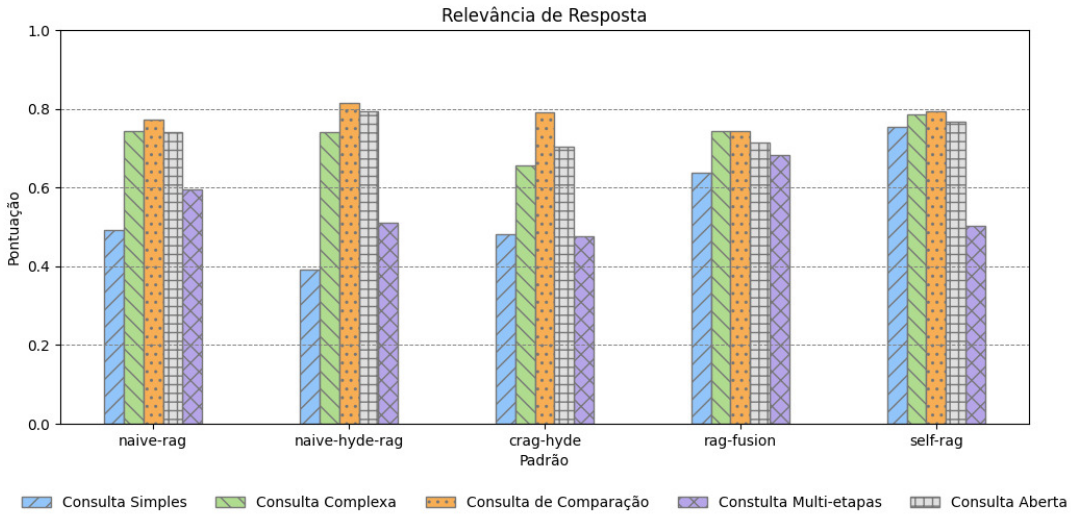

Figure 7. Relevância de resposta

A análise dos diferentes tipos de consulta revela que Self-RAG e Naive são mais indicadas para consultas diretas e complexas, com Self-RAG sendo também eficiente para consultas abertas. RAG-Fusion, por sua vez, apresenta bom desempenho em consultas multi-hop, nas quais é necessário estabelecer conexões entre múltiplos pontos de informação, porém com menor eficiência em fidelidade em consultas abertas. Esse perfil de desempenho diferencia as arquiteturas em termos de adequação para contextos jurídicos específicos, sugerindo que a escolha da arquitetura RAG ideal pode depender do tipo de consulta jurídica predominante.

Em síntese, a arquitetura Self-RAG se destaca pela robustez e consistência em fidelidade e relevância da resposta, adaptando-se bem a múltiplos tipos de consulta no contexto jurídico. Já Naive e Naive-HyDE se mostram alternativas eficientes em contextos específicos, como consultas diretas e comparativas. Por outro lado, RAG-Fusion e CRAG apresentam limitações em termos de fidelidade e revocação em certos tipos de consulta, especialmente aquelas que exigem maior precisão de contexto. Esses resultados sugerem que Self-RAG pode ser considerada a arquitetura mais versátil e confiável, enquanto arquiteturas como RAG-Fusion e CRAG podem demandar ajustes para contextos de consulta mais exigentes.

5. Conclusão

Neste estudo, identificamos cinco tipos principais de consultas entre usuários e sistemas de perguntas e respostas (Q&A) no contexto jurídico: consultas simples, consultas complexas, consultas de comparação, consultas multi-hop e consultas abertas. Cada tipo de consulta apresenta desafios específicos em termos de recuperação de informação e geração de respostas precisas e relevantes.

Analisamos o impacto dos diferentes padrões arquiteturais RAG,especificamente Naive, Naive-Hyde, CRAG, RAG-Fusion e Self-RAG, na qualidade das respostas e con-

textos recuperados no contexto jurídico. Observamos que a escolha do padrão RAG influencia significativamente a fidelidade, a revocação e precisão do contexto, além da relevância da resposta. Por exemplo, o padrão Self-RAG consistentemente produziu respostas com maior fidelidade e relevância, enquanto o RAG-Fusion destacou-se em termos de precisão do contexto recuperado.

O desempenho dos padrões RAG variou de acordo com os diferentes tipos de consulta. Para consultas simples e complexas, o Self-RAG apresentou desempenho superior em fidelidade e relevância da resposta. Em consultas *multi-hop*, o RAG-Fusion mostrou-se mais eficaz na recuperação de contextos precisos, essenciais para responder a perguntas que exigem encadeamento lógico de informações. Nas consultas abertas, tanto o Naive-Hyde quanto o Self-RAG tiveram os desempenhos mais notáveis, com o Naive-Hyde alcançando os maiores escores em revocação e precisão do contexto, e o Self-RAG com maior fidelidade e relevância nas respostas.

A avaliação demonstrou que os tipos de consulta no contexto jurídico impactam diretamente o desempenho das arquiteturas RAG. A escolha da arquitetura deve considerar não apenas a qualidade geral das respostas, mas também como cada arquitetura lida com os desafios específicos de cada tipo de consulta. Nossas descobertas sugerem que o Self-RAG é uma opção robusta para a maioria dos tipos de consultas, especialmente quando a fidelidade e a relevância são prioritárias. O RAG-Fusion é recomendado em cenários que exigem alta precisão no contexto recuperado, como em consultas *multi-hop*. Compreender essas dinâmicas é essencial para o desenvolvimento de sistemas de Q&A mais eficazes e confiáveis no domínio jurídico.

Dessa forma, é evidenciado que para profissionais da área jurídica, esses sistemas podem oferecer suporte valioso em tarefas críticas, como a tomada de decisões e a elaboração de documentos, desde que possuam arquiteturas robustas e bem ajustadas ao domínio. Para os desenvolvedores, a principal implicação é a necessidade de projetar pipelines que minimizem erros e inconsistências, utilizando verificações intermediárias e mecanismos de reexecução para assegurar que as respostas sejam fundamentadas na base jurídica específica, reforçando a relevância e a fidelidade dos resultados.

5.1. Trabalhos Futuros

Para otimizar sistemas de consulta, em trabalhos futuros, pretende-se implementar um mecanismo de roteamento que direcione consultas para pipelines específicas, utilizando um classificador para alocar a melhor arquitetura RAG. Além disso, novos paradigmas como Cadeia de Pensamento [Wei et al. 2022] e Self-Discovery [Zhou et al. 2024] serão aplicados para aprimorar o raciocínio e a adaptação das respostas, após identificar tarefas específicas dos profissionais jurídicos. No campo da avaliação de arquiteturas RAG, destaca-se a necessidade de sistemas como o ARES (Automated Retrieval Evaluation System), que realiza ajuste fino com perguntas validadas, permitindo comparações mais precisas e métricas alinhadas à satisfação do usuário final [Saad-Falcon et al. 2023].

Observa-se a necessidade de aprimorar o sintetizador para garantir a relevância dos pares de perguntas e respostas, já que a sintetização pode não refletir com precisão as consultas reais. Uma solução envolve a coleta de perguntas e respostas genuínas, validadas quanto à relevância, para ajuste fino de modelos ou uso de few-shot prompting. Planeja-se também expandir os estudos futuros, aumentando de 11 para cerca de

100 perguntas por tipo, o que não foi possível devido a restrições financeiras com APIs. Para superar isso, pretende-se realizar experimentos utilizando máquinas locais de uma instituição jurídica, aumentando a confiabilidade dos resultados.

References

Asai, A., Wu, Z., Wang, Y., Sil, A., and Hajishirzi, H. (2023). Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv*.

Es, S., James, J., Espinosa-Anke, L., and Schockaert, S. (2023). Ragas: Automated evaluation of retrieval augmented generation. *arXiv*.

Fan, Wenqi, Ding, Yujuan, Ning, Liangbo, Wang, Shijie, Li, Hengyun, Yin, Dawei, Chua, Tat-Seng, Li, and Qing (2024). A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.

Gao, L., Ma, X., Lin, J., and Callan, J. (2022). Hprecise zero-shot dense retrieval without relevance labels. *arXiv*.

Jagerman, R., Zhuang, H., Qin, Z., Wang, X., and Bendersky, M. (2023). Query expansion by prompting large language models. *arXiv*.

Krasadakis, Panteleimon, Sakkopoulos, Evangelos, Verykios, and S, V. (2024). A survey on challenges and advances in natural language processing with a focus on legal informatics and low-resource languages. *Electronics*, 13(3):648.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv*.

Ma, X., Gong, Y., He, P., Zhao, H., and Duan, N. (2023). Query rewriting for retrieval-augmented large language models. *arXiv*.

Rackauckas, Z. (2024). Rag-fusion: a new take on retrieval-augmented generation. *arXiv*.

Saad-Falcon, Jon, Khattab, Omar, Potts, Christopher, Zaharia, and Matei (2023). ARES: An automated evaluation framework for retrieval-augmented generation systems. *arXiv preprint arXiv:2311.09476*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *arXiv*.

Wiratunga, Nirmalie, Abeyratne, Ramitha, Jayawardena, Lasal, Martin, Kyle, Massie, Stewart, Nkisi-Orji, Ikechukwu, Weerasinghe, Ruvan, Liret, Anne, Fleisch, and Bruno (2024). Cbr-rag: Case-based reasoning for retrieval augmented generation in llms for legal question answering. In *International Conference on Case-Based Reasoning*, pages 445–460. Springer.

Yan, S.-Q., Gu, J.-C., Zhu, Y., and Ling, Z.-H. (2024). Corrective retrieval augmented generation. *arXiv*.

Zhou, P., Pujara, J., Ren, X., Chen, X., Cheng, H.-T., Le, Q. V., Chi, E. H., Zhou, D., Mishra, S., and Zheng, H. S. (2024). Self-discover: Large language models self-compose reasoning structures. *arXiv preprint arXiv:2402.03620*.