PAKTON: A Multi-Agent Framework for Question Answering in Long Legal Agreements

Petros Raptopoulos[1], **Giorgos Filandrianos**[1,2], **Maria Lymperaiou**[1], **Giorgos Stamou**[1]
[1]School of Electrical and Computer Engineering, AILS Laboratory,
National Technical University of Athens, Greece
[2]Instituto de Telecomunicações, Portugal
petrosrapto@gmail.com, {geofila, marialymp}@islab.ntua.gr, gstam@cs.ntua.gr

Abstract

Contract review is a complex and time-intensive task that typically demands specialized legal expertise, rendering it largely inaccessible to non-experts. Moreover, legal interpretation is rarely straightforward—ambiguity is pervasive, and judgments often hinge on subjective assessments. Compounding these challenges, contracts are usually confidential, restricting their use with proprietary models and necessitating reliance on open-source alternatives. To address these challenges, we introduce PAKTON: a fully open-source, end-to-end, multi-agent framework with plug-and-play capabilities. PAKTON is designed to handle the complexities of contract analysis through collaborative agent workflows and a novel multi-stage retrieval-augmented generation (RAG) component, enabling automated legal document review that is more accessible, adaptable, and privacy-preserving. Experiments demonstrate that PAKTON outperforms both general-purpose and pretrained models in predictive accuracy, retrieval performance, explainability, completeness, and grounded justifications as evaluated through a human study and validated with automated metrics.[1]

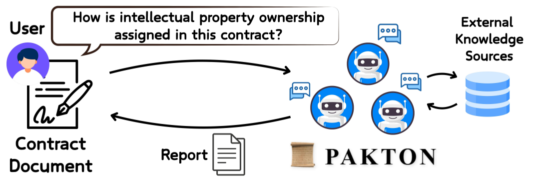


Figure 1: PAKTON user flow: legal query submission followed by comprehensive report generation.

1 Introduction

Contracts are among the most foundational legal documents, governing a wide range of personal, commercial, and governmental relationships. Yet, analyzing contracts remains a complex and time-consuming task that typically requires legal expertise. As a result, contract review is often inaccessible to the general public and remains demanding even for professionals. Research by World Commerce & Contracting (WorldCC) reveals that organizations lose an average of 9.2% of their annual revenue due to contract mismanagement, with that figure rising to 15% for larger enterprises (World Commerce & Contracting, 2020). In parallel, the Institute for Supply Management (ISM) has reported that a typical Fortune 1000 company manages between 20,000 and 40,000 active contracts at any given time (Institute for Supply Management), while even simple agreements can take over a week to approve. These challenges highlight the growing need for better support in understanding contracts and reducing the time required for manual review.

Recent advancements in large language models (LLMs) show promise in natural language understanding, question answering, and document summarization (Achiam et al., 2023; Anthropic, 2025). While LLMs are powerful, their application in specialized domains such as legal contract analysis presents unique challenges (Frei, 2016; Eisenberg, 2022; Kant et al., 2025), including justifying decisions with clear reasoning and referencing both the contract and relevant external sources (Zhang et al., 2025). Contract documents also exhibit several peculiarities that require specialized handling. They often contain complex legal terminology, calling for domain-specific language understanding. Overlapping or even contradictory clauses appear frequently (Marques et al., 2024; Ichida and Meneguzzi, 2021; Aires et al., 2019), requiring robust clause retrieval and conflict resolution (Aires et al., 2019; Zhou et al., 2024). Similarly, exceptions and references to different document parts also request robust retrieval mechanisms. Ambiguous phrasing and multiple interpretations are common, making careful contextual analysis essential. Additionally, legal differences across ju-

[1]Code can be found at github.com/petrosrapto/PAKTON.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 7948–7984
November 4-9, 2025 ©2025 Association for Computational Linguistics

risdictions necessitate consultation with external legal databases to maintain precision and relevance.

Retrieval-Augmented Generation (RAG) (Gao et al., 2023; Fan et al., 2024; Wang et al., 2025; Gao et al., 2024) directly addresses these limitations by integrating targeted retrieval of internal and external documents into the LLM generation process, grounding outputs in domain-specific, verifiable evidence. This approach enhances not only factual accuracy but also transparency and explainability, which constitute critical attributes for legal contract analysis. Moreover, RAG plays a pivotal role in the legal domain, where the inability to retrieve relevant spans can compromise reasoning and produce unsupported conclusions (Pipitone and Alami, 2024). Furthermore, given that legal contracts frequently contain sensitive or confidential information, proprietary models are often unsuitable, necessitating open-source alternatives capable of operating under limited computational resources. Consequently, the development of efficient mechanisms for contract analysis becomes imperative.

Within this context, we introduce PAKTON[2] (Figure 1), a multi-agent framework designed to analyze contract documents and provide explainable, legally grounded answers to user queries along with a comprehensive report. Inspired by Shao et al. (2024), PAKTON is composed of three specialized collaborative agents: (1) the *Archivist*, which interacts with the user and manages structured document input; (2) the *Researcher*, which retrieves relevant internal and external information using hybrid and graph-aware retrieval; (3) the *Interrogator*, which engages in multi-step reasoning to iteratively refine the report. Each agent is dedicated to separate core legal aspects, ensuring optimal attribution of tasks. At the same time, the tri-agent structure facilitates implementation using different backbone LLMs, thus ultimately offering a plug-and-play, highly customizable solution. PAKTON departs from black-box models by prioritizing transparency, progressive refinement, and grounded justifications. It generates structured legal reports with topic summaries, legal reasoning, key findings, and precise citations to contract clauses and external sources, while explicitly flagging knowledge gaps to avoid unsupported claims. All components are open source, lightweight, and support on-premise deployment. We benchmark PAKTON on five contract analysis tasks using ten criteria combining automated and human metrics, and find that it substantially surpasses general-purpose LLMs in accuracy and explainability.

[2] *PAKTON* comes from the ancient Greek word that means agreement or contract, related to the Latin "pactum".

2 Related Work

Recent advances in applying LLMs and RAG to legal tasks significantly boost contract analysis, legal QA, and document review (Shu et al., 2024; Lai et al., 2024). Domain-specific RAG frameworks, such as Legal Query RAG, improve accuracy and relevance by combining fine-tuned legal LLMs, evaluation agents, and recursive feedback, reducing hallucinations and enhancing responses to complex queries (Wahidur et al., 2025). Literature surveys highlight a sharp rise in research on LLM-driven contract review, legal research, and regulatory compliance, alongside increasing methodological sophistication and expansion into multilingual, cross-jurisdictional contexts (Siino et al., 2025).

Despite these advances, the community faces persistent challenges. A major issue is the tendency of LLMs to generate hallucinated or misleading responses, especially when lacking deep domain knowledge or when retrieval mechanisms fail to surface the most relevant legal context (Wahidur et al., 2025; Zhao et al., 2024b; Saha et al., 2024). One strategy that has been investigated to mitigate this issue is fine-tuning language models on dedicated legal corpora, with the aim of deepening their understanding of legal terminology, reasoning styles, and contextual subtleties (Colombo et al., 2024b,a; Huang et al., 2023). The evaluation of generated content remains difficult, as automated metrics often do not align with expert legal judgment, and human evaluation is costly and time-consuming (Wahidur et al., 2025; Ryu et al., 2023; Veturi et al., 2024). Data availability and quality are also significant hurdles, particularly for specialized legal domains or languages with limited resources (Ryu et al., 2023; Akarajaradwong et al., 2025). Furthermore, the complexity of legal language and reasoning, as well as the need for transparency and explainability in AI-generated outputs, present ongoing obstacles (Wahidur et al., 2025; Akarajaradwong et al., 2025; Bianchini et al., 2024). Approaches, such as integrating case-based reasoning with RAG (Wiratunga et al.), leveraging knowledge graphs (Bianchini et al., 2024), and developing new retrieval and evaluation strategies (Akarajaradwong et al., 2025; Saha et al., 2024; Ryu et al., 2023), are being explored to address these limitations.

Additionally, multi-agent frameworks have recently gained attention, enhancing the reasoning capabilities and reliability of legal AI systems (ShengbinYue et al., 2025; Sun et al., 2024; Zhao et al., 2024a; Liu et al., 2025). Inspired by prior work such as STORM (Shao et al., 2024), which introduces collaborative multi-turn interactions among diverse agents for content generation, and ChatLaw (Cui et al., 2023), which employs a knowledge graph-enhanced multi-agent design for legal assistance, we adopt a similar approach tailored specifically for contract analysis. PAKTON is designed to address the complex reasoning and rigorous transparency requirements inherent in legal tasks by orchestrating agents that iteratively interrogate, retrieve, and refine multi-source information.

3 PAKTON Framework

PAKTON aims to analyze user-provided contracts and generate query responses grounded in the contract's content while integrating pertinent external knowledge. A key design goal is to ensure transparent and traceable reasoning by referencing evidence spans from the contract and articulating the rationale behind each conclusion. The final output is delivered as a structured legal report.

To operationalize this functionality, we employ a tri-agent, model-independent architecture comprising the *Archivist*, *Interrogator*, and *Researcher*, each fulfilling a clear separation of concerns by assigning distinct roles regarding the retrieval, reasoning, and synthesis pipeline. This agentic structure mirrors how legal professionals work in practice, aligning the system's workflow with real-world roles for greater trust and interpretability, improving contract analysis organization. Each agent is implemented as a reasoning loop following the ReAct paradigm (Yao et al., 2022), combining reasoning and action steps via a backbone LLM. This design enables agents to reason over their current state, generate tool-specific actions and iteratively refine their decisions based on intermediate outcomes. An overview of PAKTON is illustrated in Figure 2. Further details are provided in App. A.

3.1 Archivist

The *Archivist* is responsible for gathering and organizing relevant user information. It plays a central role in the initial stages of the system by collecting and summarizing the user's query, any accompanying instructions, and contextual background. This information is then structured and passed to the *Interrogator* agent. The *Archivist* also manages storage and embedding of the user's contract document and actively engages in dialogue to resolve ambiguities or request missing user information, ultimately implementing three core functionalities:

Document Parsing. To enable practical deployment, the *Archivist* ingests diverse document formats within an integrated information pipeline. It natively parses text-based files, preserving structure such as headings, paragraphs, and numbered clauses. For PDFs, it applies OCR methods (Smith, 2007, 2013) with visual layout analysis to recover document hierarchies, including sections, tables, and multi-column formats. All extracted content is then filtered and normalized to ensure consistent, structured input for downstream processing.

Hierarchical Parsing. Legal documents typically feature complex structures, cross-references, and nuanced semantic variations that render flat or linear representations inadequate for accurate analysis. To overcome this, we generate a hierarchical tree that mirrors the contract's internal organization, introducing one core design novelty of PAKTON. The contract text is parsed into this tree structure by leveraging structural cues in conjunction with either semantic similarity measures based on BERT embeddings (Wang et al., 2020) or from LLMs.

The structural parsing process involves two main steps as shown in Figure 5: (1) identifying the distinct document sections, and (2) determining the hierarchical relationships between them. Examples of sections include titles, clauses, paragraphs, and enumerated list items. A section is considered the child of another if, structurally, it appears nested within the parent section based on indentation, numbering patterns, visual layout (in OCR-extracted documents), or semantic proximity.

Encode Document. The tree representation of the contract is used to create contextualized chunks that are then embedded for retrieval. Context-aware chunking has been shown to reduce failed retrievals by preserving semantic coherence across sections (Anthropic, 2024a). In particular, both overlapping (Wang et al., 2024a) and dynamic chunking (Duarte et al., 2024) have demonstrated gains in retrieval performance by maintaining relevant context around each chunk boundary. To capture different contextual information levels, we gen-

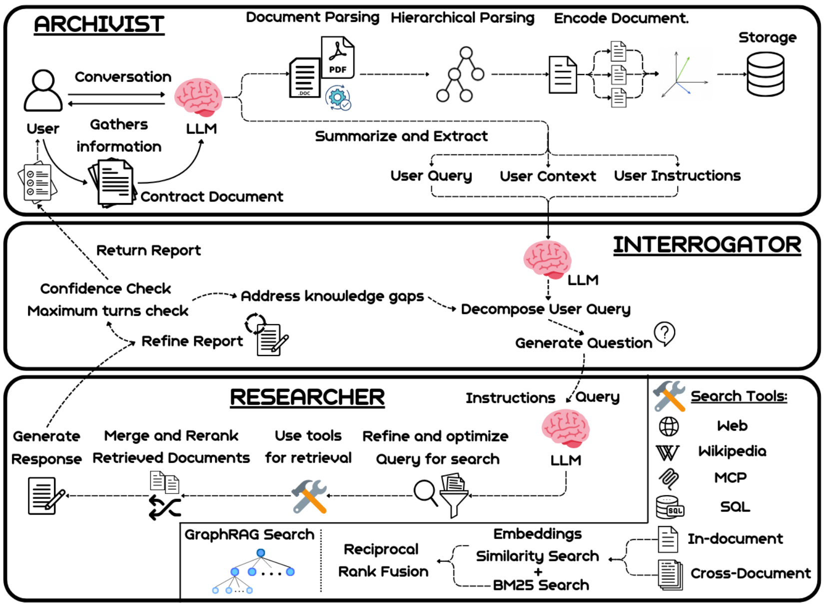


Figure 2: An overview of the proposed PAKTON framework and its internal components.

erate three chunk types for each tree node, which are later concatenated into a unified chunk set. Duplicate or highly similar chunks are filtered out to improve embedding diversity and avoid redundancy. The chunking strategies are as follows:

- **Node-level:** Encodes each node in isolation to preserve fine-grained semantics. This supports precise alignment with user queries targeting specific clauses or legal provisions and ensures that each atomic unit is independently retrievable.
- **Ancestor-aware:** Concatenates a node with its hierarchical ancestors, capturing inherited context from section headers, articles, and titles. This contextualization aids disambiguation and enhances understanding of a section's role within the broader contractual structure.
- **Descendant-aware:** Aggregates a node with its descendants to embed cohesive semantic units, such as clauses with their subclauses or enumerations. This is particularly beneficial for reasoning over compound or multi-part provisions, where meaning is distributed across nested content.

Each chunk is further enriched with metadata, including its structural location within the tree, document position, filename, and a contract-level summary. This information facilitates both cross-document retrieval by guiding vector store routing, as well as intra-document relevance ranking. By integrating multiple structural perspectives, our chunking framework improves the likelihood of retrieving relevant content across a spectrum of query granularities, from clause-specific lookups to section-level reasoning, advancing the encoding process in comparison to existing works.

3.2 Interrogator

The *Interrogator* is responsible for generating the final report presented to the user as the system's response, orchestrating a multi-step reasoning process that addresses queries with accuracy and confidence. It receives the user's query, contextual information, and instructions, and initiates an iterative interrogation of the *Researcher*. This process involves decomposing the original query and generating a series of questions aimed at refining and deepening the system's understanding. Once a response is received from the *Researcher* for the first question, the *Interrogator* generates a preliminary report, attempting to directly answer the user's query. This initial response also serves to identify potential knowledge gaps that require further research. Based on this initial exchange, the *Interrogator* formulates follow-up questions aimed at clarifying missing information in the preliminary answer. When generating each subsequent question, it considers the user query, context, instructions, a summary of previously asked questions, and the current draft of the report. Questions that are most likely to significantly improve the final response are prioritized. With each new response from the *Researcher*, the *Interrogator* incrementally refines the draft report. The process continues until one of two stopping conditions is met: 1) the *Interrogator* determines that the answer is sufficiently confident and complete, or 2) the maximum number of allowed interrogation turns (simulating retrieval depth), set by the user, has been reached.

The *Interrogator* ensures the structural integrity and completeness of the report, which must include: 1) Title and topic summary, 2) Legal reasoning and key findings, 3) Preliminary answer and suggested research directions, 4) Knowledge gaps and follow-up questions, 5) Cited sources and evidentiary support. Incorporating query-specific information e.g. title and summary into the final report not only enhances user readability but also improves model performance by encouraging query rephrasing at each refinement step, thus deepening task comprehension (Mekala et al., 2024). The *Interrogator*'s iterative architecture further facilitates knowledge gap identification, targeted information retrieval, and progressive query disambiguation. The inherent repetition in this process has been shown to enhance in-context learning and output accuracy (Xu et al., 2024). These design choices are integral to the overall effectiveness of PAKTON.

3.3 Researcher

The *Researcher* is responsible for retrieving relevant information to support the *Interrogator* in answering the user's query, equipped with multiple retrieval methods. Depending on the nature of the query and the accompanying instructions, the *Researcher* autonomously selects the most suitable retrieval method, or combination of methods. This selection process is driven by prompting an LLM to choose the optimal set of tools based on the query content and tool descriptions. Retrieval methods are categorized into the following:

- **In-document retrieval**: Leveraging the document chunks and embeddings provided by the *Archivist*, the *Researcher* retrieves spans to address the query. A hybrid retrieval approach is employed, combining BM25 (Robertson and Zaragoza, 2009), dense embeddings (Lewis et al., 2020), and Reciprocal Rank Fusion (RRF) (Cormack et al., 2009). This is further enhanced by LightRAG (Guo et al., 2024) (a lightweight version of GraphRAG (Edge et al., 2024)), improving entity- and relation-level matching within and across documents, and is particularly effective for answering global queries that require reasoning over multiple interrelated documents.
- **Cross-document retrieval**: This method retrieves relevant spans from other documents, which the *Interrogator* can leverage as exemplars in few-shot prompting.
- **Retrieval of external knowledge**: PAKTON provides retrieval tools including web search, Wikipedia, SQL databases, and external legal sources via the Model Context Protocol (MCP) (Anthropic, 2024b) to supplement model knowledge or provide real-time information. For fair evaluation, these retrieval capabilities were not used in our experiments.

To ensure both high recall and precision, the *Researcher* employs a two-step retrieval and reranking process. Initially, high-recall retrievers, such as the aforementioned ones, are used to collect a broad set of potentially relevant passages maximizing coverage. Subsequently, a cross-encoder model reranks the retrieved results by jointly encoding the query and each passage, allowing for more accurate semantic relevance estimation (Karpukhin et al., 2020). This reranking stage serves as a high-precision filter that refines the initial candidate set. Relevance scores are normalized using a sigmoid function, and passages exceeding a predefined similarity threshold are passed to the response generation module, which synthesizes the final answer.

4 Experiments and Results

Due to its model-independent design, PAKTON operates by accommodating different LLMs as agent instantiations. Specifically, our experimentation comprises Mistral (Jiang et al., 2023), Qwen (Qwen et al., 2025), Gemma (Team et al., 2025), Llama (Grattafiori et al., 2024), Claude (Anthropic, 2025), Deepseek (DeepSeek-AI et al., 2025) and GPT-4 (Achiam et al., 2023) model families.

PAKTON is evaluated using both quantitative and qualitative methods. Quantitatively, it is tested on contract analysis tasks, consistently outperforming all baselines in accuracy. Traditional ablation studies at the agent level are not feasible, as each agent plays an essential and interdependent role. Instead, we conduct targeted evaluations to isolate contributions. The retrieval capabilities of the *Archivist–Researcher* pair are measured across four benchmark datasets, where PAKTON achieves state-of-the-art results. The overall generation performance of the end-to-end system, including the *Interrogator*, is assessed quantitatively on a contract analysis task and qualitatively. The qualitative evaluation uses nine criteria aligned with PAKTON's goal of producing interpretable, actionable

Model	Method	Acc.	F1 [W]	F1 [E]	F1 [C]	F1 [N]
Saul7B	Inst. Tun.	0.4196	0.2900	0.0589	0.0680	0.5920
Saul54B	Inst. Tun.	0.7020	0.6792	0.7727	0.1729	0.7024
Mistral 7B	ZS	0.5364	0.5042	0.5279	0.0248	0.5951
	FS	0.5065	0.4702	0.6053	0.0082	0.4379
	FS+Spans	0.4940	0.4576	0.6085	0.0076	0.4053
	PAKTON	**0.7032**	**0.6789**	0.7782	0.2469	0.6828
Mixtral 8x7B	ZS	0.5423	0.5475	0.6445	0.4103	0.4770
	FS	0.6002	0.5804	0.6836	0.1931	0.5642
	FS+Spans	0.6150	0.6017	0.6901	0.1951	0.6060
	PAKTON	**0.7423**	**0.7429**	0.7864	0.6655	0.7187
Qwen 2.5 72B	ZS	0.7728	0.7699	0.8248	0.5776	0.7579
	FS	0.7351	0.7241	0.8094	0.4920	0.6892
	FS+Spans	0.7484	0.7432	0.8196	0.4378	0.7357
	PAKTON	**0.8192**	**0.8188**	0.8353	0.7737	0.8132
Gemma 3 27B	ZS	0.7886	0.7860	0.8316	0.6348	0.7739
	FS	0.7191	0.7049	0.7815	0.4608	0.6891
	FS+Spans	0.7720	0.7639	0.8287	0.4728	0.7662
	PAKTON	**<u>0.8287</u>**	**<u>0.8283</u>**	0.8487	0.7546	0.8255
Llama 3.3 70B	ZS	0.6767	0.6716	0.7366	0.5378	0.6346
	FS	0.6657	0.6565	0.7326	0.4431	0.6268
	FS+Spans	0.6915	0.6879	0.7382	0.4244	0.6982
	PAKTON	**0.8217**	**0.8207**	0.8422	0.7488	0.8165
Llama 3.1 70B	ZS	0.5811	0.5577	0.5216	0.3152	0.6555
	FS	0.5729	0.5506	0.5421	0.2381	0.6358
	FS+Spans	0.5538	0.5180	0.4471	0.3014	0.6468
	PAKTON	**0.7916**	**0.7903**	0.8097	0.6846	0.7960
Claude 3.5	ZS	0.7916	0.7977	0.8757	0.5722	0.7691
	FS	0.7778	0.7816	0.8588	0.5702	0.7505
	FS+Spans	0.7999	0.8034	0.8678	0.6046	0.7826
	PAKTON	**0.7990**	**0.8000**	0.8157	0.7046	0.8072
Claude 3.7	ZS	0.7704	0.7781	0.8633	0.5602	0.7398
	FS	0.7590	0.7602	0.8463	0.5607	0.7165
	FS+Spans	0.7724	0.7766	0.8538	0.5805	0.7417
	PAKTON	**0.8247**	**0.8254**	0.8386	0.7495	0.8304
Deep-seek V3	ZS	0.7886	0.7875	0.8487	0.6117	0.7648
	FS	0.7681	0.7607	0.8346	0.6104	0.7182
	FS+Spans	0.7743	0.7714	0.8377	0.5812	0.7465
	PAKTON	**0.8192**	**0.8200**	0.8315	0.7615	0.8224
GPT-4o	ZS	0.6121	0.6366	0.7490	0.4162	0.5698
	FS	0.6640	0.6789	0.7372	0.4734	0.6666
	FS+Spans	0.6482	0.6574	0.6664	0.4636	0.6950
	PAKTON	**0.7966**	**0.7972**	0.7964	0.7592	0.8068

Table 1: Comparison of PAKTON versus other methods across models on the ContractNLI test set. The best accuracy/F1[W] per prompting method are shown in **bold**, and the best overall results are <u>underlined</u>.

responses and includes a structured human study with five expert attorneys and a Supreme Court Justice, as well as automated evaluation with G-EVAL (Liu et al., 2023). In both analyses, PAKTON consistently surpasses GPT-4o, outperforming it in eight of nine criteria.

4.1 Quantitative Results

4.1.1 Performance on a classification dataset

Setup. For the quantitative evaluation, PAKTON is evaluated on the ContractNLI dataset (Koreeda and Manning, 2021). In this dataset, a premise denotes an entire contract document, with each premise paired with a corresponding hypothesis. The classification task involves determining whether the hypothesis is entailed by, contradicted by, or not addressed in (neutral with respect to) the associated contract. Given the substantial length of the contracts, the dataset also provides annotated spans that indicate the specific portions of text necessary for making the classification decision.

We evaluate PAKTON on ContractNLI by comparing its performance against several baselines, including models specifically pretrained on legal corpora (Saul (Colombo et al., 2024b)) and LLMs employing different prompting techniques (Table 1). The evaluation metrics presented include overall accuracy and the weighted F1-score (F1[W]), alongside the individual F1-scores for the entailment E, contradiction C, and neutral N classes. The prompting approaches examined include zero-shot (ZS), few-shot (FS)—where entire contractual documents are used as exemplars—and an alternative few-shot setting (FS-Spans), in which only the relevant spans influencing classification are provided. More details are presented in Appendix G.

Results from Table 1 indicate a clear superiority of PAKTON across all evaluated methods, notably outperforming even domain-specific models. Firstly, PAKTON consistently outperforms the Saul baselines regardless the backbone embedded LLM, with marginal benefits emerging even from small models, such as Mistral 7B, even though Saul is explicitly fine-tuned on legal data. Furthermore, PAKTON is highly competitive against the reported prompting baselines; particularly, even the small Mistral 7B with PAKTON outperforms standalone GPT-4o FS prompting, justifying experimentally the rationale behind the selected agentic structure and the agents' curated functionalities.

Furthermore, an interesting observation emerges from comparing the performance across different core LLMs utilizing our framework. The variation in performance among models integrated with PAKTON is minimal, highlighting the robustness of the proposed framework. For example, the F1[W] for Llama 3.1 70B score with PAKTON is 79.03%, while for Gemma 3 27B is 82.83%, yielding a modest difference of only 3.8 percentage points. In contrast, the performance gap between these models under the ZS prompting scenario is significantly larger (22.83%), clearly demonstrating that Gemma 3 27B substantially outperforms Llama 3.1 70B in the baseline ZS setting. This considerable reduction in performance disparity indicates that PAKTON's architecture effectively mitigates the inherent variability among underlying LLMs, ensuring consistently high performance regardless of the core model employed.

To further investigate PAKTON's robustness, we conduct a targeted statistical analysis focusing on variability and dependency. First, we compute the coefficient of variation (CV) across all models' PAKTON–based F1[W] scores, obtaining a low CV of 5.7 (for comparison, ZS scores typically exhibit CVs $> 16\%$). This indicates minimal relative variability—reduced to less than one third under PAKTON compared to ZS—and thus enhanced robustness, yielding consistently stable results regardless of the underlying model's performance. Second, we perform an one–way ANOVA by splitting models into two groups according to the median of their ZS scores. The test reveals no statistically significant difference in PAKTON performance between the high– and low–ZS groups ($F - statistic = 4.19$, $p = 0.075$). Lastly, linear regression analysis assesses the direct relationship between baseline ZS performance and PAKTON, yielding a shallow slope of 0.33, demonstrating that every unit of baseline gain translates into only about $1/3^{rd}$ under PAKTON, thereby compressing absolute performance gaps. Figure 3 plots ZS F1[W] scores (x-axis) versus PAKTON F1[W] scores (y-axis), with the dashed line denoting perfect correlation ($y = x$). The figure shows that, regardless of their initial baseline performance, models converge to a similar outcome under PAKTON, indicating that any underlying LLM can be employed while achieving consistently stable results.

Collectively, these findings demonstrate that PAKTON substantially reduces performance disparities among diverse LLMs, by redistributing the reliance on model quality across other subcomponents of the system. Such robustness is particularly advantageous in the legal domain, where relying on open–source models alleviates the privacy risks associated with sending sensitive contractual or legal information to proprietary systems.

4.1.2 Performance of RAG

Setup. In addition to ContractNLI, we leverage the LegalBench-RAG benchmark (Pipitone and Alami, 2024) to assess the retrieval pipeline (specifically, the *Archivist* and *Researcher* modules) independently of answer generation. This isolation is essential, as PAKTON's overall performance hinges on the relevance of retrieved context. LegalBench-RAG covers four contract-related domains (NDAs, M&A agreements, commercial contracts, and consumer-facing privacy policies), enabling a robust evaluation of retrieval accuracy across heterogeneous legal corpora. We benchmark PAKTON's RAG module against four baselines used in Pipitone and Alami (2024), namely: Naive (fixed 500-character chunks with OpenAI embeddings (OpenAI, 2024a)), RCTS (structure-aware splitting (LangChain, 2024)), Naive+Cohere (chunking with Cohere reranking[3]), and RCTS+Cohere (structural chunking with reranking). Full details are provided in App. F.

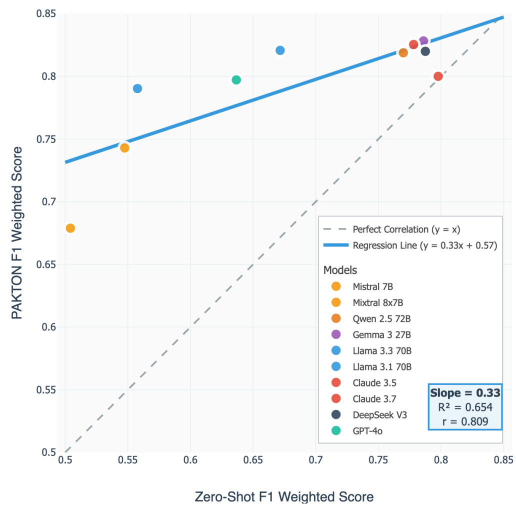


Figure 3: PAKTON F1[W] plotted against ZS scores, with the dashed line denoting perfect correlation and the solid line the regression fit.

Results. Table 2 presents a comparative analysis of retrieval performance across diverse legal corpora, using precision and recall at various k thresholds to benchmark PAKTON against established baselines. Across all datasets, PAKTON consistently and significantly outperforms alternatives at every evaluated k. For instance, on ContractNLI, PAKTON achieves a Recall@1 of 53.14%, nearly 5 times higher than the strongest baseline (11.32%), a trend that holds across all k values. Similar improvements are observed in other datasets: on PrivacyQA (Ravichander et al., 2019), MAUD (Wang et al., 2023), and CUAD (Hendrycks et al., 2021), PAKTON surpasses the best Recall@1 scores by margins often exceeding 20%. Aggregate results

[3]rerank-english-v3.0

		Precision @ k							Recall @ k						
Dataset	**Method**	1	2	4	8	16	32	64	1	2	4	8	16	32	64
PrivacyQA	Naive	7.86	7.31	6.41	5.06	3.58	2.41	1.54	7.45	12.53	20.88	32.38	42.45	54.27	66.07
	RCTS	14.38	13.55	12.34	9.03	6.06	4.17	2.81	8.85	15.21	27.92	42.37	55.12	71.19	84.19
	Naive + Cohere	14.38	13.55	12.34	9.02	6.06	4.17	2.81	8.85	15.21	27.92	42.37	55.12	71.19	84.19
	RCTS + Cohere	13.94	15.91	13.32	9.57	6.88	4.68	3.28	7.32	16.12	25.65	35.60	51.87	64.98	79.61
	PAKTON	**19.94**	**16.84**	**11.44**	**8.62**	**7.38**	**6.42**	**6.08**	**13.34**	**22.43**	**32.67**	**43.39**	**61.65**	**82.30**	**89.42**
ContractNLI	Naive	16.45	14.80	12.53	9.73	6.70	4.65	3.04	11.32	19.10	29.79	45.59	56.75	69.88	86.57
	RCTS	6.63	5.29	3.89	2.81	1.98	1.29	0.90	7.63	11.33	17.34	24.99	35.80	46.57	61.72
	Naive + Cohere	6.63	5.28	3.89	2.81	1.98	1.29	0.90	7.63	11.34	17.34	24.99	35.80	46.57	61.72
	RCTS + Cohere	5.08	5.59	5.04	3.67	2.52	1.75	1.17	4.91	9.33	16.09	25.83	35.04	46.90	62.97
	PAKTON	**33.02**	**30.34**	**17.33**	**9.98**	**5.87**	**4.68**	**4.52**	**53.14**	**67.47**	**80.06**	**89.71**	**95.50**	**99.56**	**99.82**
MAUD	Naive	3.36	2.65	2.18	1.89	1.48	1.06	0.75	2.54	3.12	4.53	8.75	13.16	18.36	25.62
	RCTS	2.65	1.77	1.96	1.40	1.39	1.15	0.82	1.65	2.09	4.59	6.18	12.93	21.04	28.28
	Naive + Cohere	2.64	1.77	1.96	1.40	1.38	1.15	0.82	1.65	2.09	5.59	6.18	12.93	21.04	28.28
	RCTS + Cohere	1.94	2.63	2.05	1.77	1.79	1.55	1.12	0.52	2.48	4.39	7.24	14.03	22.60	31.46
	PAKTON	**25.47**	**17.45**	**10.51**	**7.24**	**5.08**	**3.18**	**1.85**	**23.99**	**30.09**	**34.49**	**46.42**	**59.74**	**74.96**	**82.80**
CUAD	Naive	9.27	8.05	5.98	4.33	2.77	1.77	1.09	12.60	19.47	27.92	40.70	51.02	64.38	75.71
	RCTS	1.97	4.03	4.83	4.20	2.94	1.99	1.25	1.62	8.11	17.72	31.68	44.38	60.04	74.70
	Naive + Cohere	1.97	4.03	4.83	4.20	2.94	1.99	1.25	1.62	8.11	17.72	31.68	44.38	60.04	74.70
	RCTS + Cohere	3.53	4.18	6.18	5.06	3.93	2.74	1.66	3.17	7.33	18.26	28.67	42.50	55.66	70.19
	PAKTON	**11.02**	**8.83**	**6.81**	**4.72**	**2.78**	**2.07**	**1.62**	**16.52**	**24.76**	**33.34**	**46.67**	**59.53**	**77.08**	**86.23**
ALL	Naive	2.40	3.76	4.97	4.33	3.39	2.17	1.29	3.37	8.44	21.30	34.51	48.88	64.47	76.39
	RCTS	6.41	6.16	5.76	4.36	3.09	2.15	1.45	4.94	9.19	16.90	26.30	37.06	49.71	62.22
	Naive + Cohere	6.41	6.16	5.76	4.36	3.09	2.15	1.45	4.94	9.19	16.90	26.30	37.05	49.71	62.22
	RCTS + Cohere	6.13	7.08	6.65	5.02	3.78	2.68	1.81	3.98	8.82	16.10	24.34	35.86	47.54	61.06
	PAKTON	**22.34**	**18.37**	**11.52**	**7.63**	**5.26**	**4.08**	**3.52**	**26.77**	**36.32**	**45.26**	**56.66**	**69.17**	**83.50**	**89.58**

Table 2: Comparative retrieval performance on LegalBench-RAG, evaluated using Precision@k and Recall@k.

reinforce this pattern, with PAKTON achieving more than five-fold increase in Recall@1 (26.77% vs. 4.94%). These improvements are especially critical in the legal domain, where high recall is essential. Failing to retrieve relevants spans can result in flawed reasoning or unsupported conclusions, particularly when legal documents contain conflicting clauses, exceptions, or interdependent provisions that must be interpreted in context.

4.2 Qualitative Results

Setup. To evaluate PAKTON's practical utility, we conduct a human study using 15 benchmark legal questions curated by five attorneys and a Supreme Court Justice to capture authentic legal reasoning across diverse scenarios. Each question is chosen to test a distinct cognitive process and retrieval pattern relevant in legal analysis, ensuring breadth of evaluation. In a *Prolific*[4] survey, participants compare paired responses from PAKTON and GPT-4o[5] across 9 criteria: explainability, evidence, legal/contextual understanding, ambiguity handling, gap acknowledgment, conciseness, coherence, relevance, and completeness. Given PAKTON's emphasis on explainability, evidence tracking, and accessibility, we deliberately evaluate its performance from the perspective of an average, non-expert user, which is its primary target audience. For each criterion, they select the best answer and provided justification, with "None" and "Not Sure" options to ensure robustness. GPT-4o serves as a baseline given its adoption and, for fairness, leverage RAG when external documents are provided (OpenAI, 2024b). Our human study considers a larger amount of evaluated samples over a wider range of criteria in comparison to similar endeavors (Jiang et al., 2024; Shao et al., 2024). Additional details are in Appendix B.

Results. Sixty participants completed the evaluation, comparing PAKTON and GPT-4o outputs across 9 criteria for 15 legal questions, yielding 540 individual judgments. As illustrated in Figure 4, PAKTON is favored over GPT-4o on the majority of evaluation dimensions, a trend consistent across all legal scenarios assessed. The most pronounced advantage is observed in "Completeness," where PAKTON consistently provides more

[4] https://www.prolific.com

[5] gpt-4o-2024-11-20

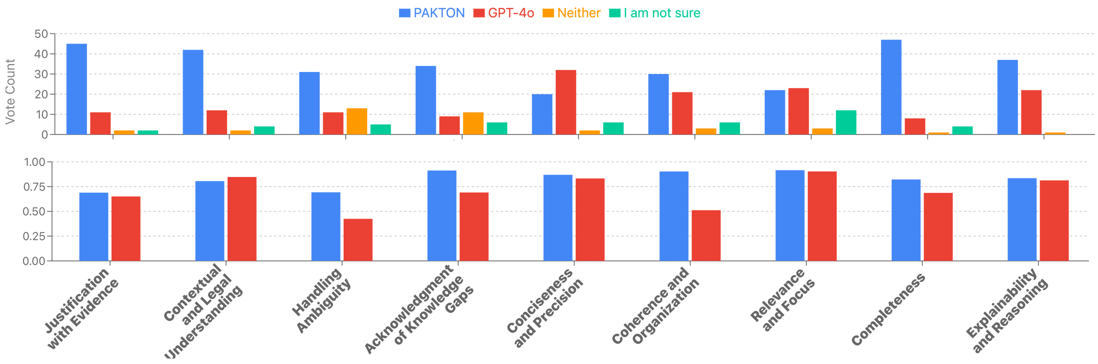


Figure 4: Comparison analysis of PAKTON and GPT-4o. Top plot presents human preferences across nine evaluation criteria aggregated for all questions. Bottom plot shows G-EVAL scores for the same criteria, aggregated across all ContractNLI outputs.

comprehensive responses. PAKTON also excels in "Explainability and Reasoning," offering clearer, stepwise rationales, an important attribute for non-specialist users. To assess the participants' discriminative capacity, we include competing criteria such as "Completeness" versus "Conciseness and Precision," recognizing the inherent tension between exhaustive and succinct responses; as expected, PAKTON excels in completeness, while GPT-4o is preferred for conciseness. For "Relevance and Focus," both models receive comparable scores, reflecting a trade-off between breadth and specificity. These outcomes confirm PAKTON's core objective to generate detailed, report-like answers, with the observed completeness-precision trade-off both anticipated and justified.

4.3 Evaluation using LLMs

Setup. To comprehensively assess PAKTON at scale, we employ G-EVAL (Liu et al., 2023), a widely used LLM-as-a-Judge evaluation framework (Liusie et al., 2024; Wang et al., 2024b; Chiang and Lee, 2023) for NLG output quality across the same criteria as the human study. We evaluate 102 randomly selected ContractNLI samples, comparing PAKTON and GPT-4o on matched inputs. G-EVAL generates criterion-specific scores, supporting fine-grained analysis of response quality, explainability, and reasoning. Further details are demonstrated in Appendix C.

Results across all samples and evaluation criteria prove PAKTON's superiority over GPT-4o in eight out of nine dimensions (Figure 4): PAKTON scores higher in "Explainability and Reasoning", "Justification with Evidence", "Completeness", and "Handling Ambiguity". These outcomes are consistent with PAKTON's design objectives that prioritize detailed and well-supported responses that explicitly reason through legal content. For "Conciseness and Precision", and "Relevance and Focus" criteria, both models demonstrate comparable performance, an anticipated outcome, as these dimensions often conflict with "Completeness", a domain in which PAKTON significantly surpasses GPT-4o, thereby illustrating the inherent trade-off between brevity and depth. The sole criterion in which PAKTON underperforms relative to GPT-4o is "Contextual and Legal Understanding"; while this initially appears counterintuitive, a closer examination of the G-EVAL rationale reveals that PAKTON's responses frequently acknowledge knowledge gaps under uncertainty. Although this is desirable from a transparency perspective, the evaluation framework interprets such acknowledgments as evidence of limited understanding, resulting in lower scores. The alignment between G-EVAL and human judgments is evident in Fig. 4, where results for both methods appear nearly superimposed. A more rigorous statistical analysis that strengthens and inter-validates this agreement is presented in App. C.1.

5 Conclusion

We propose PAKTON, a multi-agent framework for contract analysis that promotes explainability, grounded reasoning, and modular retrieval. By coordinating three specialized agents in a model-agnostic manner, and without further tuning needed, PAKTON generates transparent, well-justified, and query-specific legal reports. Automated and human evaluations prove PAKTON's superiority against both prompted and fine-tuned baselines, not only in accuracy but also in critical dimensions such as explainability, completeness, and evidentiary support—core requirements for legal applications characterized by ambiguity and high-stakes interpretation.

Acknowledgments

This work was supported by the FCT project "OptiGov", ref. 2024.07385.IACDC (DOI 10.54499/2024.07385.IACDC), funded by the PRR under the measure RE-C05-i08.m04.

Limitations

While PAKTON demonstrates strong capabilities in contract analysis, it has several limitations that should be considered when interpreting its results and deploying it in real-world scenarios.

Language Scope. Our system has been tested only on English-language contracts. As legal language varies significantly across languages and cultures, additional adaptation and evaluation would be necessary for multilingual or cross-lingual applications.

Contract and Jurisdiction Coverage. PAKTON has been evaluated on a subset of contract types and does not currently cover the full diversity of legal documents. Similarly, the system has not been tested across different legal jurisdictions. These factors may affect the system's generalizability and legal relevance in broader contexts.

Latency and Cost. Given our focus on the quality and depth of the generated reports, the system prioritizes multi-step reasoning over speed. As a result, response times may be longer compared to general-purpose language models, particularly due to the iterative communication between agents. This design also increases computational cost, making it less suitable for low-latency or resource-constrained environments.

Explainability vs. Efficiency Tradeoff. Our framework is explicitly designed to enhance transparency and reasoning. However, this emphasis on explainability can sometimes result in longer or less concise responses. In prioritizing clarity and justification, the system may occasionally sacrifice brevity or even slightly reduce precision, especially in cases where ambiguity is high and reasoning chains are extended.

Structural Parsing Generalization. The system's structural parsing component is optimized for standard contract formats, which follow consistent patterns of headings, clauses, and subclauses. When documents deviate significantly from these conventions or lack a clearly defined structure, the benefits of structural parsing are reduced. In such cases, the parsing mechanism defaults to a more general-purpose chunking strategy. This fallback does not hinder the system's functionality but may limit the advantages gained from fine-grained hierarchical representation.

Ethical Considerations

Legal Expertise and Overreliance. Our system is developed to aid in contract analysis and increase access to legal information, but it does not serve as a substitute for qualified legal advice. There is a risk that users, particularly non-experts, may overrely on its outputs without proper legal verification. To mitigate this, we recommend clearly communicating the system's limitations and encouraging users to consult legal professionals when making important decisions. **PAKTON should be viewed as an assistive tool, not a definitive authority on legal interpretation**.

Accessibility. Contract analysis is often inaccessible to non-professionals due to its complexity and reliance on legal expertise (as discussed previously). PAKTON's goal is to help democratize contract understanding by providing explainable, user-friendly outputs that can assist individuals without legal backgrounds. We are also trying to offer free access to a publicly deployed version of the system, to the extent that it remains practically and financially feasible. Nevertheless, disparities in access to computational resources may limit the ability of low-resourced groups to deploy or benefit from the system effectively. Future iterations should explore lightweight deployments support to enhance accessibility across a broader range of users.

Security concerns and misuse. As with any system built on large language models, PAKTON may be susceptible to potential misuse and adversarial attacks. Malicious users might attempt to exploit the system to bypass safeguards, distort outputs, or extract sensitive information from retrieved content. While we employ various techniques to reduce some of these risks, further work is needed to strengthen the system's robustness against such threats. We strongly recommend responsible deployment practices and ongoing monitoring when integrating our system into real-world applications.

Human Evaluation. Our human evaluation was carried out by anonymous volunteers who were fairly compensated for their time. All participants were fully informed about the purpose of the study and followed a structured annotation protocol. No personal data was collected or stored at any stage of the evaluation.

References

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haim ing Bao, Mo Bavarian, Jeff Belgum, and Irwan Bello et al. 2023. Gpt-4 technical report.

João Paulo Aires, Roger Leitzke Granada, and Felipe Rech Meneguzzi. 2019. Concon: a contract conflict identifier. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems, 2019, Canadá.*

Pawitsapak Akarajaradwong, Pirat Pothavorn, Chompakorn Chaksangchaichot, Panuthep Tasawong, Thitiwat Nopparatbundit, and Sarana Nutanong. 2025. Nitibench: A comprehensive study of llm framework capabilities for thai legal question answering. *arXiv preprint arXiv:2502.10868*.

Anthropic. 2024a. Introducing contextual retrieval. https://www.anthropic.com/news/contextual-retrieval.

Anthropic. 2024b. Introducing the model context protocol. Accessed: 2025-04-17.

Anthropic. 2025. Claude 3.7 sonnet system card.

Filippo Bianchini, Marco Calamo, Francesca De Luzi, Mattia Macrì, and Massimo Mecella. 2024. Enhancing complex linguistic tasks resolution through fine-tuning llms, rag and knowledge graphs (short paper). In *International Conference on Advanced Information Systems Engineering*, pages 147–155. Springer.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, André Martins, Fabrizio Esposito, Vera L'ucia Raposo, Sofia Morgado, and Michael Desa. 2024a. Saullm-7b: A pioneering large language model for law. *ArXiv*, abs/2403.03883.

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Rui Melo, Dominic Culver, Sofia Morgado, Etienne Malaboeuf, Gabriel Hautreux, Johanne Charpentier, and Michael Desa. 2024b. Saullm-54b & saullm-141b: Scaling up domain adaptation for the legal domain. *ArXiv*, abs/2407.19584.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*.

Jiaxi Cui, Zongjia Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, and Dongjie Ji et al. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

André V. Duarte, João DS Marques, Miguel Graça, Miguel Freire, Lei Li, and Arlindo L. Oliveira. 2024. LumberChunker: Long-form narrative document segmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6473–6486, Miami, Florida, USA. Association for Computational Linguistics.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *ArXiv*, abs/2404.16130.

Melvin A. Eisenberg. 2022. Legal reasoning.

Wenqi Fan, Yujuan Ding, Liang bo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Knowledge Discovery and Data Mining*.

Thomas Frei. 2016. An artificial intelligence approach to legal reasoning.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *ArXiv*, abs/2312.10997.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur

Hinsvark, and 5 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *ArXiv*, abs/2410.05779.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. *NeurIPS*.

Quzhe Huang, Mingxu Tao, Chen Zhang, Zhenwei An, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023. Lawyer llama technical report. *arXiv preprint arXiv:2305.15062*.

Alexandre Yukio Ichida and Felipe Meneguzzi. 2021. Detecting logical relation in contract clauses. *arXiv preprint arXiv:2111.01856*.

Institute for Supply Management. Institute for supply management (ism). `https://www.ismworld.org/`. Accessed: 2025-04-01.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Yucheng Jiang, Yijia Shao, Dekun Ma, Sina Semnani, and Monica Lam. 2024. Into the unknown unknowns: Engaged human learning through participation in language model agent conversations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9917–9955, Miami, Florida, USA. Association for Computational Linguistics.

Manuj Kant, Sareh Nabi, Manav Kant, Roland Scharrer, Megan Ma, and Marzieh Nabi. 2025. Towards robust legal reasoning: Harnessing logical llms in law. *arXiv preprint arXiv:2502.17638*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *ArXiv*, abs/2004.04906.

Yuta Koreeda and Christopher Manning. 2021. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1907–1919, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jinqi Lai, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and Philip S Yu. 2024. Large language models in law: A survey. *AI Open*.

LangChain. 2024. Recursive text splitter. `https://python.langchain.com/v0.1/docs/modules/data_connection/document_transformers/recursive_text_splitter/`. Accessed: 2024-08-15.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401.

Pei Liu, Xin Liu, Ruoyu Yao, Junming Liu, Siyuan Meng, Ding Wang, and Jun Ma. 2025. Hm-rag: Hierarchical multi-agent multimodal retrieval augmented generation. *arXiv preprint arXiv:2504.12330*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Adian Liusie, Potsawee Manakul, and Mark Gales. 2024. LLM comparative assessment: Zero-shot NLG evaluation through pairwise comparisons using large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 139–151, St. Julian's, Malta. Association for Computational Linguistics.

Naielly Lopes Marques, Carlos de Lamare Bastian-Pinto, and Luiz Eduardo Teixeira Brandão. 2024. When contract clauses conflict: The salvador light rail vehicle concession. *Case Studies on Transport Policy*, 16:101168.

Raja Sekhar Reddy Mekala, Yasaman Razeghi, and Sameer Singh. 2024. EchoPrompt: Instructing the model to rephrase queries for improved in-context learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 399–432, Mexico City, Mexico. Association for Computational Linguistics.

OpenAI. 2024a. Embedding models. `https://platform.openai.com/docs/guides/embeddings/embedding-models`. Accessed: 2024-08-15.

OpenAI. 2024b. Optimizing file uploads in chatgpt enterprise. Accessed: 2025-04-12.

Nicholas Pipitone and Ghita Houir Alami. 2024. Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain. *Preprint*, arXiv:2408.10343.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, and Jianhong Tu et al. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Abhilasha Ravichander, Alan W Black, Shomir Wilson, Thomas Norton, and Norman Sadeh. 2019. Question answering for privacy policies: Combining computational and legal perspectives. *Preprint*, arXiv:1911.00841.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3:333–389.

Cheol Ryu, Seolhwa Lee, Subeen Pang, Chanyeol Choi, Hojun Choi, Myeonggee Min, and Jy-Yong Sohn. 2023. Retrieval-based evaluation for llms: a case study in korean legal qa. In *Proceedings of the Natural Legal Language Processing Workshop 2023*, pages 132–137.

Binita Saha, Utsha Saha, and Muhammad Zubair Malik. 2024. Advancing retrieval-augmented generation with inverted question matching for enhanced qa performance. *IEEE Access*.

Yijia Shao, Yucheng Jiang, Theodore A. Kanell, Peter Xu, Omar Khattab, and Monica S. Lam. 2024. Assisting in writing wikipedia-like articles from scratch with large language models. In *North American Chapter of the Association for Computational Linguistics*.

ShengbinYue ShengbinYue, Ting Huang, Zheng Jia, Siyuan Wang, Shujun Liu, Yun Song, Xuan-Jing Huang, and Zhongyu Wei. 2025. Multi-agent simulator drives language models for legal intensive interaction. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6537–6570.

Dong Shu, Haoran Zhao, Xukun Liu, David Demeter, Mengnan Du, and Yongfeng Zhang. 2024. Lawllm: Law large language model for the us legal system. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 4882–4889.

Marco Siino, Mariana Falco, Daniele Croce, and Paolo Rosso. 2025. Exploring llms applications in law: A literature review on current legal nlp approaches. *IEEE Access*.

Raymond W. Smith. 2007. An overview of the tesseract ocr engine. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2:629–633.

Raymond W. Smith. 2013. History of the tesseract ocr engine: what worked and what didn't. In *Electronic imaging*.

Jingyun Sun, Chengxiao Dai, Zhongze Luo, Yangbo Chang, and Yang Li. 2024. Lawluo: A chinese law firm co-run by llm agents. *arXiv preprint arXiv:2407.16252*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, and Ivo Penchev et al. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Sriram Veturi, Saurabh Vaichal, Reshma Lal Jagadheesh, Nafis Irtiza Tripto, and Nian Yan. 2024. Rag based question-answering for contextual response prediction system. *arXiv preprint arXiv:2409.03708*.

Rahman SM Wahidur, Sumin Kim, Haeung Choi, David S Bhatti, and Heung-No Lee. 2025. Legal query rag. *IEEE Access*.

Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. 2025. Chain-of-retrieval augmented generation. *ArXiv*, abs/2501.14342.

Steven H. Wang, Antoine Scardigli, Leonard Tang, Wei Chen, Dimitry Levkin, Anya Chen, Spencer Ball, Thomas Woodside, Oliver Zhang, and Dan Hendrycks. 2023. Maud: An expert-annotated legal nlp dataset for merger agreement understanding. *Preprint*, arXiv:2301.00876.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788.

Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, Ruicheng Yin, Changze Lv, Xiaoqing Zheng, and Xuanjing Huang. 2024a. Searching for best practices in retrieval-augmented generation. In *Conference on Empirical Methods in Natural Language Processing*.

Yuqi Wang, Lyuhao Chen, Songcheng Cai, Zhijian Xu, and Yilun Zhao. 2024b. Revisiting automated evaluation for long-form table question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14696–14706, Miami, Florida, USA. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Nirmalie Wiratunga, Ramitha Abeyratne, Kyle Martin Lasal Jayawardena, Stewart Massie, Ikechukwu Nkisi-Orji, Ruvan Weerasinghe, Anne Liret, and Bruno Fleisch. Cbr-rag: Case-based reasoning for retrieval augmented generation in llms for legal question answering." arxiv, april 4, 2024.

World Commerce & Contracting. 2020. The real value of contracting. `https://www.worldcc.`

com/Resources/Content-Search/Details/articleid/885. Accessed: 2025-04-01.

Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, Jian-Guang Lou, and Shuai Ma. 2024. Re-reading improves reasoning in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15549–15575, Miami, Florida, USA. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *ArXiv*, abs/2210.03629.

Kepu Zhang, Weijie Yu, Zhongxiang Sun, and Jun Xu. 2025. Syler: A framework for explicit syllogistic legal reasoning in large language models. *arXiv preprint arXiv:2504.04042*.

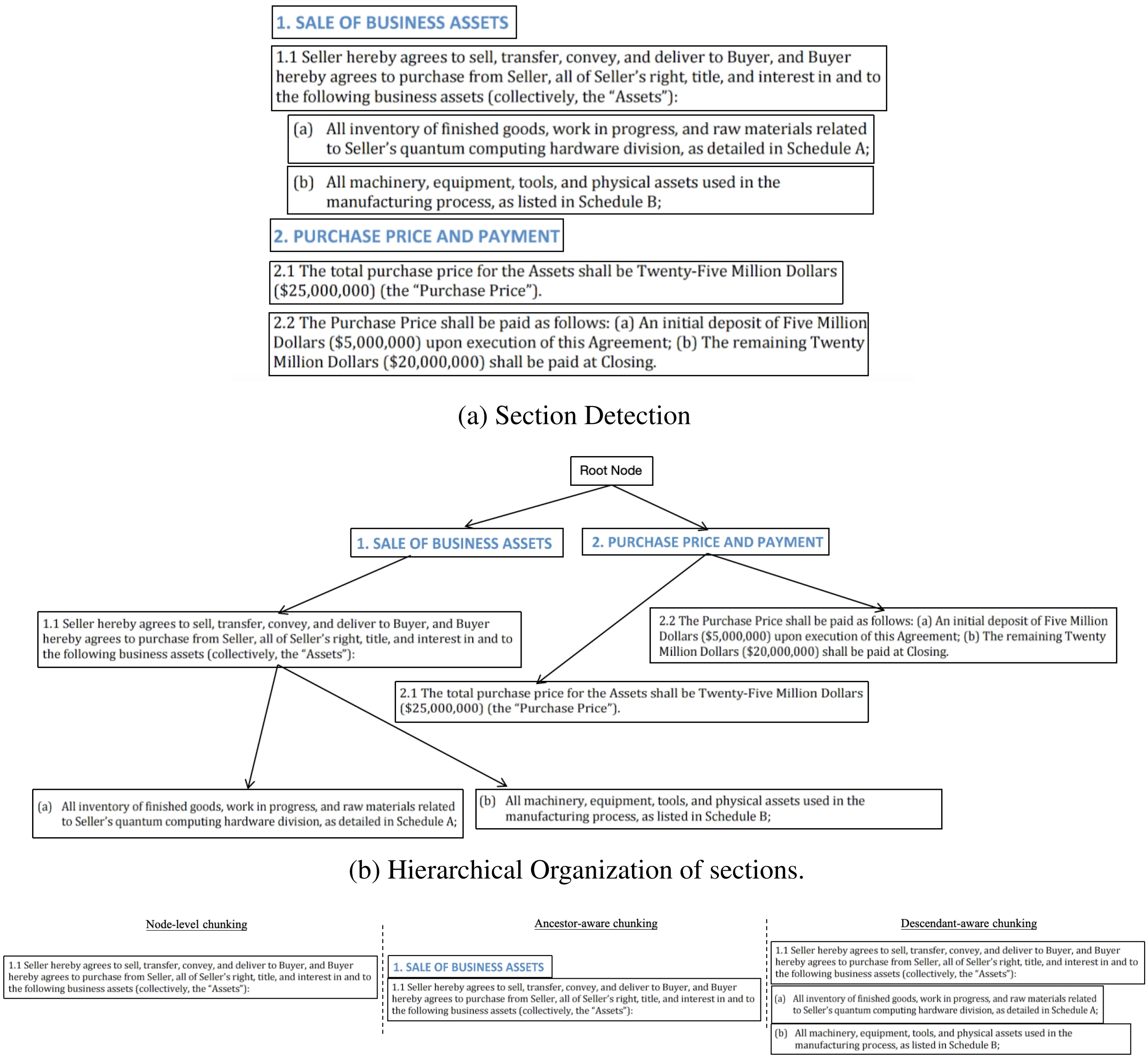
Jun Zhao, Can Zu, Xu Hao, Yi Lu, Wei He, Yiwen Ding, Tao Gui, Qi Zhang, and Xuan-Jing Huang. 2024a. Longagent: Achieving question answering for 128k-token-long documents through multi-agent collaboration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16310–16324.

Xinyang Zhao, Xuanhe Zhou, and Guoliang Li. 2024b. Chat2data: An interactive data analysis system with rag, vector databases and llms. *Proceedings of the VLDB Endowment*, 17(12):4481–4484.

Hong Zhou, Li Zhou, Binwei Gao, Wen Huang, Wenlu Huang, Jian Zuo, and Xianbo Zhao. 2024. Intelligent identification of risks in construction contract clauses based on semantic reasoning. *Engineering, Construction and Architectural Management*.

A Framework's Implementation Details

Figure 5 depicts the Tree Representation Process of the *Archivist* as described in 3.1.



(a) Section Detection

(b) Hierarchical Organization of sections.

(c) Contextual Embeddings for node "1.1 Seller ..."

Figure 5: Tree Representation and Chunking Workflow for Contract Documents

A.1 Prompts

Throughout the framework, various prompts are employed across multiple stages to guide the system's behavior. We provide the prompts used in two of the core components of the framework: the generation of interrogation questions and the refinement of the report. The complete set of prompts is available in our GitHub repository.

INTERROGATION SYSTEM PROMPT

You are a skilled legal interrogator conducting an in-depth interview with a legal researcher. Your objective is to extract **comprehensive, well-supported legal information** by formulating precise, strategic questions.
The goal is **not simply to obtain answers**, but to gather authoritative legal evidence, reasoning, and precedents to thoroughly address the following legal question:
<question>{userQuery} </question>
Additional Context: The following background information relevant to the question is provided:
<context>{userContext} </context>
Additional Instructions: You must take into account the following instructions:
<intructions>{userInstructions} </intructions>
—
Critically Consider the Existing Report Before Asking New Questions:
You have been provided with a **report summarizing the interrogation so far**. This report serves as a **synthesis of key legal arguments, findings, acknowledged knowledge gaps, and preliminary reasoning** extracted from the conversation.
Before forming your next question, **carefully analyze this report**, which includes:
- The **preliminary reasoning and draft interpretation**—a tentative legal direction that has emerged, but is still subject to revision.
- **Explicitly acknowledged knowledge gaps**—areas where the legal researcher did not provide sufficient clarity, evidence, or citations.
- **Remaining uncertainties and conflicting viewpoints**, including legal areas where additional investigation is required.
- **Follow-up questions that have already been identified** to refine the legal analysis further.

You must use this information **strategically** to craft your next question.
—
Your Role:
- You have **{remaining_questions} questions remaining**, so each question must be maximally informative.
- Your goal is to **clarify uncertainties, challenge assumptions, and press for concrete legal sources** to fill the knowledge gaps.
- Your questions should probe deeper into weak or vague responses, pressing for **specific legal precedents, case law, statutory references, and counterarguments**.
- Avoid redundancy—do not ask questions that have already been answered in the report. Instead, **build upon previous insights** and push the conversation forward.

—
...
—
Completion: Once you are fully satisfied that you have gathered all necessary legal insights, you may conclude the interrogation by stating: *"Thank you, I am now in a position to answer the question with confidence."*
You will be given:
- The report summarizing the previous exchange with the legal researcher.
- The list of previous questions asked so far.

Use this information to ensure your next question is targeted, strategic, and maximally informative.

INTERROGATION USER PROMPT

The following report summarizes the previous exchange between you and the legal researcher.
<report>{report} </report>
This report contains:
- **A preliminary interpretation or draft answer**, which is subject to revision.
- **Explicitly acknowledged gaps in legal reasoning**—areas that require further clarification.
- **Conflicting viewpoints or legal uncertainties** that need to be resolved.
- **Follow-up questions that have been identified** to improve the legal analysis.

The following questions have been asked so far:
<questions>{questions} </questions>

You must carefully analyze the above report before crafting your next question.
Your next question should:
- **Push the conversation forward**—do not repeat questions that have already been asked.
- **Target unresolved knowledge gaps** and press for **specific legal references**.
- **Challenge weak or unsupported reasoning**—seek case law, statutes, or counterarguments.
- **Refine or reassess the preliminary interpretation**, if needed.
- **Help move toward a stronger, well-supported legal answer**.

Now, continue your interrogation.

REPORT REFINEMENT SYSTEM PROMPT

You are a legal technical writer tasked with **refining** a structured, professional legal report based on new information from an interrogation-style conversation between a legal interrogator and a legal researcher.

—

Your Objective: You will be given a legal question and an **existing draft report**. Your goal is to **analyze the updated conversation** and integrate the new insights, arguments, and legal interpretations into the existing report—always ensuring that the refinements directly contribute to answering the legal question—while maintaining a **structured, authoritative, and professional** legal analysis. DO NOT just append the new information at the end. Rewrite the report so it reads as one clear, complete, and updated version.

The final/refined report must be written as if it is the only version that exists. DO NOT acknowledge the existence of the previous report and any conversation.

Your role is **not to provide a final answer or definitive conclusion**, but to further develop the **key insights, arguments, and reasoning gaps** necessary to reach a legally sound conclusion. The refined report may challenge or revise the preliminary direction taken earlier.

—

Guidelines for Writing the Report:

1. Analyze the Updated Conversation:

- Carefully review the **existing legal report** and the **new conversation transcript**.
- Identify **new legal arguments, precedents, counterarguments, or reasoning** that emerge and **critically evaluate** whether they change or reinforce the preliminary findings.
- Challenge any previous interpretations if needed—**do not assume the original direction is correct**.
- Identify **knowledge gaps or missing legal evidence** that still prevent a definitive answer.

2. Refine the Legal Report (Markdown Formatting):

- **Preserve the original report structure but enhance it where needed:**

Title: Keep or modify the title if the updated information suggests a more precise framing.

Summary:
Keep or modify the summary if the updated information suggests a more precise introduction to the topic.

Legal Reasoning & Analysis:
- Expand the reasoning section with **new legal arguments or counterarguments** introduced.
- Clearly **indicate changes or clarifications** while ensuring logical consistency.
- Ensure that all conclusions remain legally sound and properly substantiated.

Preliminary Answer & Direction for Further Research:
- Instead of refining toward a definitive answer, provide an **updated draft interpretation** or **alternative possible directions** based on new findings.
- If previous reasoning is now in doubt, **state why and explore alternative legal views**.
- Clarify **what would be required** to reach a more confident answer.

Gaps & Next Questions:
- Explicitly state what **additional legal information, precedents, or sources** are needed to refine the analysis.
- List **follow-up questions** that could help clarify uncertainties.

Sources:
- List all cited legal sources using numbered references **[1], [2]**, etc.
- If URLs or case references exist, include them in this section.
- Incorporate new references, direct quotes, and citations from the conversation where relevant.
- Ensure each reference includes metadata to help locate the original text (e.g., clause number, page number, section name, etc.).

3. Writing Style & Formatting:

- Use **formal legal writing**—precise, objective, and authoritative.
- Be **concise yet comprehensive** (approximately **500 words max**).
- Ensure **clarity and logical flow**—no redundant or unclear statements.
- **Do not reference the interrogator or researcher**—present findings as a **standalone report**.
- **Whenever possible, include direct quotes from the original context** in your references to justify your claim. Enclose these quotes in quotation marks ("") to highlight the exact supporting spans.
- **For each reference, specify how to locate the relevant information** in the original text (like clause number, page number, section name, etc.)

4. Handling Insufficient Data:

- If the conversation still lacks sufficient legal clarity or citations, **explicitly acknowledge these gaps**.

| ... |

Now, analyze the new conversation and **refine the existing legal report** accordingly.

REPORT REFINEMENT USER PROMPT

Refine the following **legal report** based on the newly provided conversation between a **legal interrogator** and a **legal researcher**. Prioritize the most important and relevant information from both the existing report and the new conversation—keeping only the content that meaningfully impacts the answer to the legal question.

—

Legal Question:
<question>
{userQuery}
</question>
Additional Context:
The following background information relevant to the question is provided:
<context>
{userContext}
</context>
Updated Legal Conversation Transcript:
<conversation>
{conversation}
</conversation>
Existing Legal Report:
<legal_report>
{existingReport}
</legal_report>

—

Refinement Guidelines:

- Carefully **incorporate relevant new legal arguments, precedents, and reasoning** from the conversation. DO NOT just append the new information at the end. Rewrite the report so it reads as one clear, complete, and updated version.
- **Critically evaluate** the existing legal report against the new conversation transcript.
- Do **not assume the existing direction is correct**—if the new insights challenge prior reasoning, revise accordingly.
- Identify **knowledge gaps and missing evidence** that prevent a definitive answer.
- **Explicitly highlight any contradictions or multiple possible legal interpretations**.
- List **follow-up questions** that need to be answered to reach a more well-founded conclusion.
- **Cite new references** where applicable and preserve the report's structured format.
- **ALWAYS ensure that every refinement you make directly enhances the accuracy and clarity of the answer to the legal question.**
- The final/refined report must be written as if it is the only version that exists. DO NOT acknowledge the existence of the previous report and any conversation.

Now, refine the legal report based on the new information.

B Human Evaluation

We deployed a survey on *Prolific*, an online research platform widely used in academic studies. Prolific ensures participant anonymity and does not allow researchers to interfere with or influence responses. Participants were asked to compare answers from PAKTON and ChatGPT for the same legal question, based on specific evaluation criteria.

We intentionally did not restrict participation to legal professionals. Our goal was to understand how well PAKTON's responses are received by the general public, especially in comparison to a widely used baseline like ChatGPT. The only eligibility requirements were that participants must be fluent in English and have completed at least compulsory education (e.g., high school level), to ensure basic reading comprehension and critical reasoning skills.

Moreover, we opted for maintaining labeled samples regarding which one stems from PAKTON and which from ChatGPT; this way, human evaluators were fully aware of the source of each response, aiming to simulate a real-world usage scenario: in practice, users do know whether they are interacting with ChatGPT or PAKTON. This transparency was crucial to ensure that any preference expressed for PAKTON over ChatGPT reflected a conscious, informed choice, especially given the fact that ChatGPT is the most widely used publicly available LLM interface, particularly in the context of contract review. In doing so, we can be confident that any favoring of PAKTON was not due to anonymity, but instead due to users deliberately valuing PAKTON's output over that of an industry-standard tool and signaling that they would be willing to use PAKTON over ChatGPT for contract analysis if it were publicly available.

Our human evaluation was carried out by anonymous volunteers who were fairly compensated for their time. All participants were fully informed about the purpose of the study and followed a structured annotation protocol. No personal data was collected or stored at any stage of the evaluation. The instructions given to human annotators are presented in Table 3.

Figure 6 presents a comparative assessment between PAKTON and GPT-4o, based on human judgments across multiple criteria. PAKTON consistently outperformed GPT-4o, both at the individual response level and in aggregate preferences.

Participants were paid £9/hour, exceeding Prolific's minimum compensation rate of £6/hour. While the expected duration of the survey was 5 minutes, the median completion time was closer to 15 minutes. To fairly account for the additional effort, participants who spent more time were rewarded with bonuses, calculated based on Prolific's minimum rate.

Evaluators assessed PAKTON's outputs via the interface shown in Figure 7, reviewing both predefined examples and interacting with the system in real time using their own queries. Feedback was overwhelmingly positive, highlighting the tool's practical utility.

Criterion	Instructions
Explainability and Reasoning	Evaluate whether the report clearly and transparently explains not only the final conclusion, but also the reasoning process and supporting evidence in a step-by-step, understandable manner. The explanation should guide the reader through the logic in a way that supports comprehension, avoiding unexplained jumps in logic.
Justification with Evidence	Determine whether the statements and claims are explicitly justified with relevant, specific, and clearly cited evidence (e.g., direct quotations, clause references). The justification should be traceable, allowing the reader to locate the original source material.
Contextual and Legal Understanding	Assess whether the report demonstrates a deep and accurate understanding of the document, its legal terminology, and the broader context. Consider whether it correctly interprets clauses and captures implied assumptions or legal concerns behind the question.
Handling Ambiguity	Determine whether the report identifies and handles ambiguities in the source material appropriately, such as by presenting multiple interpretations or justifying a chosen one clearly.
Acknowledgment of Knowledge Gaps	Evaluate whether the report explicitly acknowledges when available information is insufficient to support a conclusion, avoiding speculation or overconfidence.
Conciseness and Precision	Assess whether the report communicates clearly and efficiently, avoiding unnecessary repetition or verbosity, while still covering all key points.
Coherence and Organization	Check whether the report is logically structured, flows smoothly, and maintains clarity across sections. Transitions between ideas should be natural and helpful.
Relevance and Focus	Evaluate whether the report stays on topic and maintains focus on answering the core question, avoiding tangents or irrelevant content.
Completeness	Assess whether the report addresses all important aspects of the question and offers a contextually broad and holistic answer. It should not omit any major points or perspectives.

Table 3: Instructions given to human annotators for each evaluation criterion used in the PAKTON vs. ChatGPT comparison. Similar instructions were given to the G-EVAL framework.

C G-EVAL Experiments

We set the temperature to 0 to ensure deterministic responses, facilitating reproducibility.

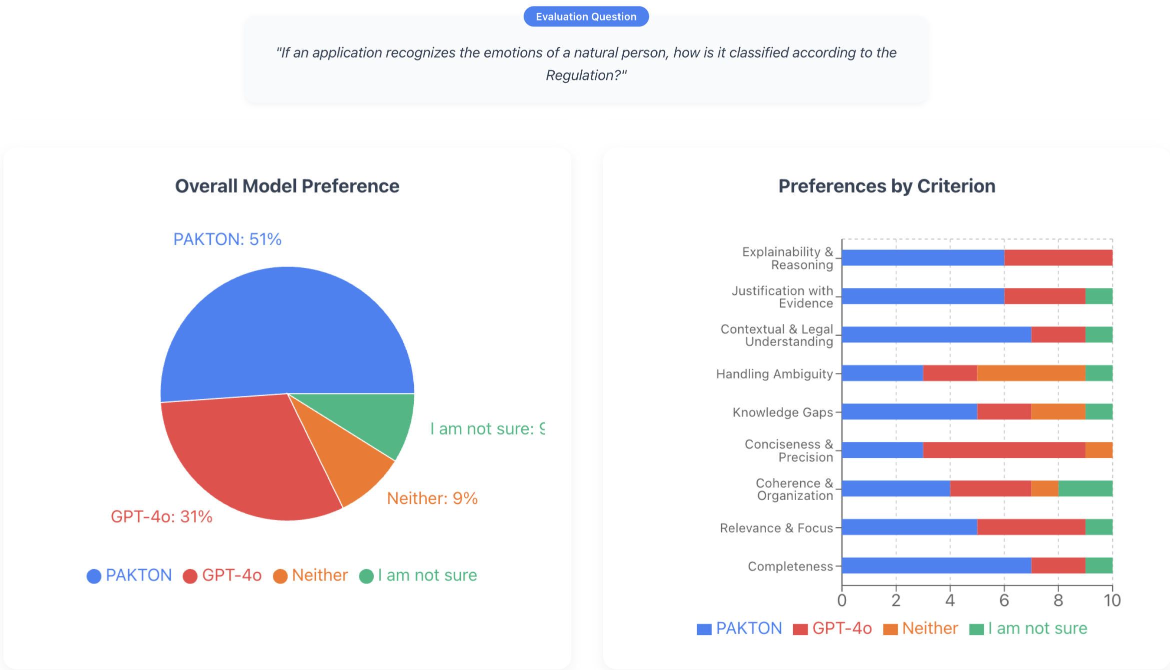
As part of our evaluation design, we ensured fair model comparison settings. For “GPT”, we implemented a (RAG) pipeline. For PAKTON, we limited tool usage to strictly in-document retrieval (disabling access to external tools like web search), and capped the number of interrogation turns at five. GPT-4o served as the underlying model for both systems to eliminate base model performance discrepancies and isolate differences due to architecture and orchestration.

The aggregated scores for all evaluated outputs of PAKTON and GPT-4o on the ContractNLI dataset are presented in Table 4. The distribution of the aggregated scores across all evaluation criteria, as computed by the G-EVAL framework, is illustrated in Figure 8. Table 5 provides a concrete example demonstrating how PAKTON’s transparency in acknowledging knowledge gaps leads to penalized scores in the "Contextual and Legal Understanding" criterion.

C.1 Statistical Agreement between LLM and Human Judgments

The alignment between the LLM-based evaluation and human judgments is visually evident in Figure 4, where the results of both methods appear nearly superimposed. While such visual inspection already suggests consistency between the two evaluation approaches, a more rigorous statistical analysis was performed to strengthen and intervalidate this agreement.

Transformation process. To enable a direct comparison with human evaluation, the absolute G-EVAL scores were transformed into categorical votes. Specifically, for each data sample and evaluation criterion, if the difference between PAKTON and GPT-4o was less than 1%, the outcome was considered a tie and placed in the *Neither* category (mirroring the human “Neither” option). Otherwise, the vote was assigned to the system with the higher score. This procedure yielded vote distributions across the three categories (*PAKTON*, *GPT-4o*, *Neither*), which were normalized to percentages. Human evaluation data were processed



(a) Preference based on responses for a single question

(b) Overall Model Preference aggregated across all criteria and all questions

Figure 6: Comparative analysis of PAKTON vs. GPT-4o based on human evaluator judgments across different criteria

analogously, excluding "I am not sure" responses.

Distributional similarity and variance consistency. Agreement was assessed on these percentage distributions. Across criteria, the average cosine similarity between LLM- and human-derived distributions was 0.88, with an average MAE of 12.6%. Crucially, sensitivity analysis excluding the single outlier criterion (*Contextual and Legal Understanding*)—which, as noted in Section 4.3, was misinterpreted by the G-EVAL framework—further strengthens the agreement: average cosine similarity rises to 0.9164, while average MAE falls to 10.88% (RMSE 14.06%). Variance-comparison tests further indicate no significant differences in dispersion between methods (all F-tests $p \geq 0.05$ across categories), and two-sample distribution tests (Kolmogorov–Smirnov and Mann–Whitney U) similarly do not detect significant differences between the LLM and human distributions (all $p \geq 0.05$), including in the outlier-excluded analysis (e.g., K–S $p = 0.28$–0.98 across categories).

Where agreement is strongest. With the outlier removed, several criteria exhibit near-identity between methods:

- **Completeness:** cosine 0.9992, MAE 1.77%
- **Explainability and Reasoning:** cosine 0.9915, MAE 4.07%
- **Relevance and Focus:** cosine 0.9901, MAE 3.86%
- **Justification with Evidence:** cosine 0.9751, MAE 6.69%

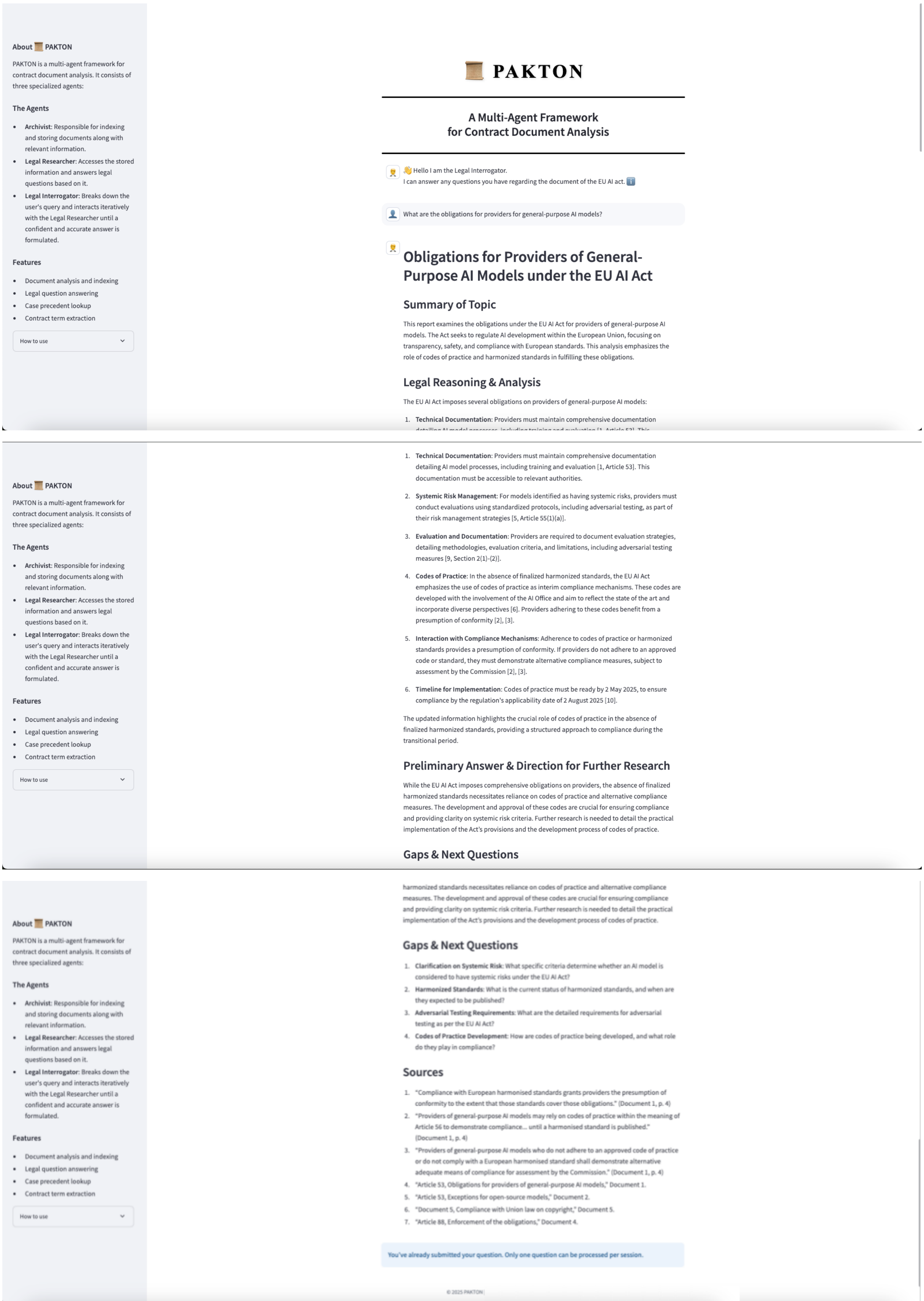


Figure 7: The user interface (UI) of PAKTON employed during the human evaluation with study participants.

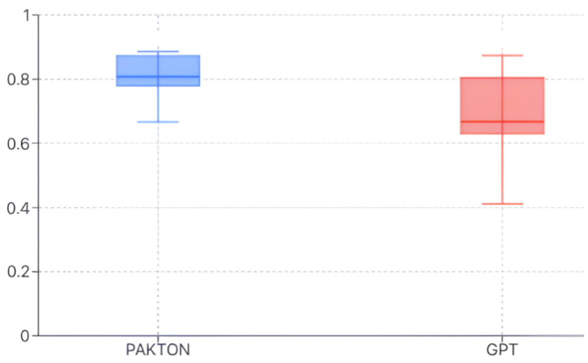


Figure 8: Comparative analysis of PAKTON vs. GPT-4o based on G-EVAL across different criteria. Distribution of the aggregated scores across all criteria.

Evaluation Criterion	PAKTON	GPT
Justification with Evidence	0.6887	0.6503
Contextual and Legal Understanding	0.8054	0.8472
Handling Ambiguity	0.6925	0.4248
Acknowledgment of Knowledge Gaps	0.9124	0.6905
Conciseness and Precision	0.8691	0.8323
Coherence and Organization	0.9024	0.5118
Relevance and Focus	0.9155	0.9030
Completeness	0.8216	0.6866
Explainability and Reasoning	0.8350	0.8127

Table 4: Comparison of PAKTON and GPT scores acquired across different evaluation criteria aggregated for all examined outputs.

Criterion	Contextual and Legal Understanding
Score	0.87549
Reason	The report accurately interprets legal terminology and context, correctly identifies that employees are not third parties, and aligns with the contract's confidentiality obligations. It fully understands the user's query, addressing legal issues and assumptions, **but suggests further research for comprehensive understanding.**

Table 5: Example illustrating score penalization of "Contextual and Legal Understanding" due to further research section of the report.

- **Acknowledgment of Knowledge Gaps:** cosine 0.9769, MAE 7.24%

Even for more demanding dimensions such as *Conciseness and Precision* and *Coherence and Organization*, the methods preserve broadly similar distributional shapes, despite larger absolute gaps (e.g., MAE up to $\sim$ 22–23%). Category-level calibration offsets are small to moderate (e.g., mean differences of $+13.7\%$ for "PAKTON," -11.6% for "GPT," and -2.1% for "Neither" in the outlier-excluded analysis) and do not overturn these distributional agreements.

Takeaway. Beyond the visual overlap in Figure 4, the quantitative evidence, particularly after excluding a single outlier criterion, shows that the LLM-based judging and human evaluation yield highly similar vote distributions across *PAKTON*, *GPT-4o*, and *Neither*. The combination of high cosine similarity, reduced MAE/RMSE, matched variances, and non-significant distribution tests substantiates the claim that the two methods are in strong statistical agreement, thereby mutually reinforcing the validity of our evaluation results.

D Representative Output from PAKTON and Comparative Analysis with GPT (RAG)

The following examples illustrate the outputs generated by PAKTON and GPT (RAG) in response to the same ContractNLI input, consisting of a contract-hypothesis pair. The input is as follows:

CONTRACT NLI INPUT

Is the following hypothesis ENTAILMENT, CONTRADICTION, or NEUTRAL according to the content of the contract: <hypothesis>Receiving Party shall destroy or return some Confidential Information upon the termination of Agreement.</hypothesis>?

The problem is categorized as a Natural Language Inference (NLI) task for contract analysis. The answer can be one of the following:

ENTAILMENT:
- The hypothesis is logically true based on the content of the contract.
- It is explicitly stated or can be directly inferred.

CONTRADICTION:
- The hypothesis directly or logically conflicts with the contract's content.

NEUTRAL:
- There is insufficient information to conclude if it is CONTRADICTION or ENTAILMENT.
- If the hypothesis isn't mentioned in the contract or is unrelated.

Based on this input, GPT produced the following output:

GPT (RAG) OUTPUT

The contract does indeed entail the hypothesis that the Receiving Party shall destroy or return some Confidential Information upon the termination of the Agreement.

As outlined in section 9 of the contract, labeled "Return of JEA Confidential Information", it specifies that
"the Recipient shall immediately return and redeliver to the other all tangible material embodying the JEA Confidential Information provided hereunder. . . and all other documents or materials ('Notes'). . . upon the earlier of:
I. the completion or termination of the dealings between the parties contemplated hereunder; or
II. the termination of this Agreement; or
III. at such time as JEA may so request."

It also provides an alternative where, with the written consent of JEA, the Recipient may destroy such information, and it provides details on how this destruction should be carried out and verified.

These obligations regarding handling Confidential Information are specific to the event of the termination of the Agreement, directly supporting the hypothesis.

While the output of PAKTON for the same input is presented as follows:

PAKTON OUTPUT

Title: Analysis of Contractual Obligations Regarding Confidential Information Post-Termination

Summary of Topic:
The legal question under consideration is whether the hypothesis that "Receiving Party shall destroy or return some Confidential Information upon the termination of Agreement" is classified as ENTAILMENT, CONTRADICTION, or NEUTRAL according to the contract's content. This involves determining whether the contract explicitly or implicitly supports, contradicts, or is neutral regarding this hypothesis.

Legal Reasoning & Analysis:
The contract under review contains specific clauses that delineate the obligations of the Receiving Party regarding the handling of confidential information upon the termination of the agreement. The updated analysis provides a more nuanced understanding of these obligations and introduces potential exceptions and modifications:

1. Return of Confidential Information:
- The contract specifies that upon request by the disclosing party, the Receiving Party is obligated to return all confidential information.
- Exceptions exist for analyses prepared by or for the Receiving Party, and they may retain one copy for compliance with applicable laws or record retention policies [1], [2], [3].

2. Destruction of Confidential Information:
- The contract provides an option for the Receiving

Party to destroy the confidential information instead of returning it, provided there is written consent from the disclosing party.
- The destruction must be verifiable, and computerized data must be non-recoverably erased [2] (Clause 9).

3. Exceptions to Obligations:
- Certain types of confidential information are exempt from the return or destruction obligations.
- This includes information received from a third party without breach, information known prior to disclosure, publicly known information, independently developed information, and information ordered for public release by a government agency [1], [3], [4], [5].

4. Retention for Legal Compliance:
- The receiving party may retain one copy of the confidential information as needed to comply with applicable law and/or the party's record retention policies.
- This retained information must be kept subject to the terms of the agreement or destroyed [1], [2], [3].
- The retention is allowed to enable compliance with statutory document retention policies and must be necessary for legal compliance, suggesting that any retention beyond this scope would not be permissible [4].

5. Impact of Legal Compliance and Record Retention Policies:
- The contract explicitly allows retention for compliance with legal and record retention requirements, thereby modifying the general obligation to return or destroy the information upon termination.
- This impacts the classification of the hypothesis as ENTAILMENT, as the contract articulates exceptions to the return or destruction requirement for compliance purposes [1], [2], [3].

Preliminary Answer & Direction for Further Research:
The hypothesis that the Receiving Party shall destroy or return some Confidential Information upon termination appears to be **ENTAILMENT** with specific conditions and exceptions.
The contract outlines obligations for returning or destroying confidential information, with allowances for retention due to legal compliance and record retention policies.
The determination of ENTAILMENT assumes that the specific conditions outlined in the contract are met, such as the disclosing party's request or consent and the applicability of exceptions for certain types of information.
Further research should focus on verifying whether these conditions apply universally to all confidential information or are subject to specific circumstances or exceptions not covered in the provided excerpts.

Gaps & Next Questions:
- Clarification on Universal Application: Is the obligation to return or destroy confidential information upon termination applicable to all types of confidential information, or are there exceptions not covered in the provided clauses?
...

- Impact of Legal Compliance: How does the retention of information for legal compliance or record retention policies affect the overall obligations regarding confidential information?

Sources:
1. Clause 5 - Limits on Confidential Information
2. Clause 4.e - Obligations of Receiving Party
3. Clause 4.e - Obligations of Receiving Party
4. Clause 5 - Limits on Confidential Information
5. Clause 5.e - Limits on Confidential Information

A comparison of the outputs generated by GPT and PAKTON in response to the same ContractNLI input reveals substantial differences in terms of depth, structure, and legal interpretability. The output from GPT is concise and directly addresses the hypothesis by selecting the label *ENTAILMENT*. It supports this classification by referencing specific contractual clauses that align with the hypothesis and briefly justifies the decision. While this approach is effective for providing a quick assessment, the explanation remains relatively shallow. It does not explore potential exceptions, conditional obligations, or broader legal implications that may influence the interpretation.

On the other hand, the output from **PAKTON** exhibits a much more detailed and structured analysis. It decomposes the legal obligations regarding confidential information into separate categories: return, destruction, exceptions, and legal compliance. Each element is examined thoroughly, with references to specific clauses and relevant conditions. PAKTON also integrates legal reasoning, raises critical questions for further investigation, and acknowledges limitations or ambiguities in the contract's language. Additionally, PAKTON considers practical legal factors such as retention for statutory compliance and record-keeping policies—factors which are absent from GPT's output. These aspects contribute to a more nuanced and legally grounded evaluation of the hypothesis.

Overall, PAKTON's response reflects a deeper engagement with the legal content, demonstrating an interpretive capacity closer to that of a legal expert. GPT, while effective in recognizing textual entailment, remains more limited in its reasoning and scope of analysis.

E Computational Costs

The runtime of PAKTON primarily reflects the computational characteristics of the LLM used within the system, rather than the framework itself.

This is because, first, the invocation of the LLM constitutes the most compute-intensive component and represents the dominant runtime bottleneck, and second, PAKTON is explicitly designed to be model-agnostic. Consequently, we analyze computational cost primarily in terms of LLM calls, while accounting for auxiliary operations through a unified time-based cost model.

E.1 Cost Metrics

We employ two complementary cost metrics:

1. **LLM Call Count** (N^{LLM}): The number of distinct LLM inference calls, which directly determines the dominant computational cost.

2. **Total Runtime** (T_{total}): The end-to-end execution time, which includes LLM calls and auxiliary operations:

$$T_{total} = \sum_{i=1}^{N^{LLM}} t^i_{LLM} + T_{aux}$$

where t^i_{LLM} is the latency of the i-th LLM call and T_{aux} represents the cumulative time for auxiliary operations (embedding, reranking, OCR, etc.).

E.2 LLM Call Analysis

Let us denote:

- N_{turns}: number of conversation turns in the user interaction
- D_{int}: interrogation depth (number of iterative reasoning rounds)
- $\mathbb{I}[\cdot]$: indicator function (returns 1 if condition is true, 0 otherwise)

Archivist. The Archivist's LLM calls comprise conversation management and optional document parsing:

$$N^{LLM}_{Archivist} = \underbrace{(N_{turns} + 1)}_{Conversation} + \underbrace{\mathbb{I}[LLM\_parsing]}_{Document\_parsing} \quad (1)$$

The conversation component includes N_{turns} calls for the interactive dialogue with the user plus one call for final summarization and extraction of the query, context, and instructions. The document parsing term takes the value 1 when zero-shot LLM-based parsing is used rather than BERT-based parsing.

Interrogator. The Interrogator executes D_{int} reasoning rounds, each involving question generation, research, and report refinement:

$$N^{LLM}_{Interrogator} = D_{int} \times \underbrace{(1}_{Question} + \underbrace{N^{LLM}_{Researcher}}_{Research} + \underbrace{1)}_{Report} \quad (2)$$

Researcher. Each Researcher invocation requires query extraction, with optional natural language response:

$$N^{LLM}_{Researcher} = \underbrace{1}_{Query\ extraction} + \underbrace{\mathbb{I}[NL\_response]}_{Response\ generation} \quad (3)$$

When natural language response is disabled, the Researcher returns raw retrieved spans without additional LLM processing.

Total LLM Calls. Combining all components:

$$\begin{aligned} N^{LLM}_{Total} &= N^{LLM}_{Archivist} + N^{LLM}_{Interrogator} \\ &= (N_{turns} + 1) + \mathbb{I}[LLM\_parsing] \\ &\quad + D_{int} \times (2 + 1 + \mathbb{I}[NL\_response]) \quad (4) \end{aligned}$$

E.3 Auxiliary Operations

While LLM calls dominate the computational cost, auxiliary operations contribute to the total runtime:

Document Processing.

- T_{OCR}: Cost of Optical Character Recognition. This step relies on lightweight neural networks and the runtime is negligible compared to LLM inference.
- T_{parse}: Cost of Hierarchical BERT parsing
- T_{encode}: Cost of Document Encoding
Each resulting document chunk is transformed into an embedding vector using an embedding model. This cost is strongly influenced by the size of the embedding model and it is proportional to the number of chunks produced.

Retrieval Operations (per interrogation round).

- T_{query}: Cost of Query Embedding

- T_{search}: Cost of Vector Search
Nearest-neighbor lookup in the vector store, Complexity: $O(\log n \times d)$ where n is the number of stored vectors and d is the vector dimensionality

- T_{rerank}: Cost of Reranking

E.4 Unified Cost Model

To compare total computational costs, we express all operations in terms of equivalent runtime:

$$
\begin{aligned}
T_{total} \\
&= \underbrace{\sum_{i=1}^{N^{LLM}} t_{LLM}^{i}}_{\text{LLM inference}} + \underbrace{T_{OCR} + T_{parse} + T_{encode}}_{\text{Document processing}} \\
&\quad + \underbrace{D_{int} \times (T_{query} + T_{search} + T_{rerank})}_{\text{Retrieval operations}} \qquad (5)
\end{aligned}
$$

E.5 Model-Specific Considerations

PAKTON supports heterogeneous model configurations, enabling different LLMs to be assigned to different stages of the pipeline. For example, a larger and more capable model may be used for the final report generation, while a much smaller, cheaper model may suffice for query extraction. This flexibility allows for an effective trade-off between performance and computational cost.

E.6 Comparison with Baseline Models

When comparing computational costs against baseline methods (e.g., on the CONTRACTNLI benchmark), we observe a key distinction:

- **Baseline models:** Require a *single* LLM call, but the input is much larger, as it must include the entire contract.

- **PAKTON:** Issues multiple LLM calls and uses larger system prompts, but only processes relevant spans of the contract, reducing the number of input tokens per call.

This design leads to a more efficient use of computational resources, particularly for long contracts, while also enabling deeper reasoning through iterative interrogation.

F Experiments on LegalBenchRAG

F.1 Baselines

In the original LegalBenchRAG paper, the authors evaluated several retrieval configurations for in-document retrieval using the LegalBench benchmark. The configurations and their corresponding experimental setups are summarized below:

- **Method 1**: Naive fixed-size chunking with a window of 500 characters and no overlap; no reranker; embeddings generated using `text-embedding-3-large`. Results reported in Table 2 as "Naive".

- **Method 2**: Recursive Character Text Splitter (RCTS) with no overlap; no reranker; embeddings generated using `text-embedding-3-large`. Results reported in Table 2 as "RCTS".

- **Method 3**: Naive fixed-size chunking (500 characters, no overlap) with reranking using the Cohere reranker `rerank-english-v3.0`; embeddings from `text-embedding-3-large`. Results reported in Table 2 as "Naive + Cohere".

- **Method 4**: Recursive Character Text Splitter with no overlap and reranking using `rerank-english-v3.0`; embeddings from `text-embedding-3-large`. Results reported in Table 2 as "RCTS + Cohere".

Among these, Method 2 was selected by the authors as the preferred configuration. However, in this work we compare PAKTON against all four baseline configurations to demonstrate consistent performance improvements.

F.2 PAKTON Evaluation Strategy

To assess the retrieval capabilities of the **PAKTON** system on LegalBenchRAG, we evaluated the indexing and in-document retrieval functionality, which constitute the core components of the pipeline. In this setup:

- The **Archivist** component is responsible for document indexing.

- The **Researcher** performs in-document retrieval.

- Interactions through the **Interrogator** were bypassed to focus exclusively on retrieval performance.

Each document from the dataset was indexed by the *Archivist*, and the corresponding queries were directly submitted to the *Researcher*. The retrieved spans were evaluated using the LegalBenchRAG scoring methodology.

Configuration 1

Archivist:

- Primary strategy: structural parsing.
- Fallback: Recursive Character Text Splitter with 1000-character chunks and no overlap if structural parsing failed.
- Embedding model: `text-embedding-3-large`.

Researcher:

- Query optimization using gpt-4o and only in-document search as tool enabled.
- BM25 retrieves top-100 chunks with a similarity threshold of 0.6.
- Dense embedding retriever returns top-100 chunks with no similarity filtering using embeddings model `text-embedding-3-large`
- Reciprocal Rank Fusion with equal weights for both retrievers to rerank chunks, pick top-64 chunks.
- Reranker: `BAAI/bge-reranker-v2-m3`, producing a top-64 reranked final list of chunks.
- Strip structural information of the chunk and keep only the original span.

LLM Filtering:

- An additional post-reranking filtering stage is applied using `command-R` (Cohere), an open-source 35B parameter model specifically fine-tuned for Retrieval-Augmented Generation (RAG) applications.
- From the top-10 reranked chunks, the model identifies and extracts the most relevant sub-span(s) within each chunk, aiming to isolate highly precise evidence.
- This step is designed to enhance overall precision by focusing retrieval results on the most contextually pertinent portions of the content.

Results for this configuration are provided in Table 6. Two variants are compared: one with and one without the LLM filtering step. The results reported and compared in Table 2 correspond to Configuration 1, specifically the variant without LLM filtering.

Configuration 2

Same as Configuration 1 with the only change being the use of an alternative LLM-based reranker:

- Reranker: `AAI/bge-reranker-v2-minicpm-layerwise` (2.72B parameters) with a 28-layer cutoff.

Results for this configuration are provided in Table 8.

F.3 Discussion

Configuration 1 vs Configuration 2: While Configuration 1 serves as the primary setup for PAKTON, our evaluation indicates that Configuration 2 yields superior performance in terms of retrieval accuracy. However, this improvement comes at a cost: the second reranker in Configuration 2 is a large language model with 2.72 billion parameters, which introduces a significant latency overhead compared to the more lightweight setup in Configuration 1.

Considering the trade-off between reranking accuracy and computational efficiency, Configuration 1 represents the most suitable choice for practical deployment scenarios where speed is a critical factor. Nevertheless, for applications where performance is prioritized over inference time, Configuration 2 may be preferred to achieve more competitive results.

No LLM Filtering vs. LLM Filtering: In the legal domain, **high recall** is often of paramount importance, as omitting relevant spans can lead to incomplete or flawed legal reasoning. The *No LLM Filtering* configuration aligns better with this paradigm minimizing the risk of excluding potentially critical information. For this reason, it is selected as the primary setup in our evaluation.

Nonetheless, there are scenarios where **precision** is more desirable—particularly when mitigating hallucination risks or when users require concise, targeted evidence rather than exhaustive retrieval. To accommodate such use cases, we evaluate an additional post-reranking stage employing the *LLM Filtering* variant, which identifies and extracts the

		Precision @ k							Recall @ k						
Dataset	**LLM filtering**	1	2	4	8	16	32	64	1	2	4	8	16	32	64
PrivacyQA	without	19.94	16.84	11.44	8.62	7.38	6.42	6.08	13.34	22.43	32.67	43.39	61.65	82.30	89.42
	with	29.11	30.33	25.99	23.35	22.64	22.33	22.33	10.51	18.13	23.33	26.35	27.65	28.19	28.19
ContractNLI	without	33.02	30.34	17.33	9.98	5.87	4.68	4.52	53.14	67.47	80.06	89.71	95.50	99.56	99.82
	with	59.59	51.36	46.32	45.00	45.00	44.87	44.87	38.53	45.25	51.95	54.94	58.00	58.69	58.69
MAUD	without	25.47	17.45	10.51	7.24	5.08	3.18	1.85	23.99	30.09	34.49	46.42	59.74	74.96	82.80
	with	38.87	36.99	33.54	33.12	32.77	32.33	32.29	19.06	22.60	24.06	26.52	27.51	27.64	27.64
CUAD	without	11.02	8.83	6.81	4.72	2.78	2.07	1.62	16.52	24.76	33.34	46.67	59.53	77.08	86.23
	with	29.14	29.53	29.31	28.86	28.89	28.79	28.77	25.31	30.30	34.33	37.68	38.24	38.68	38.68
ALL	without	**22.34**	**18.37**	**11.52**	**7.63**	**5.26**	**4.08**	**3.52**	**26.77**	**36.32**	**45.26**	**56.66**	**69.17**	**83.50**	**89.58**
	with	**39.17**	**37.03**	**33.78**	**32.58**	**32.26**	**32.08**	**32.05**	**23.37**	**29.07**	**33.42**	**36.36**	**37.84**	**38.29**	**38.29**

Table 6: Performance comparison on different datasets for Precision and Recall at various k values for PAKTON's *Researcher* and *Archivist* under Configuration 1.

		Precision @ k							Recall @ k						
Dataset	**LLM filtering**	1	2	4	8	16	32	64	1	2	4	8	16	32	64
PrivacyQA	without	35.08	30.37	23.69	17.86	14.79	11.73	10.88	19.65	32.23	43.52	58.46	75.93	89.34	94.18
	with	33.51	32.98	28.01	26.16	25.61	25.43	25.39	20.36	31.65	37.62	42.49	43.94	46.89	46.89
ContractNLI	without	58.76	39.69	24.10	13.63	7.86	5.99	5.87	53.74	68.56	80.50	89.86	95.45	99.48	99.74
	with	63.73	54.12	49.40	48.01	47.90	47.81	47.81	54.10	61.30	68.17	71.60	72.90	73.41	73.41
MAUD	without	35.05	23.97	15.34	11.08	8.02	5.27	3.08	32.32	38.24	43.56	54.98	67.17	81.12	85.45
	with	41.58	36.01	32.22	31.34	30.72	30.33	30.29	36.90	41.18	44.18	47.77	48.55	48.81	48.81
CUAD	without	20.10	15.46	10.82	8.13	5.30	3.91	3.08	16.86	24.98	33.95	47.84	60.85	77.66	84.38
	with	35.75	30.93	29.38	29.11	28.83	28.82	28.81	30.99	37.81	43.14	47.04	48.85	49.28	49.28
ALL	without	37.24	27.37	18.49	12.68	8.99	6.73	5.73	30.64	41.00	50.38	62.79	74.85	86.90	90.94
	with	43.64	38.51	34.75	33.66	33.27	33.10	33.08	35.59	42.99	48.28	52.23	53.56	54.60	54.60

Table 7: Performance comparison across different datasets in terms of Precision and Recall at various k values, using PAKTON's *Researcher* and *Archivist* components under Configuration 1, based on span-based calculation of Precision and Recall.

		Precision @ k							Recall @ k						
Dataset	**LLM filtering**	1	2	4	8	16	32	64	1	2	4	8	16	32	64
PrivacyQA	without	18.64	17.66	14.13	10.39	7.51	6.34	6.01	17.00	26.39	39.99	62.73	79.24	94.90	97.35
	with	33.02	31.85	29.33	26.66	26.11	25.70	25.68	10.35	18.20	23.12	26.95	28.20	28.58	28.76
ContractNLI	without	**42.21**	**36.86**	**19.81**	**10.58**	**6.13**	**4.63**	**4.53**	**67.67**	**83.35**	**90.87**	**95.16**	**98.19**	**99.81**	**100.00**
	with	77.51	67.15	64.02	63.02	63.22	63.23	63.23	51.78	58.93	63.10	64.52	65.07	65.24	65.24
MAUD	without	17.77	11.35	9.41	8.32	5.84	3.50	1.92	17.06	19.94	30.57	49.94	68.82	80.10	86.51
	with	37.71	33.60	31.01	32.12	32.11	31.96	31.92	17.92	21.62	24.69	28.02	29.90	30.48	30.49
CUAD	without	2.61	5.10	5.72	4.18	2.84	1.84	1.29	4.38	16.11	34.09	53.00	69.13	80.65	86.20
	with	25.51	28.62	30.28	31.13	30.27	30.21	30.22	15.54	23.74	30.06	33.11	33.60	33.60	33.65
ALL	without	20.31	17.74	12.27	8.37	5.58	4.08	3.44	26.53	36.45	48.88	65.21	78.845	88.87	92.52
	with	43.44	40.30	38.66	38.23	37.93	37.78	37.76	23.90	30.62	35.24	38.15	39.19	39.48	39.54

Table 8: Performance comparison across different datasets in terms of Precision and Recall at various k values, using PAKTON's *Researcher* and *Archivist* under Configuration 2.

most relevant sub-spans within each of the top-10 reranked chunks, thereby filtering the retrieval output to focus on the most contextually pertinent segments.

As shown in Tables 6 and 8, this LLM Filtering variant significantly improves **precision**. However, this gain comes at the cost of **recall**, especially as the value of `top_k` increases, due to the stricter content selection. Based on these findings, we conclude that LLM Filtering is particularly advantageous for low `top_k` settings, where focused and precise evidence is preferred. In contrast, for higher `top_k` values, the unfiltered setup is more appropriate to maintain broader recall.

Character-Based vs. Span-Based Calculation of Precision and Recall: The LEGALBENCH-RAG paper adopts a character-based approach for computing precision and recall, and we follow the same protocol for our primary evaluation. However, we observe that this method may penalize retrieval strategies—particularly those targeting precision like the *LLM Filtering* variant—due to the fine-grained nature of the retrieved spans. Specifically, in many cases the retrieved content consists of sub-spans (often smaller than a sentence) that lie within the annotated answer span. Under the character-based metric, such partial matches are treated as incomplete, thereby reducing recall—even when the retrieved content is semantically relevant and informative.

To further investigate this effect, we introduce a complementary *span-based* evaluation. In this setting, a retrieved span is considered a *hit* if it overlaps with any ground truth span, and a *miss* otherwise. This binary overlap-based metric provides clearer insight into how often irrelevant spans are retrieved or relevant spans are entirely missed. The corresponding results are reported in Table 7 and should be interpreted in comparison with the character-based results in Table 6.

Overall, the span-based evaluation yields consistently higher values, with a particularly notable improvement in recall for the *LLM Filtering* variant. This suggests that character-level metrics may disproportionately penalize methods optimized for precision, potentially underestimating their effectiveness. It is also important to consider that different datasets contain varying numbers of ground truth spans per example, which can limit achievable recall at low `top_k` values. For reference, Table 9 presents the upper bounds on recall achievable under 100% precision.

Variation on Character Volume per chunk for different retrieval methods: Our analysis of precision and recall is conducted across varying `top_k` values. However, it is important to note that different retrieval methods return varying volumes

	Recall @ k						
Dataset	1	2	4	8	16	32	64
PrivacyQA	61.91	86.97	96.42	99.12	100.00	100.00	100.00
ContractNLI	82.16	98.69	100.00	100.00	100.00	100.00	100.00
MAUD	68.19	92.35	98.21	99.86	100.00	100.00	100.00
CUAD	82.14	94.30	98.44	100.00	100.00	100.00	100.00
ALL	**73.63**	**93.09**	**98.27**	**99.75**	**100.00**	**100.00**	**100.00**

Table 9: Recall @ k for perfect retrieval using ground-truth snippets. Note: Precision is 100% for all cases.

of text, even when the same number of chunks is retrieved. For instance, one method may retrieve significantly fewer characters on average per chunk compared to another, despite retrieving the same number of chunks. To better understand this disparity, Table 10 reports the total number of characters retrieved at various k values, the average answer length, and the average document length for each dataset.

This analysis allows us to assess the actual amount of information passed to the LLM across methods. Notably, the *LLM Filtering* variant consistently retrieves fewer characters than its unfiltered counterpart for the same number of chunks.

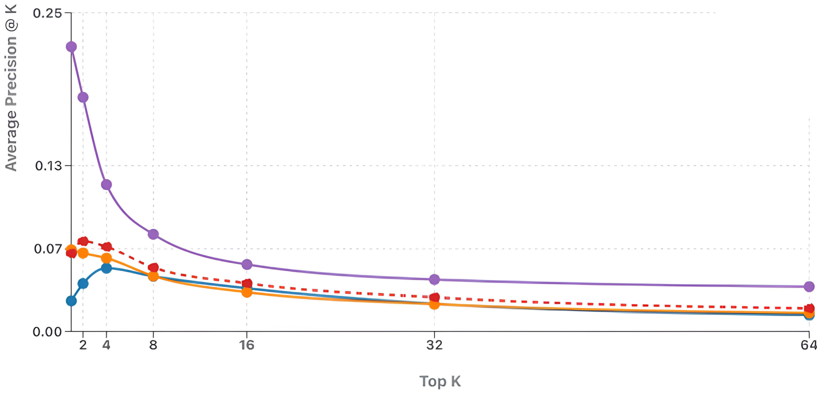
Moreover, this analysis provides insight into the degree of document compression achieved during retrieval. For example, in the MAUD dataset, our *span-based* Recall@64 reaches 85.45% while retrieving, on average, 56,523 characters—compared to an average document length of 353,718 characters—indicating an approximate 84% compression of the original document content. Similarly, in the ContractNLI dataset, the *LLM Filtering* variant achieves a Recall@32 of 74.41% using only 1,081 characters, which corresponds to nearly 90% information compression.

Plots. Figure 9 presents the evaluation plots for all four methods of LegalBenchRAG and PAKTON, including Precision@k, Recall@k, and Precision–Recall curves across all datasets. The results indicate that PAKTON consistently outperforms the other methods across all metrics.

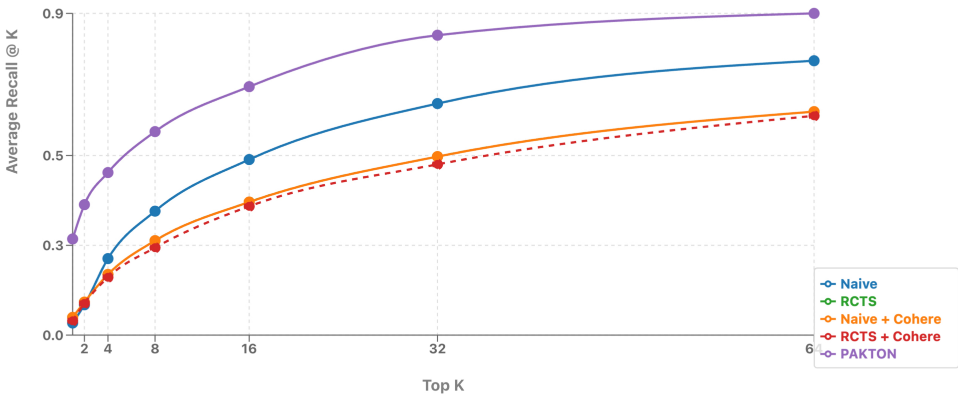
Conclusion: Taking into account the span-based evaluation metrics—which provide a more representative measure of retrieval quality—and the observed degree of information compression, the *Researcher* module demonstrates strong performance in the task of long-document retrieval within a practical deployment context.

Dataset	LLM Filtering	Number of Characters @ k 1	2	4	8	16	32	64	Avg. Lengths Answer	Document
PrivacyQA	without	682	1415	2985	5706	10341	17961	21080		
	with	207	403	711	1088	1362	1476	1523	1200	25266
ContractNLI	without	789	1371	2738	4869	8126	10774	11112		
	with	250	412	651	885	1036	1081	1081	438	10782
MAUD	without	908	1765	3542	7069	14248	28699	56523		
	with	301	477	678	906	1027	1059	1071	1259	353718
CUAD	without	771	1643	3496	7002	13694	24057	35750		
	with	250	383	527	749	844	896	971	600	69303

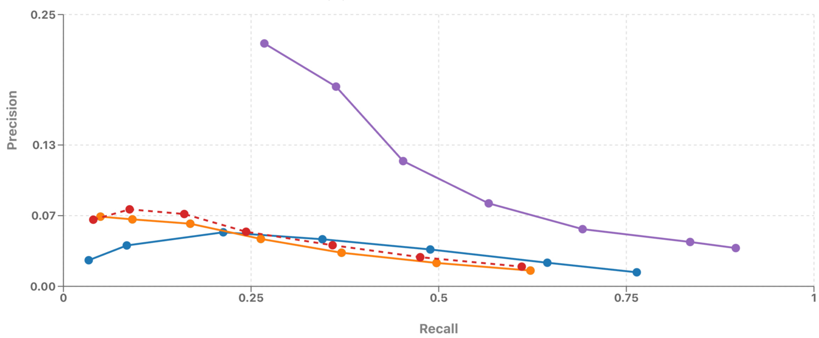
Table 10: Average number of characters retrieved @ k for each dataset for PAKTON configuration 1. Comparison with average length of ground truth (answer) and document lengths.



(a) Precision @ k across all datasets



(b) Recall @ k across all datasets



(c) Precision - Recall across all datasets

Figure 9: Precision and Recall values for different k across all datasets for all methods for LegalBenchRAG.

G Experiments on ContractNLI

G.1 Dataset Overview

We conduct all experiments on the test split of the ContractNLI dataset (Koreeda and Manning, 2021), which contains 2,091 samples. Each sample consists of a full non-disclosure agreement (NDA) as the *premise*, a legal statement as the *hypothesis*, and an *inference label* indicating whether the hypothesis is entailed, contradicted, or neutral with respect to the contract.

Data Access. We use the Hugging Face implementation of the dataset: `https://huggingface.co/datasets/kiddothe2b/contract-nli`

Predictions. For each experimental configuration, predictions and their associated data points are stored in structured `.json` files, which will be made publicly available upon publication.

G.2 Dataset Subsets

- **`contractnli_b`:** The full version of the dataset. It includes 7.19K samples in the training split, 1.04K in the validation split, and 2.09K in the test split. The full contract is used as the premise for each example.

- **`contractnli_a`:** A filtered version of `contractnli_b` in which only the minimal spans necessary to determine the correct label are retained as the premise, significantly reducing the input length. Experiments using this subset simulate ideal retrieval conditions and serve as an upper bound for the potential performance of a perfect RAG system.

G.3 Prompting Strategies

We experiment with the following prompting techniques:

- **`naive zero-shot (ZS)`:** No examples are given. A basic description of the label classes is provided, and the full contract is used as the premise. This serves as a solid baseline for performance.

- **`optimized zero-shot (opt. ZS)`:** Uses hardcoded explanations of the classes and improved prompt structure. Still uses the full contract as the premise. Explores the effect of manual prompt engineering.

- **`naive few-shot (FS)`:** Based on (Brown et al., 2020), builds on the optimized zero-shot format, but includes three random training examples (contract, hypothesis, and label) in the prompt. The full contract is used as the premise in both the examples and the current input. Demonstrates the effect of using cross-document retrieval.

- **`naive few-shot isolated spans (FS+Spans)`:** Similar to naive few-shot, but in the training examples, only the relevant spans (rather than the entire contract) are provided as the premise.

- **`naive few-shot isolated spans (same hypothesis) (FS+Spans+Hyp)`:** A refinement of the above method, in which few-shot examples are dynamically selected to match the hypothesis of the current sample, increasing semantic alignment. Shows the effect of using effective cross-document retrieval and chunking setting the upper boundary for the ideal cross-document RAG system.

- **`Chain of Thought (CoT)`:** Based on (Wei et al., 2022), extends the previous method by also including reasoning steps (i.e., rationales or justifications) in the answers of the few-shot examples, encouraging more explicit reasoning in the final output. Examines the impact of explicitly articulating the reasoning behind decisions.

G.4 Experimental Setup

Hardware: All local experiments were conducted on a server equipped with 4x NVIDIA A6000 GPUs, each with 48GB VRAM, using the AI daptiv infrastructure.

Quantization: We employed multiple quantization levels to evaluate performance under varying resource constraints. The specific techniques and configurations used will be made available in our GitHub repository upon publication.

Scale: In total, we ran 210 distinct experiments (each one for the whole test set of ContractNLI) across model, prompting, and input configurations.

G.5 Detailed Results

All the experiments and the corresponding results are shown in detail in Tables 11, 12, 13, 14, 15 and 16.

No	Model ID	Acc.	F1[W]	F1[E]	F1[C]	F1[N]	Quant.	Try	Prompting
0	Llama-3-8B-Instr	0.4868	0.4612	0.5677	0.0327	0.4514	no	1	ZS
1	Llama-3-8B-Instr	0.4883	0.4613	0.5639	0.0252	0.4576	no	2	ZS
2	Llama-3-8B-Instr	0.4969	0.4691	0.5758	0.0166	0.4651	no	3	ZS
3	Mistral-7B-Instr	0.5301	0.4804	0.4551	0.0261	0.6178	no	1	ZS
4	Mistral-7B-Instr	0.5340	0.4843	0.4627	0.0175	0.6206	no	2	ZS
5	Mistral-7B-Instr	0.5328	0.4839	0.4590	0.0342	0.6201	no	3	ZS
6	Saul-7B-Instr	0.4223	0.2933	0.0570	0.0910	0.5960	no	1	ZS
7	Saul-7B-Instr	0.4218	0.2960	0.0760	0.0542	0.5907	no	2	ZS
8	Saul-7B-Instr	0.4146	0.2809	0.0437	0.0588	0.5893	no	3	ZS
9	Mixtral-8x7B-Instr	0.5608	0.5659	0.6548	0.4087	0.5088	normal	1	ZS
10	Mixtral-8x7B-Instr	0.5605	0.5652	0.6551	0.4099	0.5065	normal	2	ZS
11	Mixtral-8x7B-Instr	0.5612	0.5661	0.6545	0.4105	0.5092	normal	3	ZS
12	Llama-3-8B-Instr	0.5151	0.4670	0.6331	0.0000	0.4026	no	1	opt. ZS
13	Llama-3-8B-Instr	0.5227	0.4738	0.6372	0.0000	0.4140	no	2	opt. ZS
14	Llama-3-8B-Instr	0.5261	0.4785	0.6384	0.0000	0.4237	no	3	opt. ZS
15	Llama-3-8B-Instr	0.5332	0.5013	0.6144	0.0090	0.5000	no	1	FS+Spans
16	Llama-3-8B-Instr	0.5232	0.4938	0.5977	0.0177	0.4986	no	2	FS+Spans
17	Llama-3-8B-Instr	0.5333	0.5024	0.6089	0.0089	0.5086	no	3	FS+Spans
18	SaulLM-54B-Instr	0.7021	0.6806	0.7726	0.1832	0.7032	lowest	1	ZS
19	SaulLM-54B-Instr	0.7001	0.6764	0.7724	0.1594	0.6995	lowest	2	ZS
20	SaulLM-54B-Instr	0.7040	0.6807	0.7731	0.1760	0.7046	lowest	3	ZS
21	Llama-3-70B-Instr	0.6241	0.5990	0.7313	0.4972	0.4819	normal	1	ZS
22	Llama-3-70B-Instr	0.6236	0.5987	0.7320	0.4943	0.4813	normal	2	ZS
23	Llama-3-70B-Instr	0.6208	0.5953	0.7297	0.4943	0.4759	normal	3	ZS
24	DS-R1-Dist.-Llama-70B	0.6054	0.6078	0.6552	0.4837	0.5874	normal	1	ZS
25	gpt-4o	0.6112	0.6367	0.7497	0.4119	0.5704	-	1	ZS
26	gpt-4o	0.6112	0.6356	0.75	0.4151	0.5667	-	2	ZS
27	gpt-4o	0.6141	0.6376	0.7474	0.4216	0.5725	-	3	ZS
28	gpt-4o	0.6275	0.6265	0.5957	0.4416	0.7046	-	1	opt. ZS
29	gpt-4o	0.6165	0.6169	0.5916	0.4455	0.6856	-	2	opt. ZS
30	gpt-4o	0.6227	0.6240	0.5973	0.4364	0.6983	-	3	opt. ZS
31	gpt-4o	0.6523	0.6693	0.7315	0.4577	0.6542	-	1	FS
32	gpt-4o	0.6590	0.6733	0.7297	0.4624	0.6642	-	2	FS
33	gpt-4o	0.6805	0.6941	0.7503	0.5	0.6813	-	3	FS
34	gpt-4o	0.6413	0.6507	0.6556	0.4533	0.6936	-	1	FS+Spans
35	gpt-4o	0.6538	0.6627	0.6724	0.4719	0.6989	-	2	FS+Spans
36	gpt-4o	0.6495	0.6587	0.6712	0.4656	0.6924	-	3	FS+Spans
37	gpt-4o	0.6906	0.7108	0.7919	0.4866	0.6784	-	1	ZS RAG
38	Llama3-8b-Instr	0.4776	0.4278	0.6029	0.0536	0.3314	-	1	ZS
39	Mistral-7b-Instr	0.527	0.4876	0.4863	0.0248	0.6018	-	1	ZS
40	Mistral-8x7b-Instr	0.5872	0.5948	0.656	0.4427	0.5664	-	1	ZS

Table 11: Baseline performance of models across multiple evaluation runs on the ContractNLI test set. (Part 1 of 3)

G.6 Execution Details of PAKTON

To simulate a *cross-document retrieval* setting, we indexed the training and validation splits of the ContractNLI dataset, preserving each example alongside its corresponding ground-truth label. Each contract chunk was embedded and stored in the most appropriate index, where grouping was determined by the combination of the hypothesis and its associated label. This approach emulates the core behavior of the *Archivist* module, which supports organizing textual segments into logically distinct indices—such as by contract type or clause category.

For example, all instances associated with the hypothesis "The Receiving Party shall not disclose the fact that the Agreement was agreed or negotiated" and labeled as *Neutral* were stored within a single index, while examples labeled as *Entailment* or *Contradiction* were assigned to their respective indices.

Each set of indices corresponding to the same hypothesis was interconnected into a composable graph using the `LlamaIndex` framework. These hypothesis-specific graphs were then integrated into a unified, higher-level composable graph. Every node—whether a graph or a leaf index—was annotated with a brief natural language description summarizing the content it encapsulated.

At inference time, this hierarchical structure was traversed recursively. At each level of the graph, a similarity comparison was conducted between the input query and the textual descriptions of child nodes to determine the most relevant subgraph to explore. This hierarchical traversal mechanism enables efficient prioritization of semantically aligned indices, thereby improving retrieval relevance.

No	Model ID	Acc.	F1[W]	F1[E]	F1[C]	F1[N]	Quant.	Try	Prompting
41	Llama3-70b-Instr	0.571	0.536	0.4493	0.3969	0.6628	-	1	ZS
42	Claude-3-Opus	0.7547	0.7676	0.8339	0.5083	0.7596	-	1	ZS
43	Claude-3-Opus	0.7461	0.7592	0.8279	0.4992	0.7489	-	2	ZS
44	Claude-3-Opus	0.7475	0.7608	0.8319	0.4970	0.7489	-	3	ZS
45	Claude-3.5-Sonnet	0.7944	0.8002	0.8757	0.5789	0.7732	-	1	ZS
46	Claude-3.5-Sonnet	0.7881	0.7949	0.8749	0.5609	0.7661	-	2	ZS
47	Claude-3.5-Sonnet	0.7924	0.7981	0.8764	0.5770	0.7679	-	3	ZS
48	Claude-3-Opus	0.7819	0.7891	0.8432	0.5622	0.7863	-	1	opt. ZS
49	Claude-3-Opus	0.7857	0.7935	0.8502	0.5559	0.7906	-	2	opt. ZS
50	Claude-3-Opus	0.7819	0.7886	0.8441	0.5606	0.7847	-	3	opt. ZS
51	Claude-3.5-Sonnet	0.7901	0.7960	0.8676	0.5710	0.7740	-	1	opt. ZS
52	Claude-3.5-Sonnet	0.7905	0.7967	0.8643	0.5728	0.7787	-	2	opt. ZS
53	Claude-3.5-Sonnet	0.7915	0.7975	0.8694	0.5681	0.7762	-	3	opt. ZS
54	Claude-3-Opus	0.7580	0.7593	0.8382	0.5628	0.7226	-	1	FS
55	Claude-3-Opus	0.7676	0.7680	0.8410	0.5945	0.732	-	2	FS
56	Claude-3-Opus	0.7542	0.7588	0.8308	0.5406	0.7348	-	3	FS
57	Claude-3.5-Sonnet	0.7709	0.7752	0.8542	0.5654	0.7415	-	1	FS
58	Claude-3.5-Sonnet	0.7838	0.7873	0.8645	0.5781	0.7556	-	2	FS
59	Claude-3.5-Sonnet	0.7786	0.7824	0.8576	0.5671	0.7543	-	3	FS
60	Claude-3-Opus	0.7862	0.7919	0.8489	0.5673	0.7855	-	1	FS+Spans
61	Claude-3-Opus	0.7891	0.7944	0.8457	0.5964	0.7876	-	2	FS+Spans
62	Claude-3-Opus	0.7786	0.7856	0.8393	0.5575	0.7836	-	3	FS+Spans
63	Claude-3.5-Sonnet	0.7977	0.8012	0.8650	0.6032	0.7811	-	1	FS+Spans
64	Claude-3.5-Sonnet	0.8015	0.8048	0.8717	0.6049	0.7819	-	2	FS+Spans
65	Claude-3.5-Sonnet	0.8006	0.8040	0.8668	0.6056	0.7850	-	3	FS+Spans
66	Claude-3-Opus	0.7752	0.7762	0.8437	0.5900	0.7492	-	1	FS+Spans+Hyp
67	Claude-3-Opus	0.7834	0.7839	0.8486	0.6097	0.7571	-	2	FS+Spans+Hyp
68	Claude-3-Opus	0.7666	0.7682	0.8362	0.5842	0.7403	-	3	FS+Spans+Hyp
69	Claude-3.5-Sonnet	0.8192	0.8243	0.8819	0.6392	0.8076	-	1	FS+Spans+Hyp
70	Claude-3.5-Sonnet	0.8149	0.8197	0.8792	0.6346	0.8010	-	2	FS+Spans+Hyp
71	Claude-3.5-Sonnet	0.8197	0.8246	0.8817	0.6279	0.8112	-	3	FS+Spans+Hyp
72	gemma-3-27b-it	0.7886	0.7860	0.8316	0.6348	0.7739	-	1	ZS
73	qwen2.5-72b-instruct	0.7728	0.7699	0.8248	0.5776	0.7579	-	1	ZS
74	qwen2.5-72b-instruct	0.7810	0.7754	0.8374	0.5013	0.7757	-	1	opt. ZS
75	qwen2.5-72b-instruct	0.7351	0.7241	0.8094	0.4920	0.6892	-	1	FS
76	qwen2.5-72b-instruct	0.7484	0.7432	0.8196	0.4378	0.7357	-	1	FS+Spans
77	qwen2.5-72b-instruct	0.7604	0.7505	0.8239	0.6236	0.7028	-	1	FS+Spans+Hyp
78	deepseek-chat	0.7881	0.7869	0.8496	0.6087	0.7631	-	1	ZS
79	deepseek-chat	0.7886	0.7874	0.8487	0.6139	0.7640	-	2	ZS

Table 12: Baseline performance of models across multiple evaluation runs on the ContractNLI test set (Part 2 of 3).

The *Researcher* module utilized this architecture as a cross-document retrieval system, returning the top-3 most relevant examples for a given query.

As for the *Researcher* module, we utilized **Configuration 1** (see Section F.2), with *No LLM filtering* and kept the top-10 reranked chunks to generate the response back to the *Interrogator*.

Regarding the interrogation process, we capped the maximum number of turns to five in order to maintain efficiency and avoid excessively long interaction sequences.

No	Model ID	Acc.	F1[W]	F1[E]	F1[C]	F1[N]	Quant.	Try	Prompting
80	deepseek-chat	0.7891	0.7883	0.8477	0.6126	0.7674	-	3	ZS
81	deepseek-chat	0.7872	0.7829	0.8503	0.6128	0.7521	-	1	opt. ZS
82	deepseek-chat	0.7886	0.7843	0.8535	0.6103	0.7526	-	2	opt. ZS
83	deepseek-chat	0.7891	0.7847	0.8519	0.6154	0.7539	-	3	opt. ZS
84	deepseek-chat	0.7681	0.7607	0.8346	0.6104	0.7182	-	1	FS
85	deepseek-chat	0.7743	0.7714	0.8377	0.5812	0.7465	-	1	FS+Spans
86	deepseek-chat	0.7963	0.7939	0.8612	0.6479	0.7573	-	1	FS+Spans+Hyp
87	deepseek-reasoner	0.7398	0.7455	0.8281	0.5229	0.7112	-	1	ZS
88	deepseek-reasoner	0.7394	0.7433	0.8305	0.5270	0.7024	-	2	ZS
89	claude-3.7-Sonnet	0.7704	0.7781	0.8633	0.5602	0.7398	-	1	ZS
90	Claude-3.7-Sonnet	0.7671	0.7733	0.8546	0.5586	0.7383	-	1	opt. ZS
91	Claude-3.7-Sonnet	0.7590	0.7602	0.8463	0.5607	0.7165	-	1	FS
92	Claude-3.7-Sonnet	0.7724	0.7766	0.8538	0.5805	0.7417	-	1	FS+Spans
93	Claude-3.7-Sonnet	0.8034	0.8068	0.8746	0.6309	0.7769	-	1	FS+Spans+Hyp
94	Llama3.1-70b-Instr	0.5758	0.5462	0.5014	0.2749	0.6604	-	1	opt. ZS
95	Llama3.1-70b-Instr	0.5849	0.5559	0.5208	0.2561	0.6667	-	2	opt. ZS
96	Llama3.1-70b-Instr	0.5725	0.5417	0.5046	0.2303	0.6574	-	3	opt. ZS
97	Llama3.1-70b-Instr	0.5729	0.5506	0.5421	0.2381	0.6358	-	1	FS
98	Llama3.3-70b-Instr	0.6767	0.6716	0.7366	0.5378	0.6346	-	1	ZS
99	Llama3.3-70b-Instr	0.7164	0.7123	0.7704	0.4745	0.7080	-	1	opt. ZS
100	Llama3.3-70b-Instr	0.6657	0.6565	0.7326	0.4431	0.6268	-	1	FS
101	Llama3.3-70b-Instr	0.6915	0.6879	0.7382	0.4244	0.6982	-	1	FS+Spans
102	Llama3.3-70b-Instr	0.7102	0.6974	0.7840	0.5294	0.6455	-	1	FS+Spans+Hyp
103	Mistral-8x7b-Instr	0.5423	0.5475	0.6445	0.4103	0.4770	-	1	ZS
104	Mistral-8x7b-Instr	0.6006	0.5931	0.6717	0.1592	0.6146	-	1	opt. ZS
105	Mistral-8x7b-Instr	0.6002	0.5804	0.6836	0.1931	0.5642	-	1	FS
106	Mistral-8x7b-Instr	0.6150	0.6017	0.6901	0.1951	0.6060	-	1	FS+Spans
107	Mistral-8x7b-Instr	0.5323	0.5302	0.6494	0.4152	0.4305	-	1	FS+Spans+Hyp
108	Mistral-7b-Instr	0.5364	0.5042	0.5279	0.0248	0.5951	-	1	ZS
109	Mistral-7b-Instr	0.5084	0.4821	0.6055	0.0000	0.4672	-	1	opt. ZS
110	Mistral-7b-Instr	0.5065	0.4702	0.6053	0.0082	0.4379	-	1	FS
111	Mistral-7b-Instr	0.4940	0.4576	0.6085	0.0076	0.4053	-	1	FS+Spans
112	Mistral-7b-Instr	0.4873	0.3660	0.6460	0.0090	0.1528	-	1	FS+Spans+Hyp

Table 13: Baseline performance of models across multiple evaluation runs on the ContractNLI test set (Part 3 of 3).

No	Model ID	Acc.	F1[W]	F1[E]	F1[C]	F1[N]	Quant.	Try	Prompting
0	gpt-4o	0.7328	0.7427	0.7575	0.5936	0.7630	-	1	ZS
1	gpt-4o	0.7248	0.7367	0.7598	0.5760	0.7516	-	2	ZS
2	gpt-4o	0.7303	0.7420	0.7629	0.5799	0.7593	-	3	ZS
3	gpt-4o	0.7102	0.7059	0.6685	0.6364	0.7584	-	1	opt. ZS
4	gpt-4o	0.7047	0.7008	0.6653	0.6220	0.7537	-	2	opt. ZS
5	gpt-4o	0.7012	0.6968	0.6611	0.604	0.7532	-	3	opt. ZS
6	gpt-4o	0.7353	0.7363	0.7305	0.6333	0.7658	-	1	FS
7	gpt-4o	0.7283	0.7290	0.7195	0.6341	0.7602	-	2	FS
8	gpt-4o	0.7343	0.7347	0.7222	0.6562	0.7652	-	3	FS
9	gpt-4o	0.7233	0.7255	0.7183	0.6174	0.7575	-	1	FS+Spans
10	gpt-4o	0.7328	0.7337	0.7321	0.6324	0.7587	-	2	FS+Spans
11	gpt-4o	0.7378	0.7382	0.7268	0.6534	0.7689	-	3	FS+Spans
12	gpt-4o	0.7835	0.7842	0.7908	0.7339	0.7894	-	1	CoT
13	gpt-4o	0.7835	0.7841	0.7911	0.7403	0.7874	-	2	CoT
14	gpt-4o	0.7760	0.7766	0.7824	0.7273	0.7824	-	3	CoT
15	Claude-3-Opus	0.8177	0.8194	0.8398	0.7174	0.8232	-	1	ZS
16	Claude-3-Opus	0.8217	0.8231	0.8457	0.7221	0.8246	-	2	ZS
17	Claude-3-Opus	0.8177	0.8197	0.8458	0.7024	0.8215	-	3	ZS
18	Claude-3.5-Sonnet	0.8237	0.8253	0.8450	0.7302	0.8283	-	1	ZS
19	Claude-3.5-Sonnet	0.8217	0.8233	0.8420	0.7302	0.8269	-	2	ZS
20	Claude-3.5-Sonnet	0.8237	0.8253	0.8438	0.7343	0.8285	-	3	ZS
21	Claude-3-Opus	0.8282	0.8296	0.8534	0.7164	0.8327	-	1	opt. ZS
22	Claude-3-Opus	0.8242	0.8255	0.8502	0.7110	0.8281	-	2	opt. ZS
23	Claude-3-Opus	0.8262	0.8273	0.8500	0.72	0.8301	-	3	opt. ZS
24	Claude-3.5-Sonnet	0.8378	0.8385	0.8593	0.7420	0.8408	-	1	opt. ZS
25	Claude-3.5-Sonnet	0.8312	0.8319	0.8513	0.7420	0.8339	-	2	opt. ZS
26	Claude-3.5-Sonnet	0.8408	0.8413	0.8598	0.7607	0.8421	-	3	opt. ZS
27	Claude-3-Opus	0.8378	0.8382	0.8629	0.7489	0.8349	-	1	FS
28	Claude-3-Opus	0.8338	0.8346	0.8602	0.7311	0.8337	-	2	FS
29	Claude-3-Opus	0.8302	0.8311	0.8556	0.7338	0.8298	-	3	FS
30	Claude-3.5-Sonnet	0.8398	0.8405	0.8604	0.7511	0.8419	-	1	FS
31	Claude-3.5-Sonnet	0.8438	0.8441	0.8639	0.7699	0.8420	-	2	FS
32	Claude-3.5-Sonnet	0.8413	0.8420	0.8607	0.7527	0.8446	-	3	FS
33	Claude-3-Opus	0.8348	0.8352	0.8633	0.7373	0.8306	-	1	FS+Spans
34	Claude-3-Opus	0.8343	0.8344	0.8619	0.7380	0.8301	-	2	FS+Spans
35	Claude-3-Opus	0.8277	0.8285	0.8514	0.7342	0.8281	-	3	FS+Spans
36	Claude-3.5-Sonnet	0.8348	0.8356	0.8544	0.7420	0.8391	-	1	FS+Spans
37	Claude-3.5-Sonnet	0.8413	0.8417	0.8582	0.7696	0.8424	-	2	FS+Spans
38	Claude-3.5-Sonnet	0.8428	0.8432	0.8621	0.7625	0.8435	-	3	FS+Spans
39	Claude-3-Opus	0.8378	0.8373	0.8629	0.7385	0.8354	-	1	FS+Spans+Hyp
40	Claude-3-Opus	0.8368	0.8365	0.8605	0.7466	0.8340	-	2	FS+Spans+Hyp

Table 14: Detailed baseline performance of models across multiple evaluation runs. Evaluations are conducted on isolated spans—subsections of the contract text directly relevant to the hypothesis—. (Part 1 of 3)

No	Model ID	Acc.	F1[W]	F1[E]	F1[C]	F1[N]	Quant.	Try	Prompting
41	Claude-3-Opus	0.8358	0.8355	0.8588	0.7426	0.8344	-	3	FS+Spans+Hyp
42	Claude-3.5-Sonnet	0.8498	0.8499	0.8679	0.7919	0.8459	-	1	FS+Spans+Hyp
43	Claude-3.5-Sonnet	0.8508	0.8509	0.8644	0.8080	0.8477	-	2	FS+Spans+Hyp
44	Claude-3.5-Sonnet	0.8569	0.8569	0.8747	0.7891	0.8553	-	3	FS+Spans+Hyp
45	gemma-3-27b-it	0.8247	0.8238	0.8536	0.7378	0.8148	-	1	ZS
46	gemma-3-27b-it	0.8207	0.8183	0.8512	0.7037	0.8128	-	1	opt. ZS
47	gemma-3-27b-it	0.8127	0.8104	0.8462	0.6945	0.8025	-	1	FS
48	gemma-3-27b-it	0.8137	0.8116	0.8469	0.6939	0.8047	-	1	FS+Spans
49	gemma-3-27b-it	0.8182	0.8160	0.8472	0.7103	0.8104	-	1	FS+Spans+Hyp
50	qwen2.5-72b-instruct	0.8277	0.8263	0.8559	0.7345	0.8189	-	1	ZS
51	qwen2.5-72b-instruct	0.8217	0.8189	0.8599	0.6865	0.8098	-	1	opt. ZS
52	qwen2.5-72b-instruct	0.8212	0.8189	0.8579	0.6873	0.8116	-	1	FS
53	qwen2.5-72b-instruct	0.8227	0.8202	0.8593	0.6859	0.8134	-	1	FS+Spans
54	Qwen2.5-72b-Instr.	0.8418	0.8401	0.8651	0.7775	0.8304	-	1	FS+Spans+Hyp
55	deepseek-chat	0.8368	0.8365	0.8559	0.7692	0.8333	-	1	ZS
56	deepseek-chat	0.8373	0.8370	0.8574	0.7682	0.8330	-	2	ZS
57	deepseek-chat	0.8368	0.8366	0.8562	0.7675	0.8335	-	3	ZS
58	deepseek-chat	0.8353	0.8348	0.8564	0.7606	0.8311	-	1	opt. ZS
59	deepseek-chat	0.8348	0.8343	0.8568	0.7589	0.8299	-	2	opt. ZS
60	deepseek-chat	0.8353	0.8348	0.8573	0.7606	0.8301	-	3	opt. ZS
61	deepseek-chat	0.8307	0.8302	0.8547	0.7506	0.8250	-	1	FS
62	deepseek-chat	0.8302	0.8296	0.8593	0.7354	0.8227	-	2	FS
63	deepseek-chat	0.8368	0.8361	0.8632	0.7455	0.8309	-	3	FS
64	deepseek-chat	0.8317	0.8313	0.8571	0.7392	0.8276	-	1	FS+Spans
65	deepseek-chat	0.8368	0.8364	0.8570	0.7630	0.8335	-	2	FS+Spans
66	deepseek-chat	0.8383	0.8379	0.8637	0.7489	0.8335	-	3	FS+Spans
67	deepseek-chat	0.8473	0.8467	0.8689	0.7930	0.8375	-	1	FS+Spans+Hyp
68	deepseek-reasoner	0.7860	0.7903	0.8061	0.6528	0.8070	-	1	ZS
69	deepseek-reasoner	0.7810	0.7854	0.8011	0.6568	0.8000	-	2	ZS
70	deepseek-reasoner	0.7805	0.7852	0.8034	0.6458	0.8000	-	3	ZS
71	Claude-3.7-Sonnet	0.8061	0.8074	0.8180	0.7388	0.8131	-	1	ZS
72	Claude-3.7-Sonnet	0.8071	0.8084	0.8187	0.7403	0.8143	-	2	ZS
73	Claude-3.7-Sonnet	0.8051	0.8064	0.8162	0.7403	0.8122	-	3	ZS
74	Claude-3.7-Sonnet	0.8222	0.8230	0.8371	0.7609	0.8237	-	1	opt. ZS
75	Claude-3.7-Sonnet	0.8212	0.8220	0.8362	0.7593	0.8228	-	2	opt. ZS
76	Claude-3.7-Sonnet	0.8217	0.8225	0.8367	0.7609	0.8230	-	3	opt. ZS
77	Claude-3.7-Sonnet	0.8312	0.8326	0.8510	0.7439	0.8353	-	1	FS
78	Claude-3.7-Sonnet	0.8317	0.8327	0.8508	0.7556	0.8330	-	2	FS
79	Claude-3.7-Sonnet	0.8242	0.8252	0.8418	0.7531	0.8258	-	3	FS

Table 15: Detailed baseline performance of models across multiple evaluation runs. Evaluations are conducted on isolated spans—subsections of the contract text directly relevant to the hypothesis—. (Part 2 of 3)

No	Model ID	Acc.	F1[W]	F1[E]	F1[C]	F1[N]	Quant.	Try	Prompting
80	Claude-3.7-Sonnet	0.8277	0.8288	0.8429	0.7592	0.8313	-	1	FS+Spans
81	claude-3.7-Sonnet	0.8428	0.8431	0.8601	0.7617	0.8456	-	1	FS+Spans+Hyp
82	Llama3.1-70b-Instr	0.7328	0.7321	0.7321	0.6650	0.7477	-	1	ZS
83	Llama3.1-70b-Instr	0.7332	0.7325	0.7346	0.6505	0.7496	-	2	ZS
84	Llama3.1-70b-Instr	0.7411	0.7402	0.7400	0.6650	0.7578	-	3	ZS
85	Llama3.1-70b-Instr	0.7685	0.7655	0.8045	0.5699	0.7731	-	1	opt. ZS
86	Llama3.1-70b-Instr	0.7614	0.7585	0.7934	0.5675	0.7690	-	2	opt. ZS
87	Llama3.1-70b-Instr	0.7609	0.7588	0.7895	0.5898	0.7683	-	3	opt. ZS
88	Llama3.1-70b-Instr	0.7248	0.7226	0.7372	0.5707	0.7437	-	1	FS
89	Llama3.1-70b-Instr	0.7348	0.7325	0.7517	0.5660	0.7525	-	2	FS
90	Llama3.1-70b-Instr	0.7212	0.7196	0.7306	0.5885	0.7394	-	3	FS
91	Llama3.1-70b-Instr	0.7348	0.7321	0.7380	0.5962	0.7579	-	1	FS+Spans
92	Llama3.1-70b-Instr	0.7313	0.7288	0.7449	0.5532	0.7541	-	2	FS+Spans
93	Llama3.1-70b-Instr	0.7117	0.7087	0.7081	0.5789	0.7394	-	3	FS+Spans
94	Llama3.1-70b-Instr	0.8152	0.8132	0.8441	0.6818	0.8138	-	1	FS+Spans+Hyp
95	Llama3.1-70b-Instr	0.8157	0.8138	0.8437	0.6937	0.8126	-	2	FS+Spans+Hyp
96	Llama3.1-70b-Instr	0.8192	0.8175	0.8478	0.6948	0.8167	-	3	FS+Spans+Hyp
97	Llama3.3-70b-Instr	0.7845	0.7849	0.8092	0.7089	0.7790	-	1	ZS
98	Llama3.3-70b-Instr	0.8117	0.8103	0.8389	0.7136	0.8051	-	1	opt. ZS
99	Llama3.3-70b-Instr	0.7941	0.7936	0.8177	0.6929	0.7935	-	1	FS
100	Llama3.3-70b-Instr	0.8021	0.8012	0.8303	0.6998	0.7966	-	1	FS+Spans
101	Llama3.3-70b-Instr	0.8403	0.8391	0.8651	0.7536	0.8337	-	1	FS+Spans+Hyp
102	Mistral-8x7b-Instr	0.7177	0.7192	0.7610	0.6478	0.6951	-	1	ZS
103	Mistral-8x7b-Instr	0.7515	0.7425	0.8117	0.5120	0.7290	-	1	opt. ZS
104	Mistral-8x7b-Instr	0.7574	0.7516	0.8065	0.5934	0.7350	-	1	FS
105	Mistral-8x7b-Instr	0.7438	0.7361	0.8008	0.5272	0.7217	-	1	FS+Spans
106	Mistral-8x7b-Instr	0.7514	0.7474	0.8106	0.6332	0.7125	-	1	FS+Spans+Hyp
107	Mistral-7b-Instr	0.6224	0.6052	0.6083	0.2867	0.6744	-	1	ZS
108	Mistral-7b-Instr	0.6217	0.5932	0.6828	0.0437	0.6339	-	1	opt. ZS

Table 16: Detailed baseline performance of models across multiple evaluation runs. Evaluations are conducted on isolated spans—subsections of the contract text directly relevant to the hypothesis—. (Part 3 of 3)