A Multi-Agent RAG Framework for Regulatory Compliance Checking of Software Requirements

SOUVICK DAS, University of Luxembourg, Luxembourg
NOVARUN DEB, University of Calgary, Canada
NABENDU CHAKI, Department of Computer Science and Engineering, University of Calcutta, India
AGOSTINO CORTESI, DAIS Department, Ca' Foscari University, Italy

Ensuring compliance with regulations poses considerable challenges for software development, particularly during the requirements specification phase. Traditional methods rely heavily on manual inspections that are time-consuming, and prone to errors. This research proposes an innovative framework that leverages the synergy of multiple AI agents to automate software requirement compliance verification partially. The framework integrates Large Language Models (LLMs), prompt engineering, and Retrieval-Augmented Generation (RAG) to analyze, detect, and revise non-compliant requirements. The core of our proposal lies in multi-agent communication, where distinct AI agents collaborate to achieve the overarching goal of compliance checking. LLMs comprehend requirements specifications, while prompt engineering guides LLMs towards compliance-related aspects. The RAG techniques detect non-compliant requirements and suggest changes. Finally, a robust Human-in-the-Loop mechanism ensures accuracy, reliability, and adaptability. A tool, available online, is implemented to translate the technology for effective application. We discuss its ability to identify non-compliant requirements in an extensive experimental evaluation.

Additional Key Words and Phrases: Requirements Engineering, Large Language Models, Compliance Checking, RAG

1 Introduction

In today's data-driven world, ensuring software compliance with regulations like the General Data Protection Regulation (GDPR) or the Data Protection Act is paramount. However, such adherence presents significant challenges for software development, especially during the critical requirements specification phase [3]. Traditional compliance check methods, highly reliant on manual inspection, are labor-intensive, time-consuming, and prone to errors. This is further complicated by the inherent complexity of legal texts — characterized by dense language, extensive cross references, conditional exceptions, and potential ambiguities — and the informal language often used in Software Requirements Specification (SRS) documents. The challenge is to align technical specifications with evolving high-level legal regulations [21, 32]. To illustrate the subtlety of this challenge, let us consider a hypothetical requirement from the SRS document of an e-commerce platform: "*User data will be stored in a secure database using industry-standard encryption practices.*" While addressing data security at a surface level, this statement leaves several aspects of GDPR compliance unclear. It does not specify data retention periods, potentially violating the data minimization principle (Article 5). Additionally, no mechanism is mentioned for

Part of the work has been done by the first author while working as a Research Associate at Ca' Foscari University, Venice, Italy.

Authors' Contact Information: Souvick Das, souvick.das@uni.lu, University of Luxembourg , Luxembourg, Luxembourg; Novarun Deb, University of Calgary, Calgary, Canada, novarun.deb@ucalgary.ca; Nabendu Chaki, Department of Computer Science and Engineering, University of Calcutta, Kolkata, India; Agostino Cortesi, DAIS Department, Ca' Foscari University, Venice, Italy.

ACM 1557-7392/2025/12-ART
https://doi.org/10.1145/3785472

users to access, rectify, or erase their data — rights guaranteed under GDPR (Articles 15-17). Identifying such subtle gaps manually requires significant effort and legal expertise.

Although the advent of LLMs offers unprecedented potential for automating this task, applying them as standalone tools for such a critical domain presents its own significant challenges that can undermine reliability and accuracy.

- *Knowledge Cutoff and Evolving Regulations:* Knowledge of the LLM is fixed at the time of its last training. Regulatory landscapes are dynamic, with new laws, amendments, and judicial interpretations continuously emerging. A standalone LLM would be unaware of such shifts, leading to assessments based on outdated legal understanding.
- *"Hallucinations" and Factual Inaccuracy:* LLMs are generative models, not truth-finding machines. They can "hallucinate"—inventing legal clauses, misremembering article numbers, or misstating obligations with complete confidence. In regulatory compliance, where precision is paramount, such inaccuracies can lead to severe non-compliance.
- *Insufficient Domain-Specific Nuance:* While trained on vast data, a standalone LLM's understanding of highly specific legal terminology and the subtle, implicit relationships within complex regulatory frameworks can be superficial. It may not grasp the precise legal definition of "consent" under GDPR versus its interpretation in a specific national law, a distinction vital for correct assessment.
- *Lack of Transparency and Explainability:* When a monolithic LLM produces an incorrect assessment, its reasoning process is largely opaque. The primary recourse is often limited to re-prompting, without a clear path to diagnose the root cause of the error.

This study proposes an innovative framework designed to alleviate these intrinsic constraints and responsibly leverage the potential of LLMs. It tackles regulatory compliance issues by integrating four synergistic components: (1) Retrieval-Augmented Generation (RAG) dynamically grounds analysis in authoritative regulatory texts, ensuring current and contextually relevant information; (2) a multi-agent architecture employs specialized AI modules that independently evaluate requirements then reconcile findings through consensus-building; (3) Human-in-the-Loop (HITL) integration enables expert validation of ambiguous cases while continuously improving system performance through feedback; and (4) explainable AI mechanisms provide transparent decision pathways by generating audit-ready rationales and maintaining bidirectional traces between requirements and governing regulations. This integrated approach is validated through rigorous quantitative metrics and qualitative expert assessments across diverse compliance scenarios.

The effectiveness of the system is thoroughly evaluated using a dual approach that involves quantitative benchmarking against standard compliance metrics and qualitative insights provided by domain experts from various real-world projects. This involves a quantitative review of the RAG pipeline employing specialized metrics, alongside qualitative case studies from different projects. To evaluate the practical relevance and clarity of the system's outcomes, we collect feedback from domain experts and student crowdworkers equipped with training in requirements engineering. This comprehensive evaluation underscores the capability of the framework to accurately identify non-compliant requirements.

In this context, we have identified the following key research questions, explored in the different sections of our article. Section 7 presents a detailed discussion of these.

RQ1: *Can we use LLMs to analyze software requirements and check their compliance with regulations while considering the specific context in which the software will operate?* - This research question investigates how well LLMs interpret and recognize the association of regulatory policies with software requirements for specific contextual information about the software's intended purpose.

RQ2: *How can we design a transparent and reliable framework for LLM-based compliance checking, addressing the complexities of legal interpretation and the "black-box" nature of LLM?* - The research question investigates how

to design a framework that combines multiple AI perspectives and reasoning to make the AI's decision process both understandable to humans (transparent) and provide trustworthy results (reliable).
RQ3: *How can we effectively evaluate the efficiency of an LLM-based compliance checker using appropriate metrics?* - Considering the limitations of traditional metrics for complex tasks like requirements modification and requirements generation, we investigate how to assess the performance of RAG in terms of quantitative metrics. It also aims for the qualitative assessment and practicality of the system's overall outcomes.
RQ4: *Can the proposed solution be adapted to different regulatory landscapes and accommodate evolving compliance standards, demonstrating its scalability and adaptability?* - This research question investigates the potential of the proposed solution to be generalized and adapted to different regulatory landscapes and accommodate evolving compliance standards over time.

While existing solutions alternate between inflexible template-based systems [4] and unreliable monolithic LLM approaches [9], our framework pioneers a third way: blending dynamic regulatory retrieval, multi-agent consensus validation, and fully traceable decision rationales. This hybrid approach ensures precise cross-reference resolution while maintaining audit-ready compliance pathways through explainable AI components. The main contributions of this paper are as follows.

- *Multi-Agent Framework:* We present a novel framework that utilizes RAG, Prompt Engineering, and multiple AI agents collaborating to analyze software requirements and identify potential areas of noncompliance with specified regulations.
- *Multifaceted Evaluation:* We employ a comprehensive evaluation approach, combining quantitative metrics for retrieval effectiveness and qualitative crowdsourcing feedback to assess the framework's performance and user satisfaction.
- *Case Studies and Analysis:* We showcase the framework's applicability through case studies on diverse software projects, demonstrating its ability to identify non-compliant requirements.

All the experiments discussed in the article can be replicated with our tool, available online[1]. Users can apply the compliance analysis on SRS documents and policies of their preference.

The rest of the paper is organized as follows. Section 2 documents the related works existing in the literature. Section 3 delves into the details of the proposed framework architecture and its constituent modules. Section 4 outlines the details of the implementation of each module and the technologies used. Section 5 details the explainable AI mechanisms that are integrated with the framework for enhancing transparency of the system. In Section 6, we present the experimental evaluation of the framework, including the assessment of the RAG pipeline and the results of the crowd-sourcing. Section 7 provides a discussion of the research questions and the insights gained from the evaluation. Section 8 identifies potential validity threats, and finally Section 9 concludes the article with key findings and directions for future research.

2 Related Works

The field of regulatory compliance has gained significant attention from the Requirements Engineering (RE) community.

Early research on elicitation and representation of compliance requirements focused on extracting and representing compliance requirements from regulatory texts. Breaux et al. [11] pioneered the extraction of rights and obligations from legal documents. This approach was extended by Kiyavitskaya et al. [21] with a tool to analyze policy documents based on rights and obligations. Zeni et al. [36] and Sleimi et al. [32] advanced this area by extracting semantic metadata from legal texts. Urban et al. [35] examined the behavior of online advertising companies under GDPR. Alhazmi et al. [1] investigated the challenges in implementing GDPR principles and

[1]https://github.com/SutraMind/ComplAI-Agents

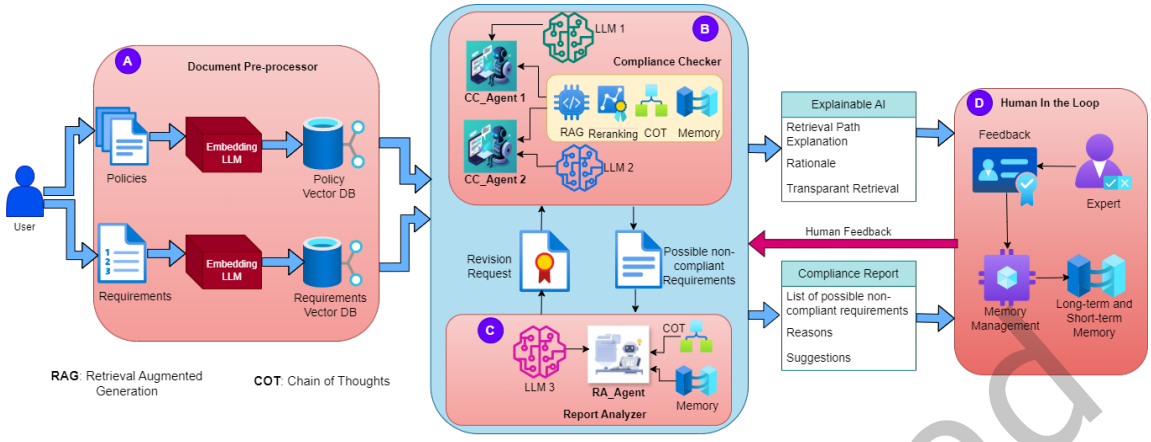


Fig. 1. Multi-Agent Compliance Checking Framework

proposed a game design framework to understand privacy-related requirements. Shastri et al. [31] identified practices in cloud-scale systems that can hinder GDPR compliance.

In terms of automated support for compliance checking, the existing research focuses on using traditional machine learning algorithms and NLP techniques to analyze privacy policies for GDPR compliance. Studies such as [3, 16] exemplify this approach. Amaral et al. [3] combine NLP and feature-based learning to extract metadata from privacy policies and automate GDPR compliance checking. Additionally, works like [10, 27] explore the use of semantic web technologies for compliance checking.

Data Processing Agreements (DPAs) have become a distinct type of regulatory document with considerable impact on software development. A study by Amaral [2] tackles the issue of ensuring the completeness of the DPA in relation to GDPR requirements. Another study by Amaral et al. [4] examines various methods, such as model-driven engineering, machine learning, and NLP with semantic frames, to evaluate DPA compliance. Additionally, Ilyas et al. [7] investigated the use of AI techniques, including traditional ML classifiers, BiLSTM, and BERT, to verify DPA compliance.

Recent progress in AI, especially with the advent of LLMs, has created new opportunities to automate legal compliance. Research such as [9, 17] has investigated the capabilities of LLMs for compliance verification and inconsistency identification. Berger et al. [9] propose integrating retrieval with LLM-based zero-shot learning to automate compliance verification for audit decisions, while Fantechi et al. [17] introduce an LLM-based approach for identifying inconsistencies in requirements specifications.

Although prior research has explored LLM-based methods, they frequently encounter difficulties in addressing the intrinsic challenges of using LLMs, such as the ”lost-in-the-middle” issue, maintaining long context, and hallucinations, while traditional ML and NLP approaches struggle with semantic comprehension. Our framework overcomes these by combining LLMs with prompt engineering and RAG to enhance regulatory policy understanding and compliance verification precision. It specifically addresses these challenges and incorporates multi-agent communication for iterative refinement, improving efficiency in the compliance process. This work contributes to the broader, rapidly evolving field of applying generative AI to the software security lifecycle—spanning tasks such as regulatory compliance checking, risk analysis, code auditing, and testing [14].

Table 1. Involved AI Agents and Corresponding Roles

AI Agent	Role
CC_Agent 1	Analyzes software requirements specifications to identify potential non-compliant requirements with respect to a given policy document, using *LLM1*.
CC_Agent 2	Analyzes software requirements specifications to identify potential non-compliant requirements with respect to a given policy document, using *LLM2*.
RA_Agent	Compares compliance reports generated by multiple CC_Agents, identifies discrepancies, and instructs the *CC_Agents* to revise their analysis. Generates the final report using *LLM3*.

3 The Compliance Checking Framework

The proposed framework consists of various software components designed to involve multiple AI agents and facilitate their communication. This setup allows for the analysis and iterative refinement of requirement specifications to detect non-compliant requirements effectively. Key components include *(i)* the *RAG Document Pre-processor*, *(ii)* the *Compliance Checker* and *(iii)* the *Report Analyzer*. The latter two modules comprise multiple AI agents. Figure 1 and Table 1 illustrate the components and the roles of AI agents, respectively. Recapitulations of the NLP approaches used in the framework are kept online[2].

The setup involves two agents, *CC_Agent1* and *CC_Agent2*. The Mixtral 8x7B [20] model and Command R-plus[3] serve as their respective LLMs to provide diverse perspectives on identifying non-compliant requirements specifications. These agents perform compliance checks and analyze their results under the guidance of the *RA_Agent* from the *Report Analyzer* component. Based on the insights from the *RA_Agent*, the *CC_Agents* iteratively refine their criteria, ensuring continuous improvement in compliance assessment. The following sections will detail the functions of the *CC_Agents* and the *RA_Agent*.

3.1 Document Pre-processor

The framework leverages a Retrieval-Augmented Generation (RAG) approach to manage and analyze regulatory documents such as GDPR or the Data Protection Act, alongside software requirement specifications (SRS). The *Document Pre-processor* module, labeled A in Figure 1, initiates the compliance pipeline by systematically breaking down complex documents using an *Agent-Based Semantic Chunking*[4] mechanism. This method is specifically tailored to tackle the inherent complexity of legal and regulatory texts, which often include dense cross-references, domain-specific jargon, exceptions, and conditional structures.

At the core of the methodology is a LLM-driven agent responsible for segmenting documents into coherent, semantically meaningful units. This includes the simplification and elaboration of syntactically dense statements into clearer propositions, enhancing accessibility and reducing structural ambiguity. For example, clauses with multiple nested conditions are separated into distinct statements that individually represent each condition while maintaining the legal meaning. To further reduce referential ambiguity and improve retrieval effectiveness, each chunk undergoes a de-contextualization process. This involves resolving anaphoric references—such as pronouns like ”it” or ”this”—by replacing them with explicit noun phrases based on contextual cues within the document. Additionally, the chunking agent inserts key noun modifiers and ensures each unit is as self-contained as possible.

[2]https://doi.org/10.5281/zenodo.16176940
[3]https://cohere.com/blog/command-r-plus-microsoft-azure
[4]https://www.ibm.com/architectures/papers/rag-cookbook/chunking

While full de-contextualization remains challenging in highly intricate texts, the downstream compliance agents (*CC_Agents* and *RA_Agent*) are designed to re-integrate broader context during analysis.

The methodology also includes strategies for handling specific legal structures. Cross-references within documents (e.g., "as defined in Article X") are identified and marked with placeholder tokens or maintained explicitly for contextual linking. In the case of inter-document references, these are flagged for retrieval from other indexed regulatory sources. Exception clauses (e.g., "unless condition A applies") are retained in conjunction with their governing statements to preserve their logical dependencies. The agent ensures that such logical constructs remain intact to facilitate accurate interpretation by downstream components. Moreover, the chunking agent leverages the LLM's broad pretraining to understand and appropriately handle domain-specific legal and technical language, without relying on an external lexicon. This capability supports the generation of semantically accurate segments that respect terminology boundaries specific to regulations like GDPR or the Data Act. Named entities such as "Data Controller" or "Personal Data Breach" are extracted and restructured into discrete propositions to improve clarity and enhance downstream semantic retrieval.

The pre-processed and semantically enriched chunks from both the policy and SRS documents are then transformed into vector embeddings. These embeddings are later used by the compliance agents to perform targeted, semantically guided information retrieval during the assessment process. This stage is foundational to addressing **RQ1** by equipping the system with context-rich document representations.

3.2 Compliance Checker

As illustrated in Figure 1, the *Compliance Checker* module is denoted by B. In this module, two agents, *CC_Agent1* and *CC_Agent2*, employ two distinct LLMs to carry out the compliance check process. This is to prevent bias towards any particular LLM. We elaborate all the techniques that agents use to perform compliance checks.

3.2.1 Prompt Engineering via Chain-of-Thoughts. Compliance checking involves breaking down tasks. The *RA_Agent* reviews reports from *CC_Agents* and further divides tasks for thorough analysis. For each compliance check, an executed plan with various prompts and responses is created. Essential prompts include extracting requirements, assessing regulatory alignment, and explaining compliance or non-compliance. CoT is key for analyzing complex issues, offering better performance than direct or few-shot prompts, which may miss detailed requirements or depth in legal reasoning. The success of the framework depends on a well-designed prompt engineering strategy following a structured methodology for transparency, accuracy, and reproducibility.

A. Initial Prompt Design Principles. The initial set of prompts for both the `CC_Agents` and the `RA_Agent` were crafted based on three key principles designed to elicit the most accurate and transparent reasoning from the LLMs:

(1) *Persona Prompting:* Each high-level prompt begins by assigning a specific role to the LLM (*e.g., "As a GDPR compliance expert, analyze..."*). This technique prepares the model to adopt the specific knowledge, terminology, and reasoning patterns associated with that expert persona, leading to more domain-aware and relevant outputs.

(2) *Chain-of-Thought (CoT) for Query Decomposition:*
Chain-of-Thought (CoT) prompting significantly enhances the rigor of LLM-based regulatory analysis. By enforcing a step-by-step methodology, it enables granular scrutiny of data against specific legal clauses, identifies complex interdependencies between regulatory principles, and grounds the model's reasoning in precise legal citations. This approach also improves the detection of subtle or implicit violations, such as improperly bundled consents, leading to a more thorough and legally defensible compliance assessment. Figure 2 demonstrates how the LLM adapts to decompose complex compliance queries. Figure 3 proposes the prompt structure for query decomposition into subqueries. Subqueries allow for better document

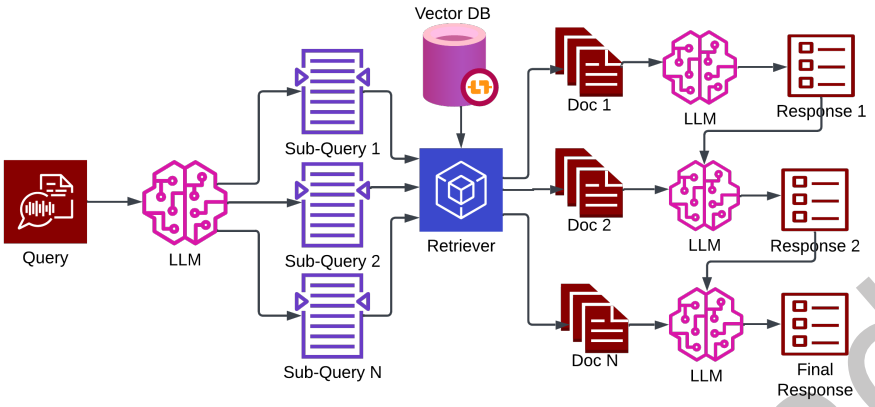


Fig. 2. Query Decomposition using CoT

targeting. *CC_Agents* coordinate with *RA_Agent* ensuring contextual continuity through conversational history. The decomposed prompts, constructed from high-level prompts, are then executed in sequence. A sample prompt breakdown is shown below.

Example: Consider this High-level Prompt: "*As a GDPR compliance expert, analyze the software requirements specification provided regarding personal data handling, storage and processing. Identify potential areas of non-compliance with the GDPR articles. The GDPR articles are provided in the context.*" Summary of the sub-prompts as follows:

- *Sub-prompt 1:* List Requirements Related to Data Privacy, Protection, and Processing.
  - Identify requirements addressing data privacy, protection, and processing.
  - Look for keywords such as "personal data," "consent," and "data retention."
- *Sub-prompt 2:* Identify Relevant GDPR Clauses.
  - Map each requirement to relevant GDPR articles (e.g., Articles 5, 6, 12-23, 24-43).
- *Sub-prompt 3:* Evaluate Compliance Level.
  - Assess whether each requirement is compliant, partially compliant, or non-compliant with the GDPR clauses.
- *Sub-prompt 4:* Identify Gaps and Suggest Improvements.
  - For non-compliant requirements, identify gaps and suggest improvements to meet GDPR standards.

You are a helpful assistant that creates and generates multiple sub-questions related to an input question. The goal is to break down the input into a set of sub-problems / sub-questions that can be answered in isolation.

Assess the given question and break it into 3 - 4 subproblems. Create a plan to execute each subproblem step by step. Question : {question}

Output (Execution plan with 3 - 4 queries):

Fig. 3. Prompt to instruct LLM for query decomposition

Note. For clarity, we adopt the Alpaca prompt template[33] to ensure reproducibility and avoid ambiguity in prompt construction.

(3) *Contextual Scaffolding:* To prevent hallucination and ground the model analysis, the prompts are designed to work within the Retrieval-Augmented Generation (RAG) framework. The prompts explicitly instruct the LLM to base its analysis *only* on the provided context (i.e., the retrieved SRS chunks and regulatory articles), ensuring that all generated reports are directly traceable to the source documents.

B. Iterative Tuning via Human-in-the-Loop Feedback. The initial prompts served as a baseline and were systematically refined through an iterative tuning process driven by expert review, utilizing the Human-in-the-Loop (HITL) mechanism described in Section 3.4. This loop was crucial to addressing failures and improving the nuance of compliance checks. The process was as follows.

(1) *Baseline Execution:* We executed the framework with the initial prompts on a validation set of software projects.
(2) *Expert-Led Error Analysis:* The authors, acting as domain experts, reviewed the generated compliance reports, specifically looking for errors such as false negatives, false positives, or weak rationales.
(3) *Hypothesis-Driven Modification:* For each identified error, we analyzed the prompt that led to the failure and formed a hypothesis for improvement.
(4) *Re-evaluation:* The modified prompt was tested in the same case to verify that the error was corrected without introducing new issues. This cycle was repeated until satisfactory performance was achieved.

Example. To make this process concrete, consider the following example:

- *Initial Observation:* In an early iteration, the framework failed to flag the requirement *"User passwords will be stored and transmitted."* as non-compliant with GDPR.
- *Error Analysis:* The model focused only on the general act of storage and transmission but missed the *manner* of storage (i.e., the lack of specified encryption). The baseline prompt for "data security" was too general.
- *Hypothesis:* We hypothesized that the prompt needed to be more specific by explicitly referencing encryption and data-in-transit/at-rest principles from GDPR Article 32.
- *Modified Sub-Prompt:* We refined a sub-prompt in the CoT sequence to be more explicit:
  "For each requirement related to user credentials or sensitive data, verify that it explicitly mandates strong, state of the art encryption for both data-at-rest (in storage) and data-in-transit (during transmission), as required by GDPR Article 32. If encryption is not mentioned, flag it as a critical gap."
- *Result:* Upon re-evaluation with the refined prompt, the framework correctly identified the requirement as non-compliant and provided a precise rationale referencing the need for encryption under Article 32.

Detailed comparative examples illustrating these advantages for GDPR scenarios (data minimization, purpose limitation, consent validity) are provided in the supplementary document

3.2.2 Retrieval Augmented Generation. The *Compliance Checking* module analyzes requirements specification documents to identify potential non-compliant elements relative to a given policy. It uses advanced techniques such as prompt engineering and RAG (see Fig. 4) for efficient processing. The RAG mechanism involves saving vector embeddings of documents in vector databases, followed by *CC_Agents* receiving queries to recognize non-compliant requirements and engaging in CoT reasoning to address subtasks individually. The methodology for automated compliance verification using a multi-agent framework comprises of three key components:

- *Retrieval of Relevant Requirements: CC_Agents* use a document retriever to obtain specifications related to data protection, privacy, and security from the Requirements Vector DB. The LLM then assesses their alignment with relevant policy articles like GDPR, or the Data Act.

- *Retrieval of Targeted Policies: CC_Agents* utilize the Policy Vector DB to efficiently retrieve the most relevant policy documents based on specific requirements, thus enhancing compliance analysis with supplementary context.
- *LLM-Generated Compliance Summary:* The LLM evaluates all pertinent requirements and regulatory policy documents, producing a compliance report containing a summary of compliance findings, identifying potential non-compliant requirements, and suggesting measures for full compliance.

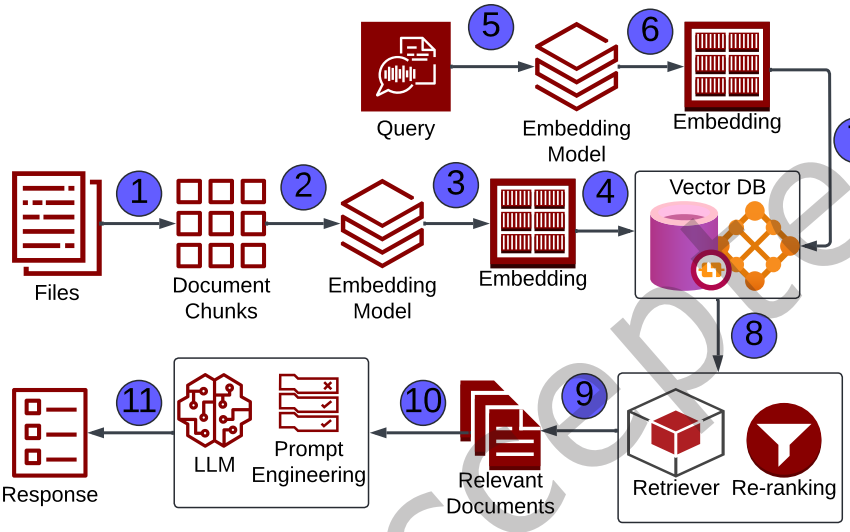


Fig. 4. Retrieval Augmented Generation

3.2.3 Enhancement of Retriever Using Re-ranking. A standard RAG pipeline can suffer from the 'Lost in the Middle' phenomenon [23], where LLMs struggle to utilize information located in the middle of a long context. To address this and enhance overall retrieval quality, our framework incorporates a re-ranking mechanism. This mechanism transforms the retrieval from a single step into a more robust two-stage process designed to balance high recall with high precision.

In the first stage, a computationally efficient retriever (e.g., based on vector similarity) casts a wide net to fetch a set of potentially relevant candidate documents. This stage prioritizes recall, ensuring that no important information is missed.

In the second stage, a more sophisticated and computationally intensive re-ranking model, such as a transformer-based cross-encoder, is applied exclusively to this smaller candidate set. Unlike the initial retriever, which compares embeddings independently, a cross-encoder processes the query and each candidate document *together*, allowing it to capture much deeper contextual and semantic similarities. This refinement stage provides several key benefits:

- *Improved Precision and Relevance:* The re-ranker intelligently promotes documents that are most directly relevant to the query to the top of the list, filtering out thematically similar but less useful results.
- *Mitigation of the 'Lost in the Middle' Effect:* By placing the most critical documents at the beginning of the context window, we ensure this information is in the optimal position for the final LLM to process, directly addressing this common failure mode.

- *Efficient and Focused Analysis:* This two-stage approach provides a cleaner, less noisy context to the final LLM, reducing the cognitive load on the model and leading to more faithful and accurate generated responses, while remaining computationally efficient by not applying the expensive cross-encoder to the entire knowledge base.

The re-ranking mechanism employs a two-stage process to enhance document retrieval efficiency and accuracy (see Fig. 5). In the first stage, a lightweight ranking model, such as vector similarity, filters out irrelevant documents, retaining a subset of potentially relevant candidates. In the second stage, an advanced ranking model, like a cross-encoder model, re-ranks these candidates by capturing semantic similarities and contextual information. In addition, criteria from *RA_Agent*, such as legal compliance, risk assessments, and ethical considerations, are used to prioritize the most relevant and appropriate sources of information. This two-stage approach reduces the computational burden on advanced ranking models. The mechanism aims to address the 'Lost in the Middle' phenomenon, improve relevance and accuracy, and optimize computational resource utilization.

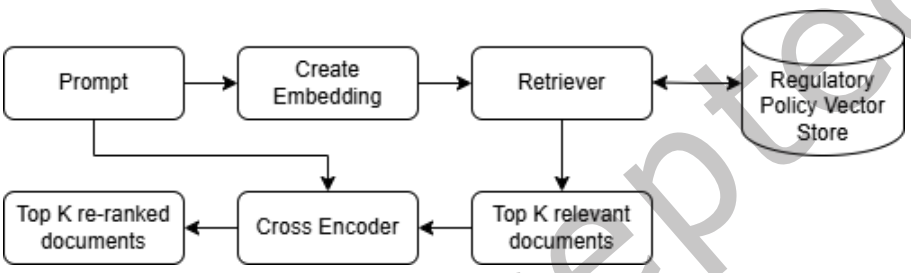


Fig. 5. Retrieved Documents Re-ranking

3.3 Report Analyzer

The module responsible for the analysis of the report, denoted C in Figure 1, relies on a multi-agent communication workflow. The *RA_Agent* retrieves compliance checking reports from *CC_Agents* and performs a comparative analysis to identify discrepancies in non-compliant requirements. The agent systematically examines the reports to highlight inconsistencies and determines if any non-compliant requirements were missed by one agent but identified by another. This ensures a thorough analysis and helps to uncover potential gaps in compliance assessment.

The Report Analyzer (RA) agent has enhanced capabilities that elevate its role beyond simple report comparison. These features enable the *RA_Agent* to provide more nuanced feedback, refine the analysis of *CC_Agents*, and ultimately contribute to more accurate and reliable compliance verification.

3.3.1 Web Search and Knowledge Integration. The *RA_Agent* is equipped with the ability to perform web searches and integrate external knowledge related to best practices and relevant information concerning regulatory compliance in software development. This dynamic knowledge acquisition allows the *RA_Agent* to go beyond the information provided by the *CC_Agents* and access a broader range of resources. For instance, if a *CC_Agent* flags a requirement related to data retention as potentially non-compliant, the *RA_Agent* can search for specific guidelines and best practices related to data retention under GDPR. This enriched understanding allows the *RA_Agent* to better assess the compliance report, identify potential nuances that the *CC_Agents* might have missed, and provide more targeted feedback for improvement.

Privacy-Preserving Query Generation: In order to prevent the leakage of proprietary SRS information during web searches, the framework employs a *Query Abstraction Protocol*. Prior to external interaction, the system

executes a sanitization step inspired by techniques such as Truthful Text Sanitization [28]. This process identifies and replaces project-specific entities with abstract legal categories (e.g., transforming "Project X's 10-year storage of driver scans" into "long-term retention of biometric employee data"). The *RA_Agent* then formulates a purely regulatory question based on this abstraction (e.g., "GDPR retention limits for biometric data"). Consequently, only generic concepts are transmitted to external search engines, ensuring that sensitive application context remains confined within the secure inference environment.

3.3.2 Memory Management for Human Feedback Integration. The effectiveness of the *RA_Agent* is largely attributed to its sophisticated memory management system, which not only archives past interactions but also seamlessly incorporates human feedback as a fundamental component. As the agent is responsible for evaluating compliance reports related to complex regulations, the expertise of human stakeholders is essential to refine its decision-making abilities. This advanced memory management framework enhances the performance of the *RA_Agent* in compliance assessments by organizing information into structured layers that support both short-term and long-term memory. It efficiently stores recent interactions and task-specific data while also retaining general knowledge and expert insights. This dual-layer approach fosters continuous learning, enabling the agent to adapt and improve its understanding of evolving regulatory requirements.

By integrating human feedback, the *RA_Agent* refines its decision-making capabilities and becomes more adept at navigating complex scenarios. The system ensures that frequently accessed information is readily retrievable, optimizing performance and ensuring that the agent remains accurate and responsive in the face of a dynamic regulatory landscape. A detailed explanation of the memory management scheme is provided in Section 3.4.2.

In this context, the process of report analysis is summarized in the following steps.

Step-1: Evaluating Missed Non-Compliance.
The *RA_Agent* analyzes the rationale provided by *CC_Agents* for a non-compliant requirement missed by another *CC_Agent*. This step involves validating the rationale to ensure that the requirement actually violates regulatory provisions, thereby confirming the reliability of the findings.

Step-2: Prompting Revision.
The *RA_Agent* generates messages for *CC_Agents* to revise their analysis of missed non-compliant requirements. These messages detail the missed requirement, the rationale for noncompliance, and instructions for re-evaluation, promoting collaboration and iterative improvement. *CC_Agents* are encouraged to provide detailed explanations if they disagree with the noncompliance assessment.

Step-3: Reviewing Updated Reports.
The *RA_Agent* reviews updated compliance reports from *CC_Agents* to verify that previously missed non-compliant requirements are addressed, including incorporating feedback and satisfactory explanations for disagreements.

Step-4: Compiling Final Report.
The *RA_Agent* compiles a comprehensive final compliance report from the updated reports provided by the *CC_Agents*. This report provides a clear summary of the compliance status and actionable recommendations for stakeholders.

Step-5: Memorizing Human Feedback and Review. Before implementing any changes to the SRS document, the framework incorporates a mandatory review stage involving a human expert (requirements engineer, legal expert). The human expert examines the compliance report generated by the *RA_Agent*. The expert examines the compliance report and evaluates each identified issue, providing input on whether modifications are appropriate or if further analysis is needed. For each decision, the human expert must provide a brief justification. This feedback helps in identifying discrepancies between the *RA_Agent*'s analysis and the expert's judgment. The *RA_Agent*'s memory management system is key to the human-in-the-loop process. Expert feedback, along with rationales, is stored in the agent's long-term memory as training

data. This allows the *RA_Agent* to learn from human input, refine its understanding of compliance, and improve accuracy in the analysis of future reports. Human-in-the-loop mechanism is elaborated in the subsequent section.

3.4 Human-in-the-Loop (HITL)

The proposed framework acknowledges the crucial role of human expertise in navigating complex legal regulations and making informed decisions. To ensure accuracy, reliability, and adaptability, the framework incorporates a robust HITL mechanism. The respective module of the HITL is indicated as D in Figure 1. This mechanism seamlessly integrates human judgment into the automated compliance workflow, facilitating validation, refinement, and continuous improvement of the system analysis. The HITL mechanism serves two primary objectives. (i) *Expert Validation and Refinement*, where requirements engineers and legal experts review and validate AI-generated compliance reports, addressing edge cases and improving the accuracy and legal soundness of the system; and, (ii) *Continuous Improvement*, where Human feedback is systematically used to improve the reasoning of the framework, improve regulatory understanding, and adapt to evolving legal standards.

3.4.1 Stages and Interactions. The HITL mechanism is structured into a series of well-defined stages, designed to facilitate meaningful human interaction and feedback:

- *Mandatory Compliance Review by Experts:* Following the automated analysis, the compliance report is presented to the requirements engineers and legal experts for review. This review is facilitated through an interactive platform that provides:
  - *Issue Summary:* A concise overview of potentially non-compliant elements identified in the SRS.
  - *Regulation Mapping:* Direct links to the specific regulation clauses potentially violated by each flagged issue, allowing easy reference to relevant legal text.
  - *Rationale Visualization:* A transparent presentation of the reasoning process employed by each *CC_Agent*, highlighting areas of agreement and disagreement, along with the *RA_Agent*'s justifications for its conclusions.
  - *Confidence Scores:* Each identified issue is accompanied by a confidence score assigned by the *RA_Agent*, reflecting the level of certainty in its assessment.
- *Human Decision and Feedback:* Experts interact with the interactive report and provide their judgment on each flagged issue.
  - *Accept:* The expert agrees with the *RA_Agent*'s assessment and accepts the suggested modifications (if any) to the SRS.
  - *Reject:* The expert overrules the *RA_Agent*'s assessment, indicating that the requirement is compliant and providing a justification for their decision.
  - *Escalate:* When the compliance issue involves ambiguous or complex legal nuances that require input from specialized legal experts, the expert flags the issue for further analysis by a legal expert.

  Each decision (accept, reject, escalate) is accompanied by a brief justification, offering insights into human reasoning and legal interpretation to refine the framework and prompt strategies.

3.4.2 Memory Management Scheme. The framework efficiently maintains expert feedback through a structured memory management scheme, enabling AI agents such as *RA_Agent* and *CC_Agent* to learn continuously from past interactions and expert input. By organizing memory into layered components, this system allows these agents to retain both recent, task-specific information and long-term knowledge derived from cumulative human insights. This approach ensures that the agents evolve in their decision-making, adapting to complex scenarios with improved accuracy and responsiveness over time. The memory management scheme is structured into multiple layers:

- *Main Context:* This layer handles the active short-term memory required for real-time compliance assessments and human interactions. It comprises:
  - *System Instructions (SI):* Directs the agent's task processing and rule application.
  - *Working Context (WC):* Stores immediate task-specific information, including scenario-based human feedback and current compliance rules.
  - *Conversational Context (CC):* Maintains a history of the ongoing conversation between the agent and the user, allowing contextual understanding across interactions.
- *Long-Term Memory:* This layer stores generalized knowledge derived from repeated examples and human feedback. It includes:
  - *Conversational Summary:* Summarizes key insights from past user dialogues.
  - *Event Summary:* Stores details about specific compliance reports, rules, and analyzed events.
  - *Facts:* Contains general regulatory knowledge and facts frequently used in assessments.
  - *Scenario-Based Expert Feedback:* Stores human expert feedback on compliance errors and improvements.
  - *Feedback with Reasons and Flaws:* Logs past mistakes and critiques from human reviewers, facilitating iterative learning.
- *Cache Management:* Redis is employed as a caching layer to optimize memory retrieval for frequently accessed information.

3.4.3 Feedback Integration and System Learning. The human feedback collected during the review process is systematically integrated into the framework for continuous improvement:

- *Data Collection:* Expert decisions, justifications, and associated compliance issues are logged and stored within the agent's memory management system.
- *Short-term Memory Update:* The system's short-term memory is updated with specific examples and feedback, allowing immediate learning and adaptation to similar scenarios.
- *Long-term Memory Update:* Generalized rules and principles derived from expert feedback are integrated into the long-term memory of the system, facilitating greater learning and adaptation to evolving regulatory landscapes.
- *Agent Retraining:* The AI agents, particularly the *RA_Agent* and *CC_Agents*, are retrained based on accumulated human feedback. This retraining refines the agent's prompt engineering strategies, ensuring that future compliance checks better align with expert interpretations and legal judgments.

This HITL mechanism ensures that the framework benefits from both the efficiency of automated analysis and the nuanced judgment of human expertise. The continuous feedback loop fosters continuous improvement, leading the system to a more robust, reliable, and adaptable.

4 Deployment of the Framework

This section presents the details of the implementation of the proposed framework, including the tools, models, and databases used to realize each conceptual component described in Section 3.

4.1 Document Pre-processing

The implementation begins with preprocessing two document types: (i) the regulation policy document and (ii) the software requirements specification (SRS). These are handled by an LLM-based chunking agent which performs the semantic processing strategies described in Section 3.1. Following chunking, the semantically enriched segments are vectorized using the *BGE-large*[12] embedding model, chosen for its exceptional performance on the MTEB dataset[25], open-source availability, and lightweight design. The resulting embeddings are indexed using the *Faiss* vector database, which structures them for efficient semantic retrieval. Faiss indexing techniques

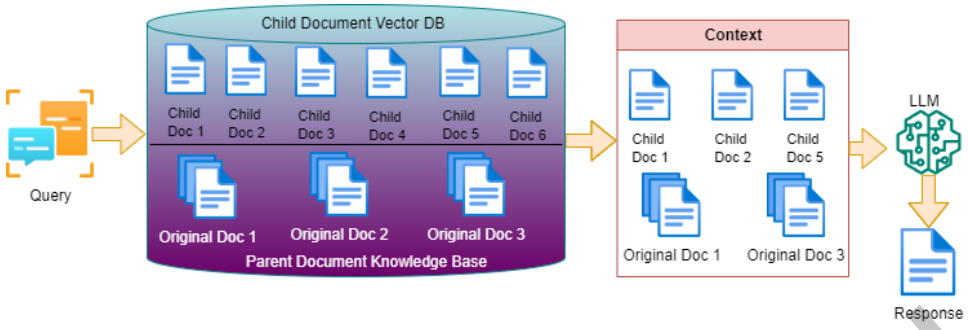


Fig. 6. Parent Document Retrieval

enable for rapid similarity searches in high-dimensional vector spaces. Separate indexes are maintained for the policy and requirement documents to facilitate context-aware retrieval during compliance checking. This implementation ensures that the conceptual pipeline laid out in Section 3.1 is realized through robust tooling and infrastructure, preparing the documents for effective use in the downstream compliance verification modules.

Note. We do not advocate the use of specific LLMs. Instead, our framework is designed with a plug-and-play architecture, enabling end users to select LLMs that align with their preferences.

4.2 Compliance Checking

This module integrates the theoretical constructs described in Section 3.2 into an executable system. The implementation focuses on query decomposition using CoT, optimized document retrievers, generative LLM integration, prompt engineering, and a re-ranking pipeline.

4.2.1 Integrated Retriever. After creating vector indices (Section 4.1), an efficient retriever mechanism is essential. Standard retrievers like EmbeddingRetriever [34] and DensePassageRetriever [24] struggle with balancing document size—small documents must preserve meaning, while lengthy ones risk losing context. The *Parent Document Retriever* resolves this by splitting data into chunks, retrieving them, and then fetching their parent documents to ensure context retention (see Figure 6).

We propose an approach that combines the KNN Retriever[8] and the Parent Document Retriever, along with the Maximal Marginal Relevance (MMR) concept[19] to enhance information retrieval by maintaining diversity in retrieved content. The KNN Retriever integrates the Dense Passage Retriever and the K-Nearest Neighbors algorithm to identify documents closest to a query using vector similarity. It employs a multi-step mechanism involving:

(a) Converting text into vector embeddings using a BGE-large pre-trained model.
(b) Creating an index for the embeddings with FAISS to optimize search operations.
(c) Applying L2 normalization to index and query embeddings.
(d) Calculating cosine similarity between query and indexed documents.
(e) Retrieving the top k documents based on similarity scores.
(f) Applying Maximum Marginal Relevance (MMR) to rank and filter these documents, balancing relevance (cosine similarity) and diversity to ensure comprehensive and non-redundant results.

4.2.2 Selection of Generative LLMs. The Compliance Checking module of the framework uses two AI agents: *CC_Agent1* and *CC_Agent2*, powered by the Mixtral 8x7B model and the Command R+ model, respectively.

Mixtral supports a 32K token context, while Command R+ offers 128K, making both ideal for RAG applications. These models are open-source and accessible via APIs from Groq (free) and Cohere (free until production). The *RA_Agent* uses Claude-Opus, which supports a 200K-token context window and has outperformed GPT-4 in recent benchmarks [5]. Although it is not open-source, Claude-Opus maintains the balance between cost and performance—processing a 20K-token document for under $1. This setup enables the agent to conduct compliance checks and analyze long reports efficiently.
The choice of Mixtral 8x7B and Command R+ for the two *CC_Agents* was a deliberate design decision to validate the core principles of our framework. Our methodology is fundamentally model-agnostic, but these specific models were selected for our experiments based on three key criteria:

(1) *Architectural Diversity:* The primary goal of our multi-agent system is to mitigate the risk of single-model bias. By employing two high-performing models from different developers (Mistral AI and Cohere), we ensure they approach the compliance task from distinct architectural perspectives. This diversity is crucial for the *RA_Agent*'s cross-validation role, as it can more effectively identify and reconcile discrepancies that stem from different reasoning paths.
(2) *Large Context Windows:* A critical, non-negotiable technical requirement for our use case is the ability to process lengthy regulatory and requirements documents. Mixtral (32K tokens) and Command R+ (128K tokens) both offer large context windows, making them technically suitable for the demanding RAG tasks in our pipeline.
(3) *Strong General-Purpose Reasoning:* We aimed to demonstrate that our framework's *architecture*—not a hyper-specialized model—is the key contributor to performance. We therefore selected models renowned for their state-of-the-art general purpose and instruction-following capabilities [20]. Their success in our experiments validates that our structured, multi-agent approach can effectively guide general-purpose LLMs to perform highly specialized tasks like regulatory compliance checking.

Note: The choice of LLMs (Mixtral 8x7B, Command R+, and Claude-Opus) is illustrative rather than prescriptive—the framework is model-agnostic, with no hard dependencies on specific architectures. These models were selected for demonstration purposes on the basis of their context lengths, cost efficiency, and benchmark performance. Alternatives (e.g., GPT-4, Gemini, or open-source models such as Llama 3) can be substituted without structural changes to the framework.

4.2.3 Enhancing the Retriever with Re-ranking. The re-ranking mechanism in our framework follows the multi-stage approach of Nogueira et al. [26] and involves the following steps:

(a) *Initial Retrieval:* Relevant documents are initially retrieved from the vector database using the document retriever, based on the query vector (discussed in Section 4.2.1).
(b) *Subsequent Re-ranking Stages:* The retrieved documents are iteratively refined using a more sophisticated re-ranking model in subsequent stages.
(c) *Employing Re-ranking Model:* A transformer-based cross-encoder model, *mxbai-rerank-large-v1* [30], evaluates and re-ranks document passages based on relevance scores from a fine-tuned, labeled dataset of query-passage pairs. We chose mxbai-rerank-large-v1 because its benchmark performance (e.g., Accuracy@3 = 74.9 on BEIR) shows strong precision in re-ranking domain-specific queries. While this incurs higher compute cost than simple embedding similarity re-scoring, in our regulatory-compliance domain we felt the precision gains justified the cost.
(d) *Filtering and Refinement:* The re-ranking process integrates relevance scores with criteria from the *RA_Agent* (legal compliance, risk assessments, ethical considerations) to filter out non-compliant sources and ensure compliance and relevance.

(e) *Iterative Improvement:* The process iteratively refines search results through multiple stages, adapts to more context, and improves precision and relevance with each iteration.

This multi-stage re-ranking mechanism addresses LLM limitations by refining retrieved information iteratively, ensuring highly relevant and precise results to enhance LLM performance.

4.3 Report Analysis

This section discusses how prompt engineering enables the *RA_Agent* to effectively compare compliance reports, identify discrepancies, and prompt *CC_Agents* to revise their analysis. The *RA_Agent* identifies missed non-compliant requirements, instructs *CC_Agents* to update their reports, and compiles a final comprehensive compliance report. The process involves a series of prompts:

Prompt-(i): *Retrieve and compare compliance reports from the two CC_Agents to identify discrepancies in non-compliant requirements.*

Prompt-(ii): *Determine if one agent missed non-compliant requirements identified by the other agent and list them.*

Prompt-(iii): *Analyze the rationale for each missed non-compliant requirement to validate its non-compliance to regulatory provisions.*

Prompt-(iv): *Generate messages to CC_Agents to re-evaluate missed non-compliant requirements with valid rationales, including instructions and requests for detailed explanations if there is disagreement.*

Prompt-(v): *Review updated reports from CC_Agent1 and CC_Agent2 to ensure that all missed non-compliant requirements are addressed and explanations for disagreements are satisfactory.*

Prompt-(vi): *Prepare a report summarizing the requirements evaluated, their compliance status, non-compliant items, regulatory violations, and recommendations for compliance.*

Using this series of prompts, the *RA_Agent* ensures a thorough analysis of compliance reports, identifies and addresses missed non-compliant requirements, and compiles a comprehensive final report.

4.4 Memory Management for Agent Learning

The Human-in-the-Loop (HITL) mechanism supports the continuous improvement of the *RA_Agent* through a two-tiered memory system.

(i) Short-term memory focuses on rapid access to recent interactions and feedback, enabling the agent to adapt dynamically to ongoing tasks. It organizes scenarios as a graph structure using *RedisGraph*, allowing quick retrieval and contextual relevance. An *LRU cache* ensures efficient storage and access.

(ii) Long-term memory serves as the knowledge base, supporting reasoning over accumulated feedback and compliance rules. Using a combination of a knowledge graph (*Neo4j*) and vector-based retrieval, it efficiently connects concepts, regulations, and rules for contextual analysis. New knowledge derived from HITL feedback is continuously integrated into this memory, enriching the agent's reasoning capabilities.

Combined, these memory systems allow the *RA_Agent* to effectively retrieve and apply knowledge from recent experiences and historical data, improving the accuracy and reliability of compliance analysis over time. The detailed implementation is available online.[5]

4.5 Implementation Details and Hyperparameter Settings

To ensure transparency and facilitate reproducibility, this section details the key hyperparameter settings and model selection criteria used throughout our experimental evaluation. The settings were chosen based on established best practices for RAG systems and preliminary testing to balance performance with computational efficiency (Refer Table 2). It is worth mentioning that - a key strength of our architecture is its model- and

[5]https://doi.org/10.5281/zenodo.16176940

Table 2. Key Hyperparameter Settings for the Experimental Framework

Component / Parameter	Setting / Value	Rationale
1. RAG Pipeline Configuration		
Embedding Model	`bge-large-en-v1.5`	High performance on MTEB benchmark for retrieval tasks.
Vector Database	Faiss with `IndexFlatL2`	Provides exact, brute-force similarity search for maximum accuracy.
Chunking Strategy	Agent-Based Semantic Chunking	Dynamically determines chunk boundaries to preserve semantic integrity.
Max Chunk Size	2000 tokens	Acts as an upper bound to prevent overly long chunks while allowing flexibility.
Retrieval Top-K	`k = 15`	Casts a wider net for initial candidates, relying on the re-ranker for precision.
Re-ranking Model	`mxbai-rerank-large-v1`	A powerful transformer model used to re-score and re-rank candidates for relevance.
2. Generative LLM Agent Configuration		
CC_Agents	Mixtral 8x7B, Command R+	Strong comprehension with large context windows for analysis.
- *Temperature*	`0.2`	Low temperature to favor factual, deterministic outputs.
RA_Agent	Claude 3 Opus	State-of-the-art model for complex report analysis and synthesis.
- *Temperature*	`0.1`	Even lower temperature for maximum factuality in the final report.

component-agnostic design, which facilitates the integration and evaluation of alternative configurations, making it a flexible platform for future experimentation.

5 Explainable AI (XAI) for Enhancing Transparency

While automation is crucial for efficient compliance checks, transparency and interpretability are equally vital to building trust and ensuring responsible use of AI-driven decisions. To address this need, our framework incorporates explainable AI (XAI) techniques, providing clear and interpretable insights into both compliant and non-compliant requirements identified by the system. These insights are central to our Human-in-the-Loop (HITL) approach, allowing human reviewers to understand, validate, and refine the analysis of AI agents. Through XAI, reviewers gain access to rationales, supporting evidence, and transparent decision planning, helping them understand the logical basis for compliance decisions. Furthermore, interpretable outputs allow reviewers to identify potential errors or biases and provide corrective feedback, leading to an iterative improvement of the framework.

Our XAI implementation focuses on three core aspects:

(i) *Retrieval Path Explanation:* This aspect aims to clarify the origin and relevance of the information used to support compliance decisions. Our framework employs the Parent Document Retriever, which efficiently retrieves relevant documents from the vector database containing both policy and requirement specifications. The XAI component goes beyond simply presenting the entire retrieved document. It pinpoints the specific chunks and sentences that were deemed the most relevant during the compliance analysis by the *CC_Agents* and *RA_Agent*. This is achieved by recording and displaying the document IDs, chunk IDs, and semantically similar sentences that were used during the analysis process. The framework creates a direct and transparent link between the compliance decision and its source information, allowing human reviewers to quickly verify the basis for the AI assessment.

(ii) *Rationale Generation:* To ensure that compliance decisions are understandable, our framework generates natural language explanations detailing the reasoning process of both the *RA_Agent* and *CC_Agents*. Using a CoT prompting technique, the system produces step-by-step rationales. These rationales explicitly reference the retrieved evidence, explaining which specific segments of regulatory policy were considered, how the identified requirements align or conflict with those articles, and the logic behind the agent's final decision. This transparent articulation of the AI agent's thought process enables human reviewers to gain a deeper understanding of the analysis, validate the decision-making process, and identify potential areas for refinement.

(iii) *Transparent Retrieval Process:* To increase user trust in the retrieval mechanism, the framework makes the retrieval process transparent by visualizing the internal steps taken by the agents during retrieval and

response generation. After the ingestion of relevant information - requirement specifications, regulatory policy documents by the *CC_Agents* and compliance reports by the *RA_Agent* - the XAI component provides a visualization of the actual CoT creation process. This visualization shows the dynamically generated prompts, based on the internal reasoning of the agent, during each step of the CoT process. This allows the human reviewer to trace the agent's decision-making path, understanding how the agent formulated its queries and which criteria led to the selection of specific documents. By exposing these internal mechanisms, the framework enhances user trust and allows a deeper understanding of the retrieval process.

Our XAI implementation enhances the transparency and trustworthiness of the compliance check framework. By providing interpretable insights into the analysis of AI agents, XAI supports the HITL process, allowing reviewers to understand compliance decisions, validate their accuracy through highlighted evidence and rationale, detect errors or biases, and refine the framework to improve prompt engineering, the knowledge base, and overall system performance.

False Negative Assessment

One of the significant challenges in automated compliance checking is the risk of false negatives - where non-compliant requirements are incorrectly marked as compliant. The XAI techniques of our framework play a vital role in mitigating this risk by providing human reviewers with tools to identify and address these errors effectively.

Through retrieval path explanations, human reviewers can detect gaps in the retrieved information that might suggest a false negative. For example, if a requirement relates to data breach notification but lacks references to GDPR Article 33 (Personal Data Breach), a human reviewer might identify this as a missing critical piece of compliance evidence. Additionally, the rationale generation component enables the reviewers to scrutinize the AI agent's reasoning, exposing misunderstandings or misapplications of GDPR principles that could result in an incorrect assessment. Visualizing the agent's retrieval process also helps identify potential flaws in query formulation, highlighting instances where essential documents might be omitted due to incomplete or inaccurate instructions.

For instance, consider a requirement stating that "*User passwords will be stored in plain text.*" If the system mistakenly marks this as compliant by focusing only on general data storage guidelines, it misses the violation of Article 32 of the GDPR, which mandates encryption for data security. XAI insights can reveal this oversight by showing a lack of reference to Article 32, highlighting an inadequate rationale, or exposing missing keywords in the prompt generation. By detecting such errors, XAI enables reviewers to prevent serious compliance breaches, significantly enhancing the reliability and trustworthiness of the compliance check framework.

6 Experimental Evaluation

The evaluation of the framework is based on a two-fold strategy, as traditional metrics such as BLEU are not very suitable [22]. The first part evaluates the retrieval and generation methods of the approach using the RAGAS framework [15]. In the second phase, we incorporate qualitative assessment through user feedback on the understandability and applicability of the generated explanations. This combined quantitative and qualitative approach provides a more comprehensive evaluation of the effectiveness of the framework, addressing the challenges highlighted in **RQ3**.

6.1 Evaluation of RAG pipeline

We use the RAGAS [15] (Retrieval Augmented Generation Assessment) framework to evaluate the performance of the integrated RAG mechanism. Traditional metrics like BLEU and ROUGE focus on surface-level text similarity

and are not suitable for RAG applications [22]. RAGAS provides a more comprehensive and tailored evaluation that addresses key aspects of RAG systems.

- *Contextual Understanding:* RAGAs assesses the quality and relevance of the retrieved context. In contrast, traditional metrics focus on text similarity and often miss deeper contextual nuances.
- *Retrieval Component Evaluation:* RAGAs incorporates metrics to evaluate the search phase, ensuring that the information retrieved is relevant and comprehensive. Traditional metrics predominantly evaluate the generation phase, neglecting the quality of retrieval.
- *Fact-Checking and Faithfulness:* RAGAs emphasizes the faithfulness of generated responses to the retrieved context, aiming to reduce hallucinations and factual inaccuracies.

The RAGAS framework outlines four key metrics essential for evaluating the effectiveness of the RAG mechanism.

(i) *Context Recall:* Measures how well the retrieved context aligns with the question, focusing on including all relevant attributes from the ground truth. *Example:* For a query about GDPR consent requirements, context recall is high if the documents cover all aspects, such as conditions, clear communication and withdrawal rights.
(ii) *Context Precision:* Assesses the relevance of the retrieved context items to the query. *Example:* For a query about GDPR consent for children, precision is low if the documents only discuss general consent requirements.
(iii) *Faithfulness:* Evaluates the factual accuracy of the generated answer in relation to the retrieved context. *Example:* For a query on GDPR data transfers, faithfulness is low if the response misrepresents or omits key safeguards like contractual clauses or adequacy decisions.
(iv) *Answer Relevancy:* Determines how well the generated answer addresses the query without redundancy or deviation. *Example:* For a query about GDPR penalties, the relevance is low if the response discusses unrelated topics such as data subject rights.

6.1.1 Ground truth Generation. To generate ground truth data for the RAGAs framework, we prepared datasets from two distinct document categories crucial for compliance checking: (1) dense, legislative Regulatory Policies and (2) technical Software Requirements Specifications (SRS) documents. This dual approach allows us to assess the RAG pipeline's performance in both understanding complex legal language and extracting specific information from technical specifications.

The ground truth generation process for each category was tailored to its specific nature:

- *Focus for Regulatory Policies (GDPR, Data Act):* The ground truth generation process focuses on creating question-answer pairs that test the ability of the system to analyze and interpret legal text. The questions were designed to investigate the understanding of laws, clauses, rules, and specific legal terminology.
- *Focus for SRS Documents:* These documents were sourced from the publicly available PURE dataset. For these, the ground truth generation focused on creating question-answer pairs that assess the ability of the system to analyze data-centric requirements related to privacy, security, specific features, and non-functional requirements.

The ground truth dataset includes four essential components from the Retrieval Augmented Generation (RAG) pipeline:

- *Question:* A set of questions on which the RAG system will be evaluated.
- *Contexts:* The contexts returned corresponding to each question. This is a list of lists since each question can retrieve multiple text chunks.
- *Answer:* The generated answer corresponding to each question.
- *Ground Truths:* The expected answer to each question. This is a string that corresponds to the correct answer.

In our experiment, the ground truth generation process employs the advanced LLM, Claude 3-Opus by Anthropic, which outperforms GPT-4 in all benchmark datasets. To ensure high-quality and consistent output, we did not simply ask the model to summarize the text. Instead, we used a structured, two-step process. First, large documents were segmented into smaller, digestible chunks. Second, each chunk was passed to the LLM with a specific prompt designed to elicit relevant question-answer pairs.

> *Example Prompt for Ground Truth Generation:*
> `"You are an expert in creating evaluation datasets for RAG systems. Based only on the text provided below, generate 3-5 high-quality and diverse question-answer pairs.`
> `- The questions should test the understanding of the core concepts, obligations, or requirements in the text.`
> `- The answers must be concise and directly extracted or synthesized from the provided text. Do not use any external knowledge.`
> `- Both the question and the answer must be self-contained and fully understandable without the original context.`
> `Provided Text: [ Document Chunk ]"`

While this semi-automated approach provided a high-quality initial dataset, we recognized that no LLM-generated content can be trusted as a definitive "ground truth" without stringent human oversight. Therefore, every question-answer pair generated by this process was subjected to the rigorous multi-stage human validation protocol detailed in the following section, ensuring the final benchmark dataset is both accurate and reliable.

In order to ensure the quality and accuracy of the generated ground truth dataset, a rigorous multi-stage validation protocol was implemented, involving five human annotators. Two of the annotators were authors of this paper, possessing a deep familiarity with the project's goals. The other three were PhD students specializing in Requirements Engineering, ensuring a high level of domain expertise for the validation of both regulatory and software requirements documents. The validation process proceeded as follows.

- *Initial Annotation and Review:* The dataset, which comprises approximately 5,200 question-answer pairs, was evenly distributed among the five annotators. Each annotator was responsible for reviewing their assigned subset. The task was to verify each question-answer pair against its corresponding context chunk for:
  - *Relevance:* Is the question a meaningful query for the given context?
  - *Correctness:* Is the 'ground truth' answer factually correct according to the context?
  - *Clarity:* Are both the question and the answer clear and unambiguous?
  - *Completeness:* Does the answer adequately address the question based on the information available in the context?
- *Correction and Flagging:* Annotators were instructed to directly correct minor errors in phrasing or clarity. For pairs that were deemed irrelevant, incorrect, or highly ambiguous, the annotators flagged them and provided a brief justification for modification or removal.
- *Cross-Validation and Consensus Building:* Following the initial review, all flagged or significantly modified question-answer pairs were entered into a collective review pool. Then, this pool was independently reviewed by at least two other annotators on the team. Disagreements were resolved through a discussion-based process that aimed to reach a consensus. In cases where a clear consensus could not be reached, a majority voting rule was applied to make the final decision on whether to keep, modify, or discard the pair.

Table 3. RAG evaluation Datasets for Policies and SRS

Dataset	#Questions	Size (#tokens)
GDPR	1865	460K
DataAct	879	130K
E-Store	401	24K
CCTNS	365	18K
Virtual Education	378	20K
Video Search	412	24K
ZNIX	258	16K
E-Procurement	432	28K

Table 4. RAG evaluation using Mixtral 8x7B and Command R+ models on different policy documents and SRS

Policies	Model	Context Precision	Context Recall	Answer Relevancy	Answer Faithfulness
GDPR	Mixtral 8x7B	0.71	0.65	0.92	0.88
DataAct		0.82	0.81	0.89	0.93
E-Store		0.91	0.88	0.92	0.94
CCTNS		0.90	0.89	0.92	0.92
Virtual Education		0.91	0.90	0.91	0.94
Video Search		0.92	0.90	0.93	0.94
ZNIX		0.93	0.91	0.92	0.94
E-Procurement		0.89	0.90	0.93	0.92
GDPR	Command R+	0.81	0.78	0.94	0.91
DataAct		0.90	0.89	0.92	0.92
E-Store		0.89	0.88	0.91	0.91
CCTNS		0.89	0.89	0.92	0.91
Virtual Education		0.90	0.90	0.90	0.93
Video Search		0.91	0.91	0.92	0.91
ZNIX		0.93	0.92	0.92	0.93
E-Procurement		0.90	0.90	0.91	0.92

This multi-annotator, iterative validation process was crucial for refining the dataset, minimizing automated generation artifacts, and establishing a high-quality ground truth benchmark for evaluating the performance of our RAG pipeline.

6.1.2 Evaluation of RAG pipeline . The evaluation of the RAG mechanism using two different LLMs—Mixtral 8x7B and Command R+—is presented in Table 4. The assessment includes two policy documents: GDPR and the Data Act, with GDPR being the larger of the two. As document size increases, performance degradation is observed, primarily due to the 32K token context limitation of the Mixtral model, leading to reduced context precision and recall. Nevertheless, answer relevancy and faithfulness scores remain high, attributed to the re-ranking mechanism, which improves the selection of relevant information and mitigates hallucination.

Figure 7 illustrates the impact of context length on LLM performance in the RAG pipeline for large documents such as the GDPR regulatory policy. As the context length increases, both the context precision and the recall improve. For example, at a context length of 4K, both models show less than 50% precision and recall, but at 128K, the Command R+ model achieves over 82% precision and over 80% recall. The Mixtral model, with a 32K context length, reaches 73% precision and 67% recall. The graph also indicates a positive correlation between the length

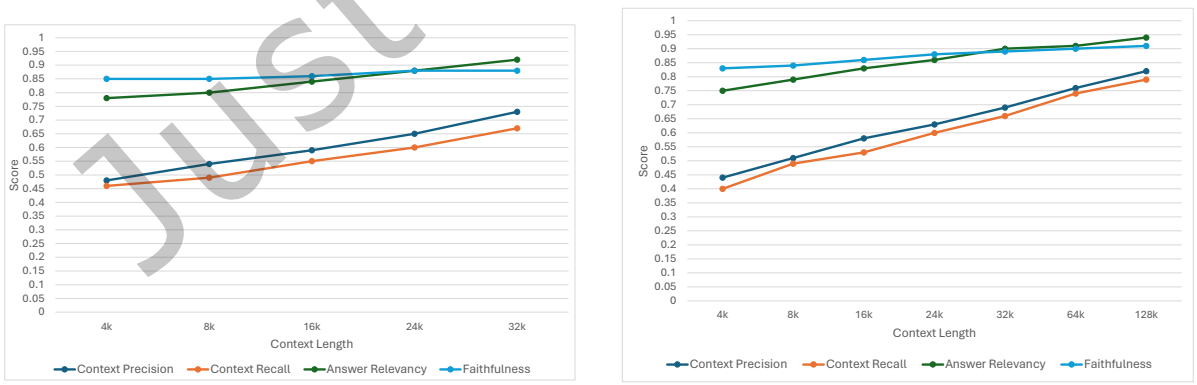
of the context and the relevancy and faithfulness of the responses. The Command R+ model, with a 128K token context length, significantly enhances performance by encompassing a broader context. Similar evaluations on smaller documents, such as SRS documents from the PURE dataset [18], are also presented in Table 4. The table shows that for these smaller SRS documents (less than 32K), both Mixtral 8x7B and Command R+ perform nearly identically, as the document sizes fall within the token context window of both models.

6.1.3 Proposed RAG vs. Baseline RAG. We compare the baseline RAG [23] with the proposed advanced variant of RAG, presented in this paper, that incorporates CoT reasoning, reranking, agent-based clustering, and parent document retrieval. As shown in Table 5, the advanced RAG consistently outperforms the baseline in all datasets and metrics, except for context precision in GDPR documents. Among these enhancements, the CoT reasoning is the most critical element, as it directs the pipeline through a refined step-by-step document analysis. CoT provides clear and structured instructions at each stage—guiding chunking, retrieval, and intermediate assessments, leading to a comprehensive and cumulative understanding of the document. The combination of CoT with other enhancements results in significant improvements in context precision, recall, answer relevancy, and faithfulness, demonstrating the effectiveness of this advanced RAG approach. A detailed comparative analysis of various prompting strategies—including CoT, direct prompting, and few-shot prompting—is provided in the supplementary document available online[6]

6.2 Evaluation of Compliance Analysis

We utilize a crowd-sourcing evaluation method for our proposed framework. This method has been used in recent studies [6, 29] to assess the acceptability of AI/ML-based solutions. This evaluation examines the framework's performance in 20 Data Act and 28 GDPR use cases, focusing on identifying incomplete requirements relative to regulatory policies. Using 270 undergraduate students and 20 domain experts as crowdworkers, it assesses the acceptability of detected non-compliant and incomplete requirements to ensure usability and reliability. Key factors include alignment with regulatory standards (e.g., GDPR) and whether identified issues are actionable and comprehensible for stakeholders. Insights from these experiments, including detailed results and observations, are discussed in the following sections.

[6]https://doi.org/10.5281/zenodo.16176940



(a) Mixtral model

(b) Command R+ model

Fig. 7. Effect on RAGAs metrics with variable LLM context length for the GDPR policy.

Table 5. Comparison of Baseline RAG and Advanced RAG with Command R+

Dataset	#Questions	Size (#tokens)	Context Precision		Context Recall		Answer Relevancy		Faithfulness	
			Baseline	Advanced	Baseline	Advanced	Baseline	Advanced	Baseline	Advanced
GDPR	1865	460K	0.77	0.71	0.62	0.65	0.75	0.92	0.87	0.88
DataAct	879	130K	0.82	0.82	0.74	0.81	0.79	0.89	0.87	0.93
E-Commerce	401	24K	0.83	0.91	0.78	0.88	0.84	0.92	0.88	0.94
CCNTS	365	18K	0.85	0.9	0.8	0.89	0.86	0.92	0.9	0.92
Online Streaming	426	25K	0.83	0.87	0.80	0.90	0.88	0.92	0.9	0.92
Video Search	506	27K	0.80	0.88	0.79	0.89	0.87	0.91	0.89	0.91

6.2.1 The Questionnaire. Two questionnaires were distributed to check compliance with GDPR and the Data Act, which contained 28 and 20 use cases, respectively. Each use case includes 4 to 5 specifications of natural language requirements and identified incompleteness in the requirements. Potential modifications are also provided to ensure compliance. The use cases are derived from the SRS documents in the PURE dataset, based on the experimental results. Crowdsource agents have four options for providing feedback on detecting and modifying non-compliant requirements.

(i) *Modified requirements are compliant*: Accuracy above 80%.
(ii) *Modified requirements are partially compliant*: Accuracy between 30-80%.
(iii) *Detected incompleteness is correct*: Accuracy above 80%.
(iv) *Detected incompleteness is partially correct*: Accuracy between 30-80%.

If crowdworkers do not select any of the options provided, it suggests that they find the proposed modifications or the incompleteness identified incorrect.

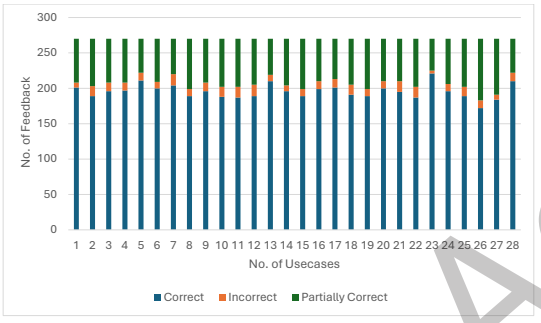
6.2.2 Specification of Crowdworkers. We collected feedback from two distinct groups: a large cohort of advanced undergraduate students to assess general usability and clarity, and a smaller group of domain experts for in-depth validation.

(1) *Student Crowdworkers:* Feedback was gathered from 270 third-year B.Tech. Computer Science and Engineering students at the Indian Institute of Information Technology Vadodara, India, as part of their Software Engineering course evaluation.
   - *Selection Criteria and Relevant Experience:* This cohort was specifically chosen because their coursework provided them with relevant foundational knowledge. All participating students had:
     – Successfully completed prerequisite courses in software engineering fundamentals.
     – Practical experience in creating multiple (over five) IEEE-standard SRS documents for different projects within the preceding six months.
     – Ongoing experience working on team-based capstone projects that required detailed requirements specification and analysis.
     – Regular exposure to requirements validation through weekly viva sessions with instructors specializing in Requirements Engineering.
   - *Training and Evaluation Protocol:* Before the evaluation, students were provided with a training session that included:
     – An overview of the core principles of GDPR and the Data Act (relevant documents were also provided as reference during the task).
     – A detailed explanation of the task, including how to interpret the output of the framework and the meaning of the four feedback options.

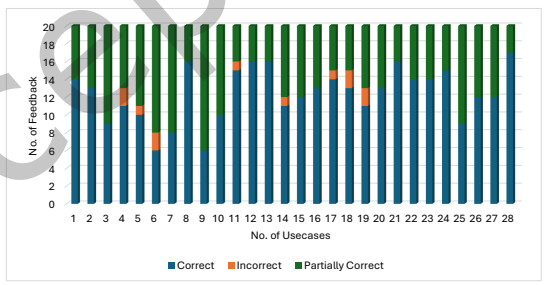
- A walk-through of two example use cases to ensure a consistent understanding of the evaluation criteria.
- In order to ensure sincerity and mitigate internal bias, feedback was collected as a graded component of their course, and students were informed that their justifications for a randomly selected subset of their responses would be manually reviewed for thoughtfulness and rationale.

Although not legal experts, this cohort represents a key target audience for such a tool: software developers and requirements engineers who need assistance in interpreting compliance requirements. Their feedback is therefore valuable for assessing the practical utility and clarity of the framework's output.

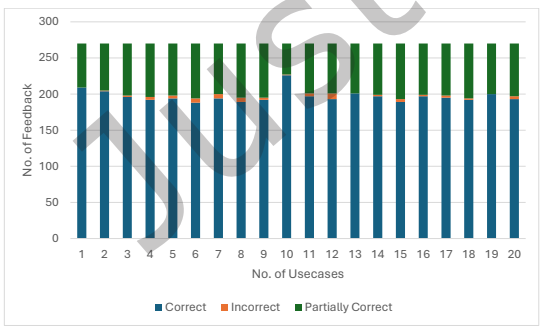
(2) *Domain Experts:* In addition to undergraduate evaluations, 20 software domain experts with more than ten years of experience in data protection and regulatory compliance evaluated the compliance check capabilities of the framework. These experts, well-versed in GDPR and the Data Act, evaluated a subset of system-generated compliance reports, focusing on the accuracy of identified non-compliant requirements, relevance of suggested modifications, and overall analysis quality. Their feedback provided valuable industry insights, complementing student evaluations and providing a comprehensive assessment of the practical value of the framework.



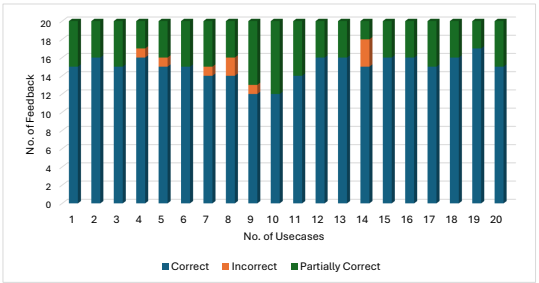
(a) Student feedback on requirements compliance checking against GDPR



(b) Expert feedback on requirements compliance checking against GDPR



(c) Student feedback on requirements compliance checking against Data Act



(d) Expert feedback on requirements compliance checking against Data Act

Fig. 8. Crowd-workers Feedback Analysis for Requirement Modification.

6.2.3 The Results. The evaluation approach provides a comprehensive perspective on the framework's efficacy, combining user feedback with expert validation. Figure 8 presents the analysis of the crowdsourced experiment for requirement modification. For the sake of brevity, the bar charts related to the analysis of *incompleteness* detection of the specification document with respect to the regulatory policies are provided in a supplementary document available online[7].

(a) Student Feedback Analysis: The analysis presented in Figures 8a and 8c reveal student perceptions of GDPR and Data Act compliance, respectively. Figure 8a shows that on average, 73% of the students view the modifications as GDPR compliant, 23% as partially compliant, and 4% as non-compliant. Similar trends are observed for the Data Act (Figure 8c), with 73% finding the modifications compliant, 26% partially compliant, and 1% non-compliant.

(b) Expert Feedback Analysis: Expert analysis for GDPR and Data Act compliance check is presented in Figures 8b and 8d respectively. The result shows that 61% of the experts, on average in all use cases, considered the suggested modifications to be correct, while 36% viewed them partially correct and 2% incorrect. This indicates a slightly lower level of agreement compared to student feedback, which could reflect the subtleties and complexities of legal interpretation. The results for the Data Act show a higher level of agreement between student feedback and expert opinion. For the Data Act compliance check, the modifications were deemed correct by 75% of the experts, partially correct by 23%, and incorrect by 2%.

The qualitative feedback collected during the evaluation indicates that the users found the framework's explanations and justifications for compliance decisions to be clear and intelligible. This positive feedback supports the effectiveness of our multi-agent approach and XAI components in addressing **RQ2**, demonstrating the framework's ability to provide transparent and reliable compliance assessments.

6.3 Case Studies

This section applies the framework to the software requirements specifications (SRS) of 20 projects, including 11 from the PURE dataset and 9 from Masters students at Ca' Foscari University, Venice, Italy. The evaluation presented in Table 6 is structured into two distinct iterations to demonstrate the automated refinement capability of the framework. It is important to clarify the mechanism distinguishing these two stages -

- *First Iteration:* The results labeled "First Iteration" represent the initial independent compliance checks performed by the two *CC_Agents* (powered by Command R+ and Mixtral 8x7B, respectively). This shows the baseline performance of each model and the common non-compliant requirements they identify without any collaboration.
- *Second Iteration (Automated Refinement):* The "Second Iteration" results showcase the impact of our automated, multi-agent communication loop. In this stage, the *RA_Agent* systematically compares the reports of the first iteration. It identifies discrepancies—specifically, non-compliant requirements that were missed by one agent but correctly identified by the other. The *RA_Agent* then generates a targeted prompt instructing the agent that missed the issue to re-evaluate the specific requirement, providing the rationale from the other agent as additional context. The second iteration reports the updated results after this agent-driven re-assessment.

It is important to note that for this specific case study evaluation, the Human-in-the-Loop (HITL) module was intentionally not invoked. The primary goal of this experiment was to isolate and measure the effectiveness of the automated multi-agent communication and refinement loop (i.e., the interaction between the *CC_Agents* and the *RA_Agent*). The role of the HITL system, as described above, is to provide the final layer of expert validation and continuous learning after this automated refinement has occurred. This controlled experimental setup allows us to distinctly evaluate the contribution of the agent-to-agent collaboration in correcting initial errors, such as the false negatives observed in the first iteration.

[7]https://doi.org/10.5281/zenodo.16176940

Table 6. Evaluation of GDPR and Data Act in non-compliant requirements detection across diverse projects

Usecase	First Iteration						Second Iteration					
	# non-compliant requirements detected by Command R+		# non-compliant requirements detected by Mixtral8x7B		# Common non-compliant requirements		# non-compliant requirements detected by Command R+		# non-compliant requirements detected by Mixtral8x7B		# Common non-compliant requirements	
	GDPR	Data Act	GDPR	Data Act.	GDPR	Data Act.	GDPR	Data Act	GDPR	Data Act.	GDPR	Data Act.
CCNTS	9	7	10	7	7	7	10	NA (7)	NA (10)	NA (7)	9	NA (7)
TachoNET	14	8	13	9	12	8	15	NA (8)	14	NA (9)	NA (10)	NA (8)
Inventory	8	7	8	5	5	4	10	NA (7)	9	6	8	6
Email	12	14	14	14	10	12	14	NA (14)	15	15	14	13
Home	9	7	7	7	6	6	NA (9)	8	8	8	7	8
e-procurement	16	11	15	10	14	10	17	10	15	NA (10)	15	NA (10)
znix	10	8	10	8	7	7	9	9	8	8	8	8
Stewards	12	6	10	6	9	6	13	NA (6)	12	NA (6)	12	NA (6)
Video Search	18	12	21	12	17	10	NA (18)	14	22	13	18	13
Virtual Education	9	9	10	9	9	9	9	NA (9)	10	NA (9)	9	NA (9)
Online Streaming	21	16	20	15	19	14	NA (21)	17	NA (20)	16	NA (19)	16
Population Management	6	9	6	10	5	9	7	NA (9)	NA (6)	NA (10)	6	NA (9)
Trek	3	6	3	5	3	5	NA (3)	NA (6)	NA (3)	6	3	6
FaceRecSpotify	8	9	10	10	8	9	9	NA (9)	NA (10)	NA (10)	9	NA (9)
Forum	6	8	6	8	6	7	NA (6)	NA (8)	NA (6)	9	6	NA (8)
Public Transport	4	3	3	3	2	3	NA (4)	NA (3)	4	NA (3)	3	NA (3)
Rent Service	9	10	8	9	8	9	NA (9)	NA (10)	9	10	9	10
Service Provider	6	12	8	11	5	10	8	13	9	12	8	12
NeighborGood	12	8	11	10	10	9	NA (12)	10	10	11	10	10
Journal Portal management	5	7	6	6	4	6	6	NA (7)	7	7	6	7

Table 6 presents the evaluation results of our framework in various projects, comparing the performance of the Command R+ and Mixtral 8x7B models across two iterations. The first iteration shows the initial, independent analysis by each agent, while the second iteration shows the improved results after the *RA_Agent* facilitates a re-evaluation of discrepancies. Instances where no further improvement was needed are marked as NA, with the first-iteration result shown in parentheses. A key challenge observed during the evaluation was the occurrence of false positives, where models erroneously identified a compliant requirement as non-compliant. These instances are highlighted in red in Table 6. For example, in the initial check of the 'znix' project, both models detected false positives, overestimating non-compliant requirements. Although the iterative process of the system helps correct some errors, as seen in the 'NeighborGood' project, where an initial false positive was resolved, some persisted. The 'Video Search' and 'Online Streaming' projects, for instance, retained false positives even after the second iteration. Specifically, the final results for the 'Video Search' project include 1 false positive from Command R+ and 4 from Mixtral 8x7B.

Note. The presence of these false positives, even in a sophisticated multi-agent system, underscores the critical need for the Human-in-the-Loop (HITL) validation stage. The automated framework is highly effective in identifying a comprehensive set of potential issues, but human expertise is indispensable to adjudicate ambiguous cases and filter out these final errors.

6.4 Comparative Analysis with a State-of-the-Art Monolithic LLM

To further validate the effectiveness of our framework's design, we conducted a comparative analysis against a leading, high-capacity monolithic LLM. For this experiment, we replaced our entire multi-agent compliance

Table 7. Comparative evaluation of non-compliant requirements detection: Our Framework vs. Monolithic Gemini 2.5 Pro

Usecase	GDPR		Data Act.	
	Our Approach	Gemini 2.5 Pro	Our Approach	Gemini 2.5 Pro
CCNTS	11	10	7	7
TachoNET	19	17	9	9
Inventory	11	11	7	8
Email	15	15	16	14
Home	10	10	8	12
e-procurement	17	15	10	10
znix	8	10	9	11
Stewards	13	13	6	7
Video Search	17	17	14	13
Virtual Education	10	10	9	9
Online Streaming	20	18	17	15
Population MGMT	7	9	9	9
Trek	3	4	6	7
FaceRecSpotify	10	11	10	12
Forum	6	6	9	9
Public Transport	5	5	3	4
Rent Service	9	10	10	10
Service Provider	9	9	13	13
NeighborGood	12	12	11	10
Journal Portal Mgmt	7	7	7	7

checking pipeline with a single, comprehensive prompt to Google's Gemini 2.5 Pro model, which is renowned for its long-context reasoning capabilities and, at the time of this study, ranked among the top-performing models on several LLM benchmarks, including MMLU-Pro, GPQA, MRCRv2, GRIND, and AIME [13]. The model was provided with the same SRS document and regulatory policy and tasked with identifying all non-compliant requirements in a single pass. This setup is designed to evaluate the performance of our structured, multi-agent RAG framework against a powerful, standalone LLM approach. The results of this comparison are presented in Table 7.

The analysis reveals two key findings. First, our multi-agent framework, which benefits from iterative refinement by the RA_Agent, correctly identifies more non-compliant requirements than the monolithic Gemini approach in 4 of the 20 use cases for GDPR compliance. For the Data Act, performance is more comparable, with Gemini. This demonstrates that our framework, utilizing smaller, more resource-efficient models like Mixtral and Command R+, can achieve on-par or even superior performance compared to a larger, state-of-the-art model. Second, this experiment highlights the limitations of relying on a single LLM call, even with a powerful model. The Gemini approach was also susceptible to false positives—incorrectly flagging compliant requirements as non-compliant—which are highlighted in red in the table. For instance, in the 'Home' and 'znix', 'faceRecSpotify' and 'Rent Service' use cases, it made several such errors. Although our approach is not immune to errors, the multi-agent validation and iterative refinement process appear to mitigate these issues more effectively.

This suggests that the architectural design of our framework, which combines specialized agents, retrieval enhancement, and iterative refinement, is a more critical factor for success than the raw power of a single underlying LLM. Our approach provides a more robust, reliable, and efficient solution for the nuanced task of compliance checking. The occurrence of errors in both systems further underscores the need for the Human-in-the-Loop (HITL) component for final validation. The evaluation results presented in Table 6 and Table 7, which

demonstrate the ability of the framework to detect non-compliant requirements across various software projects (e-commerce, education, etc.), provide evidence supporting the effectiveness of LLMs in addressing **RQ1**. The varying contexts provided by these datasets, representing different software domains and purposes, highlight the capacity of LLMs to incorporate contextual information during compliance analysis.

7 Discussion

This section revisits the research questions with detailed responses and discussion.

RQ1: *Can we use LLMs to analyze software requirements and check their compliance with regulations while considering the specific context in which the software will operate?*

Our framework shows that LLMs can effectively assess software requirements for compliance with regulations by incorporating the specific operational context of the software. Through strategic prompt engineering and structured methods like CoT and Query Decomposition, LLMs move beyond keyword matching to interpret the subtleties of requirements within specific software domains. This contextual integration enables LLMs to accurately identify non-compliant requirements, retrieve relevant compliance criteria, and suggest modifications for regulatory alignment. Enhanced with a retrieval-augmented knowledge base, the framework supports precise compliance evaluations, delivering detailed, step-by-step reasoning and actionable insights tailored to the software's operational context.

RQ2: *How can we design a transparent and reliable framework for LLM-based compliance checking, addressing the complexities of legal interpretation and the "black-box" nature of LLMs?*

Our framework addresses the "black-box" challenge through a multi-faceted approach centered on transparency, auditability, and reliability.

First, the multi-agent architecture (using Mixtral and Command R+) inherently enhances reliability. By distributing the compliance assessment, it reduces the reliance on the potential biases or weaknesses of any single model and enables automated cross-validation when the *RA_Agent* detects discrepancies between the agent outputs.

Second, the reliability of the framework is reinforced by its architectural transparency, which is very different from how unclear a single, large AI model can be. This transparency is powered by our XAI capabilities. When an error occurs, our framework functions as a "glass box", not a "black box". XAI features, such as Retrieval Path Explanation and Rationale Generation, provide a clear audit trail. This allows developers and legal experts to systematically trace the decision-making process and pinpoint the root cause of a failure—whether it was faulty retrieval, a flawed interpretation by a *CC_Agent*, or an error in the analysis of the *RA_Agent*. This level of debugging and auditable reasoning is essential for building trust in a domain where legal liability is paramount.

Third, the Human-in-the-Loop (HITL) mechanism provides the ultimate layer of expert validation. It allows legal professionals to review, correct, and refine the AI's outputs, ensuring that the final judgments of the system are aligned with complex human expertise and that the entire process remains accountable.

This architectural approach is especially important when dealing with the inherent ambiguity of regulatory texts. Terms such as "legitimate interests", "strictly necessary" often resist simple, binary interpretation. Instead of relying on a single opaque decision, our framework uses its multi-agent design to expose this ambiguity through agent disagreement. Because the heterogeneous agents (*CC_Agent1* and *CC_Agent2*) are trained on different legal and technical sources, they frequently reach different conclusions when interpreting vaguely worded GDPR clauses. The The *RA_Agent* identifies these discrepancies not merely as errors, but as high-confidence signals of underlying ambiguity. Additionally, the CoT prompting forces the agents to externalize their reasoning, often explicitly stating the assumptions made behind any decisions. By surfacing these disagreements and assumptions, the framework avoids silently "solving" the ambiguity on its own (which could result in hallucinated certainty).

Instead, it clearly flags these difficult interpretive issues and passes them to a Human-in-the-Loop, who can make a documented, expert decision that is appropriate for the specific GDPR context.

RQ3: *How can we effectively evaluate the efficiency of an LLM-based compliance checker using appropriate metrics?*

Evaluating the efficiency of our LLM-based compliance checker for complex tasks such as GDPR compliance requires moving beyond standard metrics like BLEU or ROUGE, which fall short of subtle legal interpretation. To address RQ3, we combine quantitative and qualitative approaches. Quantitatively, we applied the RAGAS framework with metrics tailored for Retrieval-Augmented Generation (RAG) systems. Metrics such as context recall, precision, faithfulness and answer relevancy measure the framework's capability to retrieve, process, and produce relevant, accurate compliance assessments. Qualitatively, we gather user feedback focused on the clarity, usability, and actionability of the compliance outputs. This feedback from requirements engineers and domain experts ensures that the evaluation aligns with real-world expectations, adding practical insight to the quantitative results.

RQ4: *Can the proposed solution be adapted to different regulatory landscapes and accommodate evolving compliance standards, demonstrating its scalability and adaptability?*

The modular design of our framework, with its separate knowledge base, is key to addressing RQ4. By simply updating the knowledge base with new regulations or revised standards, the system can be adapted to different compliance contexts without altering the core architecture. Although our current evaluation of the framework focuses on GDPR, preliminary experiments applying the framework to the Data Act suggest that it can be easily adapted to other regulatory environments, including the Federal Information Security Management Act (FISMA) and the Payment Card Industry Data Security Standard (PCI DSS). This inherent flexibility demonstrates the framework's scalability and its potential for generalization beyond GDPR, accommodating evolving legal landscapes. Future work will explore the framework's performance with other regulations (e.g., PCI DSS, FISMA) to assess its adaptability more comprehensively.

8 Threats to Validity

The main threats to the validity of the framework are summarized below.

8.1 Internal Validity

The effectiveness of the framework relies on the design and quality of prompts used for compliance checking and multi-agent communication. Suboptimal prompt design, or lack of systematic tuning, can lead to inaccurate analysis. Although the initial prompt design was guided by task logic and iteratively refined through HITL feedback, and while CoT was chosen for its established benefits in complex reasoning, no systematic comparison was conducted against alternative prompting strategies or a formal ablation study on the specific impact of CoT within our framework. More research is needed to formalize prompt engineering techniques, explore automated prompt generation or tuning for this domain, and comparatively evaluate different PE methodologies.

A separate concern is the risk of false negatives during the intermediate stages of compliance assessment. Although not prominent in the final results, some non-compliant requirements may have been initially overlooked due to the complexity of regulatory analysis. The iterative multi-agent process helps correct these, but future work should focus on strengthening early-stage detection to further minimize this risk.

A potential threat to internal validity concerns the specific implementation choices and hyperparameters. Although Section 4.5 details our selected settings for reproducibility, a comprehensive study of hyperparameter optimization or ablation — such as systematically varying embedding models, vector databases, or retrieval k values — was beyond the scope of this initial work. Such exploration may lead to further performance gains and represents a clear direction for future research.

8.2 External Validity

Further investigations are required to check the generic nature of the proposed framework for other regulations beyond the GDPR and the Data Act. Adapting to different regulatory landscapes may require adjustments in prompt engineering and knowledge-base integration. The case studies used a small selection of software projects, mainly from the PURE dataset and student projects. The performance of the framework on larger and more diverse datasets needs further exploration.

8.3 Construct Validity

A potential threat to construct validity lies in our evaluation methodology, which combines feedback from two distinct non-legal groups: undergraduate students and domain experts in software engineering. This was a deliberate methodological choice to evaluate distinct facets of the framework. The undergraduate student crowdworkers were not selected to act as legal experts, and their feedback should not be interpreted as a validation of the legal accuracy of the framework. Instead, they represent a proxy for the primary target audience of this tool - software developers and requirements engineers. As detailed in Section 6.2.2, they were selected for their significant experience in requirements specification and received task-specific training. Their feedback is therefore invaluable for assessing the usability of the framework, the clarity of its generated explanations, and its perceived utility in a software development context.

The evaluation by 20 domain experts, while not legal professionals, provides a complementary perspective grounded in the practical challenges of implementing regulatory compliance. However, we acknowledge that this study lacks an evaluation by certified legal experts. Assessing the nuanced correctness of the analysis of the complex legal issues requires deep legal training. The clear differences observed between student and expert evaluations (as shown in Figure 8), where experts identified more subtle legal gaps, underscore this point. Consequently, our results should be interpreted as strong indicators of practical utility and developer acceptance, with the final legal soundness requiring further validation. A rigorous evaluation involving legal professionals represents a critical direction for future work.

9 Conclusion

This paper presents a novel multi-agent framework for automating software requirements compliance checking. Combining LLMs, RAG, strategic prompt engineering, and multi-agent communication, the framework provides a scalable, efficient, and accurate solution for detecting non-compliance in software requirements. It explores effective LLM use for legal and technical analysis, designs a transparent and reliable compliance system, develops robust evaluation strategies, and ensures adaptability to various regulatory contexts.

Our evaluation, using RAGAS metrics and qualitative feedback from crowdsourced participants and domain experts, confirms the effectiveness of the framework in identifying non-compliant requirements and proposing modifications. Case studies demonstrate its practical applicability across diverse projects. The framework leverages XAI techniques, including rationale generation, retrieval path visualization, and transparent processes, to address LLM "black box" issues and build user trust. A Human-in-the-Loop mechanism ensures expert validation and continuous system refinement.

While showing promising results, our framework has limitations, including LLM biases and limited generalizability. Differences between expert and crowdworker assessments underscore the complexity of legal interpretation, warranting further research. To address this, future work will move towards advanced Context Engineering to assist in disambiguating vague regulatory terms. We aim to architect a richer information environment by - (1) expanding the RAG pipeline with interpretive legal guidance (e.g., official EDPB guidelines and case law precedents) to ground analysis; (2) developing metrics for ambiguity quantification based on agent disagreement. Furthermore, we plan to enhance XAI techniques to specifically address false negatives, expand

the knowledge base to support diverse global regulations, and evaluate the framework on larger, industrial-scale projects to deliver trustworthy automated solutions.

Acknowledgments

Part of this research was funded by the Luxembourg National Research Fund (FNR), grant reference NCER22/IS/16570468/NCER-FT. For the purpose of open access, and in fulfillment of the obligations arising from the grant agreement, the authors have applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission.

Work partially supported by SERICS (PE00000014 - CUP H73C2200089001) under the NRRP MUR program funded by the EU - NGEU, and iNEST- Interconnected NordEst Innovation Ecosystem funded by PNRR (Mission 4.2, Investment 49 1.5) NextGeneration EU (ECS_00000043 – CUP H43C22000540006).

References

[1] Abdulrahman Alhazmi and Nalin AG Arachchilage. 2021. A serious game design framework for software developers to put GDPR into practice. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*. 1–6.

[2] Orlando Amaral, Sallam Abualhaija, Mehrdad Sabetzadeh, and Lionel Briand. 2021. A model-based conceptualization of requirements for compliance checking of data processing against GDPR. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*. IEEE, 16–20.

[3] Orlando Amaral, Sallam Abualhaija, Damiano Torre, Mehrdad Sabetzadeh, and Lionel C Briand. 2021. AI-enabled automation for completeness checking of privacy policies. *IEEE Transactions on Software Engineering* 48, 11 (2021), 4647–4674.

[4] Orlando Amaral, Muhammad Ilyas Azeem, Sallam Abualhaija, and Lionel C Briand. 2023. Nlp-based automated compliance checking of data processing agreements against gdpr. *IEEE Transactions on Software Engineering* (2023).

[5] AI Anthropic. 2024. Introducing the Next Generation of Claude.

[6] Chetan Arora, Tomas Herda, and Verena Homm. 2024. Generating Test Scenarios from NL Requirements using Retrieval-Augmented LLMs: An Industrial Study. *arXiv preprint arXiv:2404.12772* (2024).

[7] Muhammad Ilyas Azeem and Sallam Abualhaija. 2023. A Multi-solution Study on GDPR AI-enabled Completeness Checking of DPAs. *arXiv preprint arXiv:2311.13881* (2023).

[8] Parishad BehnamGhader, Santiago Miret, and Siva Reddy. 2023. Can Retriever-Augmented Language Models Reason? The Blame Game Between the Retriever and the Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 15492–15509. doi:10.18653/v1/2023.findings-emnlp.1036

[9] Armin Berger, Lars Hillebrand, David Leonhard, Tobias Deußer, Thiago Bell Felix De Oliveira, Tim Dilmaghani, Mohamed Khaled, Bernd Kliem, Rudiger Loitz, Christian Bauckhage, et al. 2023. Towards Automated Regulatory Compliance Verification in Financial Auditing with Large Language Models. In *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 4626–4635.

[10] Piero A Bonatti, Sabrina Kirrane, Iliana M Petrova, and Luigi Sauro. 2020. Machine understandable policies and GDPR compliance checking. *KI-Künstliche Intelligenz* 34 (2020), 303–315.

[11] Travis D Breaux, Matthew W Vail, and Annie I Anton. 2006. Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. In *14th IEEE International Requirements Engineering Conference (RE'06)*. IEEE, 49–58.

[12] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216* (2024).

[13] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv preprint arXiv:2507.06261* (2025).

[14] Ao Ding, Gaolei Li, Xiaoyu Yi, Xi Lin, Jianhua Li, and Chaofeng Zhang. 2024. Generative artificial intelligence for software security analysis: Fundamentals, applications, and challenges. *IEEE Software* (2024).

[15] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217* (2023).

[16] Ming Fan, Le Yu, Sen Chen, Hao Zhou, Xiapu Luo, Shuyue Li, Yang Liu, Jun Liu, and Ting Liu. 2020. An empirical evaluation of GDPR compliance violations in Android mHealth apps. In *2020 IEEE 31st international symposium on software reliability engineering (ISSRE)*. IEEE, 253–264.

[17] Alessandro Fantechi, Stefania Gnesi, Lucia Passaro, and Laura Semini. 2023. Inconsistency Detection in Natural Language Requirements using ChatGPT: a Preliminary Evaluation. In *2023 IEEE 31st International Requirements Engineering Conference (RE)*. IEEE, 335–340.

[18] Alessio Ferrari, Giorgio Oronzo Spagnolo, and Stefania Gnesi. 2017. Pure: A dataset of public requirements documents. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*. IEEE, 502–505.
[19] Shengbo Guo and Scott Sanner. 2010. Probabilistic latent maximal marginal relevance. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. 833–834.
[20] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088* (2024).
[21] Nadzeya Kiyavitskaya, Nicola Zeni, Travis D Breaux, Annie I Antón, James R Cordy, Luisa Mich, and John Mylopoulos. 2008. Automating the extraction of rights and obligations for regulatory compliance. In *Conceptual Modeling-ER 2008: 27th International Conference on Conceptual Modeling, Barcelona, Spain, October 20-24, 2008. Proceedings 27*. Springer, 154–168.
[22] Robert Lakatos, Peter Pollner, Andras Hajdu, and Tamas Joo. 2024. Investigating the performance of Retrieval-Augmented Generation and fine-tuning for the development of AI-driven knowledge-based systems. *arXiv preprint arXiv:2403.09727* (2024).
[23] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.
[24] Xueguang Ma, Kai Sun, Ronak Pradeep, and Jimmy Lin. 2021. A replication study of dense passage retriever. *arXiv preprint arXiv:2104.05740* (2021).
[25] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. MTEB: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316* (2022).
[26] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).
[27] Monica Palmirani, Guido Governatori, et al. 2018. Modelling Legal Knowledge for GDPR Compliance Checking.. In *JURIX*, Vol. 313. 101–110.
[28] Ildikó Pilán, Benet Manzanares-Salor, David Sánchez, and Pierre Lison. 2025. Truthful text sanitization guided by inference attacks. *Applied Soft Computing* (2025), 114013.
[29] Catherine Sai, Shazia Sadiq, Lei Han, Gianluca Demartini, and Stefanie Rinderle-Ma. 2024. Identification of Regulatory Requirements Relevant to Business Processes: A Comparative Study on Generative AI, Embedding-based Ranking, Crowd and Expert-driven Methods. *arXiv preprint arXiv:2401.02986* (2024).
[30] Aamir Shakir, Darius Koenig, Julius Lipp, and Sean Lee. 2024. *Boost Your Search With The Crispy Mixedbread Rerank Models.* https://www.mixedbread.com/blog/mxbai-rerank-v1
[31] Supreeth Shastri, Melissa Wasserman, and Vijay Chidambaram. 2019. GDPR anti-patterns: How design and operation of modern cloud-scale systems conflict with GDPR. *arXiv preprint arXiv:1911.00498* (2019).
[32] Amin Sleimi, Nicolas Sannier, Mehrdad Sabetzadeh, Lionel Briand, and John Dann. 2018. Automated extraction of semantic legal metadata using natural language processing. In *2018 IEEE 26th International Requirements Engineering Conference (RE)*. IEEE, 124–135.
[33] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori Hashimoto. 2023. *Stanford Alpaca: An Instruction-following LLaMA model.* https://crfm.stanford.edu/2023/03/13/alpaca.html
[34] Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2020. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. *arXiv preprint arXiv:2010.08240* (2020).
[35] Tobias Urban, Dennis Tatang, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. 2020. Measuring the impact of the GDPR on data sharing in ad networks. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*. 222–235.
[36] Nicola Zeni, Elias A Seid, Priscila Engiel, Silvia Ingolfo, and John Mylopoulos. 2016. Building large models of law with NómosT. In *Conceptual Modeling: 35th International Conference, ER 2016, Gifu, Japan, November 14-17, 2016, Proceedings 35*. Springer, 233–247.