Justice in the Loop: Designing Responsible AI for Legal Research Support

Finn Alberts and Leonardus Pustjens
Responsible Artificial Intelligence

June 9, 2026

1 Introduction

The use of artificial intelligence (AI) in the courtroom has been discussed for many years, with a strong focus on the risks involved [1; 2]. Traditional Explainable AI (XAI) issues, such as fairness, responsibility, transparency, interactivity, interpretability, explainability, robustness, stability & satisfaction [3], are especially relevant for the legal domain and cause societal and legal difficulties [1; 2; 3]. Due to the black-box nature of most AI systems, it is often discouraged to use AI for fully automated decision-making, as the lack of interpretability and explainability raises questions about the legitimacy of decisions made by such a system and the acceptance of these decisions by society [1; 2; 4]. Additionally, current open issues such as hidden biases in large datasets and opaque model behavior can cause discriminatory results, which are unacceptable in the legal domain [1; 5].

However, AI may still support judges and lawyers in the legal domain as an assistive technology rather than replacing human decision-making entirely [1; 2; 5]. In this setting, the final responsibility for legal decisions remains with the human judge or lawyer, helping to mitigate risks associated with fully automated decision-making, such as bias and lack of accountability. At the same time, AI systems have the potential to improve the efficiency of legal processes through the partial automation of tasks such as legal research and document analysis [1].

A commonly used AI framework for information retrieval and analysis of natural language is the Retrieval-Augmented Generation (RAG) framework [6]. The core architecture consists of a retriever and a generator. The retriever retrieves relevant documents from a database (often a vector database) given a query. The retrieved documents are then fed into a generator (often a pre-trained Large Language Model (LLM)), together with the original query, to generate an answer [6]. The base idea is that answers are more grounded in external knowledge, which can be manually verified to ensure relevance.

Although a RAG-based system appears to be a solution for automating document retrieval and legal research, these systems should still be approached with caution, as they are not yet fully free of the aforementioned AI issues [3; 7]. In this paper, we will therefore propose a responsible RAG design for the legal domain, which aids more transparent and fair document retrieval and analysis, ensuring fairer decision-making.

In the design of this system, we take into account the perspectives of four different stakeholders:

1. **Defendants and other legal parties**, as incorrect or biased decisions may have major legal, financial, and personal consequences for the individuals involved.

2. **Judges**, as they aim to balance opposing arguments as fairly and carefully as possible within the boundaries of the law.

3. **Lawyers**, as they may either interact with these systems through their use by courts or use RAG-based systems themselves during legal research and case preparation. In both cases, such systems may positively or negatively influence their legal position.

4. **Judicial support staff**, as the automation of certain tasks may significantly change their day-to-day responsibilities and potentially reduce the amount of staff required.

The rest of this paper is structured as follows: Section 2 discusses the ethical issues of using a RAG-based system for document retrieval, Section 3 proposes a responsible design solution, and Section 4 concludes our work.

2 Identification of the Ethical Issue

To systematically evaluate the ethical implications of Retrieval-Augmented Generation (RAG) in litigation, it is essential to first establish what constitutes a trustworthy system. A comprehensive framework for a trustworthy RAG architecture typically encompasses six key dimensions [7]: *Factuality* (the accuracy and truthfulness of generated information), *Robustness* (reliability in resisting errors and external threats), *Fairness* (minimizing bias and ensuring equitable treatment), *Transparency* (making system processes and decisions clear and understandable), *Accountability* (mechanisms to hold the system responsible for its outputs), and *Privacy* (protection of personal data). While these dimensions serve as a necessary baseline, their application in the high-stakes legal domain introduces complex, and often conflicting, normative trade-offs.

Evaluating these dimensions requires us to look beyond abstract legal philosophy; we must analyze how code architecture forces specific compromises [5]. For instance, current legal tech is largely characterized by proprietary "walled gardens" managed by incumbents such as LexisNexis. While these systems attempt to maximize *Factuality* and *Robustness* by restricting the retriever to validated corpuses, this boundary inherently compromises *Transparency* and *Fairness* [7; 8]. Restricting data extraction to historical archives often embeds systemic societal biases directly into the vector database, making neutral retrieval nearly impossible [9; 8].

A fundamental technical hurdle is that *Fairness* cannot be trivially optimized. Engineers often attempt to mitigate bias using constraints such as demographic parity or equalized odds [3]. However, these formal definitions are frequently mutually exclusive; a system optimized for one inherently degrades the other [10]. Consequently, the selection of an embedding model, the calibration of a distance metric, or the tuning of a re-ranking algorithm is never a purely objective engineering choice. It represents an ethical stance disguised as system

configuration, effectively shifting the authority to define algorithmic justice from the judiciary to the software architect.

Additionally, another critical point of attention is that one must ensure that when the model's internal knowledge (knowledge learned by the LLM based on patterns on training data) and external knowledge retrieved by the retriever, the external knowledge is prioritized [7]. This is especially relevant for the legal domain, where answers based on the internal knowledge are less transparent and more vulnerable to biases and may directly influence legal rulings.

This bias scales poorly when pipelines cross geopolitical borders. Within Western legal traditions, the semantic understanding of justice is far from monolithic. For instance, while the German framework prioritizes a strict statutory hierarchy (*Rechtsstaat*), the Dutch system relies on a pragmatic, consensus-driven approach governed by equity (*redelijkheid en billijkheid*). A RAG pipeline operating over uncurated cross-border datasets risks flattening these jurisdictional nuances [7]. If a retriever surfaces context based on pure semantic similarity without strict regional filtering, it may present arguments that are linguistically sound but procedurally invalid for a local court [4].

Furthermore, if both parties deploy these systems, litigation risks devolving into a technological arms race. When the quality of representation becomes tied to compute power -specifically the ability to afford advanced re-ranking models or superior prompt engineering -the foundational principle of *equality of arms* is threatened [1; 2].

On a macro-ethical level, this systemic reliance risks the institutional ossification of law. Legal systems evolve through human advocacy that challenges established precedents. If RAG tools consistently flag certain arguments as historically unsuccessful, lawyers may cease to present them [1]. This creates an algorithmic feedback loop where the past dictates the future, effectively freezing the evolutionary capacity of common and civil law [2].

This issue is compounded by the "lazy jurist" problem. When an AI tool delivers accurate summaries 99% of the time, users develop automation bias - a psychological tendency to trust automated suggestions while reducing active verification [5]. Over time, legal professionals may delegate their moral and analytical responsibilities to the software architecture, allowing the system to act as an uncredited decision-maker [1].

3 Responsible Solution Design

3.1 Integration of Value-Sensitive Design

Addressing the ethical concerns outlined in Section 2 requires a clear design approach. To achieve this, we base our architecture on the Value-Sensitive Design (VSD) framework. Instead of only evaluating ethics after a system is built, VSD actively builds human values into the technology through conceptual, empirical, and technical steps.

For our legal RAG system, this means that core values like *fairness*, *equality of arms*, and *accountability* directly shape the system. We take into account how legal professionals work in practice, especially their risk of automation bias and the "lazy jurist" effect. To counter this, technical safeguards are needed. By purposefully adding friction to the *Query verifier* and *Case verifier*, and by using a transparent, non-trainable retriever (see Section 3.4), we

turn these abstract values into concrete system features. Ultimately, this approach ensures the system prioritizes legal accountability over simple efficiency.

3.2 Overall architecture

The overall architecture of our proposed design is depicted in Figure 1. We adapt the standard RAG architecture to allow for a more transparent and interpretable system, while including human-in-the-loop components to improve fairness and accuracy, therefore limiting biases.

In terms of functionality, we deliberately limit the system's generation component to summarization only, to mitigate the risk of internal and external knowledge conflicts [7] as laid out in Section 2. Although we acknowledge that this makes the system less efficient for the end-user, it is important to prevent the system from further analyzing the documents beyond simply summarization, as this could result in users requesting the system to make legal decisions, which are to be prevented to ensure fair decisions and legitimacy.

3.3 Legal case database

The foundation of the RAG system is its vector database, which must be carefully curated to ensure both relevance and jurisdictionl alignment. In the context of the Netherlands, the database leverages autoritative, open-access legal data sources. Primary sources include the open data API of *Rechtspraak.nl*, which provides anonymized judicial decisions in XML format, and *Wetten.nl* for consolidated legislation. Furthermore, the Linked Data Overheid (LiDO) dataset[1] is integrated to map relationships and citations between different cases. This network of citations allows the retriever to group documents by case, ensuring that related appellate decisions or supporting records are preserved, which directly adresses the context-loss problem identified earlier. By strictly limiting the database to these national sources, we actively prevent foreign jurispredence from entering the vector space. This avoids the cross-border flattening of legal principles, ensuring that fundamentally different legal paradigms do not contaminate the retrieval process.

To process these Dutch legal texts responsibly, the embedding model must be tailored strictly to the Dutch language and legal terminology. Relying on generic, global large language models introduces a significant risk of semantic contamination, as these models are pre-trained on a massive mixture of international laws. Using them could inadvertently blend foreign legal reasoning into the Dutch context. We therefore opt to use GPT-NL as our embedding model (which we will also use as our LLM for summarization, see Section 3.5). Although not yet released at the time of writing, it is currently one of the few language models claiming demonstrable GDPR compliance through transparent data governance and training procedures. In addition, the model is developed with a specific focus on Dutch norms and values [11]. This localized approach ensures the semantic nuances of the Dutch legal system, such as the principle of *redelijkheid en billijkheid* are better preserved while maintaining transparency in how the database processes and retrieves information.

[1] Available at: `https://linkeddata.overheid.nl/`

3.4 Ensuring retrieval of relevant documents

In a traditional RAG architecture, the retriever is typically trained to take a user query as input and rank document chunks based on their estimated relevance [6]. We argue that this approach does not fully satisfy the requirements of systems operating in the legal domain. In particular, four key challenges need to be addressed:

- **Retriever bias:** As the retriever is trained, it might include biases which should be prevented.
- **Lack of context:** Retrieving only isolated document chunks may omit important contextual information, while context is often essential for understanding legal reasoning and case outcomes.
- **Missing relevant documents:** Relevant legal documents may not be retrieved. This includes not only entirely missed cases, but also related documents such as appeal decisions or subsequent rulings associated with the original case.
- **Including non-relevant documents:** Irrelevant documents may be retrieved, potentially introducing misleading information that could affect the decision-making process.

To reduce the risk of introducing additional bias within the retrieval process, we employ a non-trainable retriever rather than a learned retrieval model. Retrieval is instead performed using embedding-based similarity scoring through a transparent distance metric. While this approach does not eliminate bias, as the embeddings themselves may still encode historical or societal biases, it makes the retrieval process more inspectable and easier to audit. Although this design choice may lead to lower retrieval performance compared to more complex learned retrieval approaches, we prioritize transparency and traceability over marginal performance gains, particularly in legal settings where accountability and trust are critical.

To mitigate the lack of context, we modify how relevant documents are retrieved. While documents are still split into chunks when populating the database, isolated chunks are not returned directly. Instead, chunks are used only to identify relevant documents, after which the complete source document of the selected chunk is returned. This approach ensures that the broader context surrounding the retrieved information is preserved.

Furthermore, to ensure that related legal documents are also included, the retrieval process actively leverages the LiDO citation network. While traditional RAG systems often struggle with retrieving documents that are procedurally linked but semantically distinct, our retriever uses LiDO's relational metadata to group documents by case. Once an initial text chunk is retrieved based on vector similarity, the system automatically pulls in the entire network of connected documents - such as appeal decisions, supporting evidence, or related statutory articles - linked via their LiDO URIs. This ensures that documents which might not match the user's initial query, but are crucial for a complete legal understanding, are still retrieved, providing vital contextual and supplementary information.

Additionally, relevant legal cases may not always be retrieved using only the original user query. Therefore, the user query is first passed to the *Query generator*. This component, implemented using a small-scale LLM, generates semantically similar queries to increase the likelihood of retrieving relevant documents. As LLM-generated queries may introduce bias

or semantic drift, the generated queries are first processed by the *Query verifier* before being forwarded to the *Retriever*.

The *Query verifier* serves as a safety mechanism in which the user validates the relevance of the generated queries. To encourage reflective decision-making, we apply principles of friction by design aimed at promoting Type 2 thinking [5]. Rather than immediately allowing interaction, generated queries are presented to the user individually, with a five-second delay before approval or dismissal options become available. This deliberate pause is intended to encourage users to actively evaluate the relevance of each generated query instead of relying on rapid, intuitive decision-making.

Lastly, to reduce the risk of including non-relevant documents, all retrieved cases are processed by the *Case verifier*. Similar to the *Query verifier*, principles of friction by design are applied by presenting retrieved legal cases to the user individually, with a built-in delay before approval or dismissal becomes available. The duration of this delay is determined by the size of the case, measured by word count. The *Case Verifier* does not present all case documents to the user, but only the final ruling (including appeal rulings if applicable), as this document often contains a summary of the case and the judge's final decision. The goal here is not for the user to fully analyze the legal case, but give them enough time and context for an initial assessment of its relevance instead. After a case has been verified, all other documents related to the case are included as well for the next step.

3.5 Summarization

In the *Summarizer*, summaries of the retrieved documents and cases are generated using a locally hosted LLM. Although LLMs may introduce biases or misrepresent information, we aim to reduce this risk by restricting the model's role to summarization rather than using it for analytical tasks or decision-making. Summaries are generated at the document level to reduce the risk of omitting potentially relevant details and preserve important contextual information.

We again opt to use GPT-NL, with the same arguments as before regarding its Dutch focus and GDPR compliance. In addition, GPT-NL considers summarization as one of its primary use cases, which makes it a good fit for our architecture [11].

4 Discussion and Conclusion

In this paper, we introduced a responsible architecture for an AI-driven document retrieval and legal research system based on a RAG architecture. The proposed architecture aims to reduce risks associated with AI systems by incorporating human-in-the-loop components and minimizing reliance on trainable components where possible.

Although the proposed architecture improves upon standard RAG pipelines from a responsibility perspective, it still has several limitations and risks that require future research. First, not all relevant legal documents are publicly available, which may limit the depth and completeness of the legal research process. As a result, important background context may be missing or relevant non-public cases may not be considered.

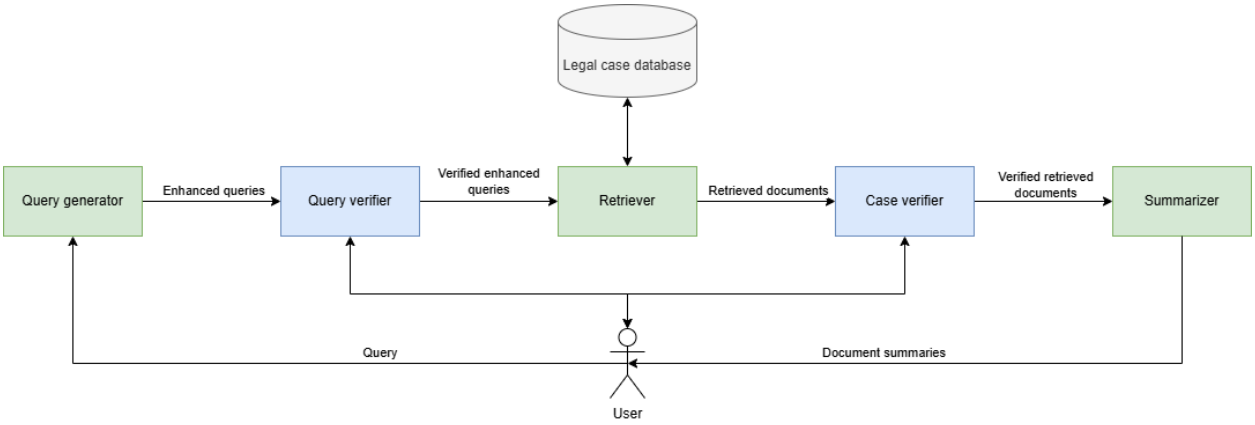

Figure 1: Overall architecture of the proposed solution. After each stage in the retrieval and generation process (green components), the output is presented to the user for validation through the verifier components (blue). The RAG architecture is simplified by using a lightweight, non-trained retriever and restricting the generator to summarization tasks only, while retrieval performance is enhanced through the automatic generation of additional semantically similar queries.

Furthermore, relevant information may still be overlooked, as this work focuses primarily on embedding similarity for retrieval without explicitly addressing more complex data modalities such as images, videos, audio recordings, or tables. Such information may be more difficult to retrieve using embedding similarity alone, requiring additional research into multimodal retrieval approaches.

Third, although measures were introduced to reduce the risk of bias within the architecture, several components may still be affected by biases present in training data. Specifically, the *Query generator*, the *Retriever*, and the *Summarizer* all rely on models trained on real-world data. While the *Query generator* and *Retriever* are supplemented with human-in-the-loop mechanisms and the *Retriever* and *Summarizer* use GPT-NL, the extent to which these measures reduce bias remains subject to empirical evaluation.

A major limitation of this work is that the proposed architecture has not yet been implemented or evaluated in practice. Consequently, the effectiveness of the proposed safeguards and the practical impact of the identified limitations remain uncertain. Future work should therefore focus on implementing and evaluating the architecture in controlled settings, with specific attention to the limitations discussed above.

Aside from technical limitations, there are also opportunities for extending the proposed architecture. In this work, we specifically focused on the Dutch legal context, as reflected in both the documents included in the *Legal case database* and the use of GPT-NL. Extending the architecture to other legal systems may reveal region-specific challenges and opportunities, particularly regarding legal frameworks, language, and data availability.

Furthermore, we identify opportunities to improve the *Case verifier* through the integration of knowledge graphs that explicitly visualize relationships between legal documents and cases. Such an approach could improve the user's ability to assess the relevance and context of retrieved cases. However, exploring these possibilities is considered outside the scope of this current work.

Overall, this work provides a first step toward more responsible AI-supported legal research by explicitly incorporating transparency, human oversight, and bias mitigation into the architecture design. While substantial work remains regarding implementation and evaluation, we believe that combining responsible AI principles with retrieval-based systems provides a promising direction for developing trustworthy AI systems in high-stakes domains such as the legal sector.

References

[1] R. W. Campbell, "Artificial intelligence in the courtroom: The delivery of justice in the age of machine learning," *Colorado Technology Law Journal*, vol. 18, no. 2, pp. 323–343, 2020. [Online]. Available: https://scholar.law.colorado.edu/ctlj/vol18/iss2/2

[2] T. Sourdin, "Judge v Robot? Artificial Intelligence and Judicial Decision-Making," *University of New South Wales Law Journal*, vol. 41, 11 2018.

[3] S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J. M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez, and F. Herrera, "Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence," *Information Fusion*, vol. 99, p. 101805, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253523001148

[4] G. Ras, N. Xie, M. van Gerven, and D. Doran, "Explainable deep learning: A field guide for the uninitiated," 2021. [Online]. Available: https://arxiv.org/abs/2004.14545

[5] J. R. Schoenherr, R. Abbas, K. Michael, P. Rivas, and T. D. Anderson, "Designing ai using a human-centered approach: Explainability and accuracy toward trustworthiness," *IEEE Transactions on Technology and Society*, vol. 4, no. 1, pp. 9–23, 2023.

[6] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," 2021. [Online]. Available: https://arxiv.org/abs/2005.11401

[7] Y. Zhou, Y. Liu, X. Li, J. Jin, H. Qian, Z. Liu, C. Li, Z. Dou, T.-Y. Ho, and P. S. Yu, "Trustworthiness in retrieval-augmented generation systems: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2409.10102

[8] S. Dai *et al.*, "No free lunch: Retrieval-augmented generation undermines fairness in llms, even for vigilant users," *arXiv preprint arXiv:2409.10012*.

[9] European Union Agency for Fundamental Rights, "Bias in algorithms: Artificial intelligence and discrimination," Publications Office of the European Union, Tech. Rep., 2022. [Online]. Available: https://fra.europa.eu/sites/default/files/fra_uploads/fra-2022-bias-in-algorithms_en.pdf

[10] F. Doshi velez and B. Kim, *Considerations for Evaluation and Generalization in Interpretable Machine Learning*, 11 2018, pp. 3–17.

[11] K. Krikhaar. (2026, Feb.) Nederlandse GPT-NL is klaar voor gebruik: 'Voldoet als enige taalmodel aan AVG'. [Online]. Available: https://tweakers.net/reviews/14412/nederlandse-gpt-nl-is-klaar-voor-gebruik-voldoet-als-enige-taalmodel-aan-avg.html