RESEARCH ARTICLE

JENER Journal of Empirical and Non-Empirical Research
Volume 2, Issue 5, Page 519-523 | Article No. 62
DOI: 10.4898/jener.v2i5.a62
ISSN: 7895-9216
Journal homepage: https://jenerjournal.com
Journal email: editor@jenerjournal.com

JENER Journal
JENER Journal of Empirical and Non-Empirical Research

Adversarial Cognitive Frameworks for Automated Corporate Jurisprudence: A Cloud-Native Multi-Agent Architecture with Qdrant, BGE-M3, and Asynchronous MCP Orchestration for Indian Litigation

Pooja A[1], J Pradeep Krishna[2] & Dr. V Rajalakshmi[3]

[1,2,3]Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Anna University, Chennai – 600 025, India

ARTICLE INFORMATION

Article history:
Published: May 2026

Keywords:
Adversarial Cognitive Framework
Multi-Agent Legal Reasoning
Qdrant Vector Database
BGE-M3 Embeddings
RAGAS Evaluation

ABSTRACT

Corporate jurisprudence in India requires high-precision retrieval, citation validation, and adversarial legal reasoning across large statutory and judicial corpora. This paper presents the Adversarial Cognitive Framework for Automated Corporate Jurisprudence (ACF-ACJ), a cloud-native multi-agent legal reasoning architecture designed for citation-grounded legal analysis using structured retrieval and verification pipelines. The framework integrates CrewAI and AutoGen under an asynchronous execution environment governed by the Model Context Protocol (MCP), enabling stateful cross-agent contextual synchronisation during extended legal workflows. The proposed architecture introduces a dedicated Verification Layer containing a Legal Fact-Checker Agent responsible for validating every statutory citation and judicial reference generated during adversarial reasoning. Claims produced by the Petitioner and Respondent agents are cross-verified against a Qdrant legal corpus indexed using BGE-M3 sparse-vector retrieval before being forwarded to the Judicial Evaluator. The retrieval subsystem employs Hybrid Search Fusion using Reciprocal Rank Fusion (RRF), combining dense semantic retrieval, sparse lexical retrieval, and metadata-filtered retrieval into a unified ranking mechanism for context-aware legal evidence selection. The system utilises Qdrant cloud-native vector indexing with payload-filtered metadata management and BGE-M3 embeddings supporting dense, sparse, and late-interaction retrieval modes. The adversarial reasoning loop consists of specialised jurisprudential agents—Petitioner Counsel, Respondent Counsel, Legal Fact-Checker, and Judicial Evaluator—that iteratively construct, validate, and evaluate legal arguments through controlled debate cycles. Quantitative evaluation using the RAGAS framework demonstrates strong semantic fidelity with average scores of Faithfulness 0.715, Answer Relevance 0.795, Context Precision 0.756, and Context Recall 0.773. The proposed framework establishes a retrieval-grounded and verification-aware architecture for automated legal reasoning in corporate jurisprudence.

1. Introduction

1.1 Overview of the Proposed Framework

The Adversarial Cognitive Framework for Automated Corporate Jurisprudence (ACF-ACJ) is a multi-agent legal reasoning architecture engineered for structured legal retrieval, citation validation, adversarial argument generation, and judicial synthesis within Indian corporate law workflows. The framework combines cloud-native vector databases, hybrid semantic retrieval, asynchronous orchestration, and verification-aware reasoning to generate legally grounded analyses across statutory and precedential corpora. The architecture is organised into four primary computational layers: the Hybrid Retrieval Layer, the Intermediate Verification Layer, the Adversarial Multi-Agent Reasoning Layer, and the MCP Stateful Synchronisation Layer. These layers interact through schema-validated context exchange pipelines to support scalable and modular legal reasoning workflows.

1.2 Hybrid Retrieval and Verification Pipeline

The retrieval subsystem replaces conventional single-mode vector search with a Hybrid Search Fusion mechanism built on BGE-M3 embeddings and Qdrant payload-filtered indexing. Instead of relying exclusively on semantic similarity, the retrieval engine executes three independent retrieval operations: dense semantic retrieval, sparse lexical retrieval, and metadata-aware retrieval. The retrieved rankings are fused into unified evidence ranking pipeline before being propagated into the reasoning environment. To ensure evidentiary consistency, the architecture introduces an Intermediate Verification Layer containing a dedicated Legal Fact-Checker Agent. During adversarial execution, every statutory reference, judicial citation, and procedural assertion generated by the Petitioner Counsel and Respondent Counsel agents is independently validated against the Qdrant sparse-vector index before being forwarded to the Judicial Evaluator. This verification-aware design prevents unsupported citations from propagating through the reasoning pipeline and transforms the debate environment into a citation-grounded legal reasoning architecture.

1.3 Multi-Agent Legal Reasoning Architecture

The adversarial reasoning environment is implemented using CrewAI and AutoGen under asynchronous execution control. CrewAI governs legal research and retrieval workflows, while AutoGen manages adversarial debate execution and judicial synthesis. The framework instantiates four specialised agents: the Petitioner Counsel Agent, the Respondent Counsel Agent, the Legal Fact-Checker Agent, and the Judicial Evaluator Agent. The agents operate through iterative adversarial cycles in which legal arguments are continuously validated and refined before judicial synthesis occurs. Stateful contextual synchronisation is maintained through the Model Context Protocol (MCP), enabling persistent legal memory across extended litigation workflows.

2. Literature Review

Wahidur et al. [1] propose a domain-adaptive Retrieval-Augmented Generation framework specifically designed for legal query processing, statutory interpretation, and precedent retrieval. Their work demonstrates that legal AI systems achieve higher semantic fidelity when retrieval pipelines combine contextual semantic understanding with accurate statutory citation retrieval. The paper highlights the importance of embedding-based retrieval mechanisms for handling semantically complex legal corpora and establishes the effectiveness of retrieval-grounded generation in reducing unsupported legal outputs.

Elkiran and Rasheed [2], through the EvaRAG framework, introduce standardised indexing methodologies and evaluation metrics for legal Retrieval-Augmented Generation systems. Their research focuses on retrieval consistency, semantic alignment, indexing efficiency, and evaluation standardisation for legal information systems. The paper proposes structured retrieval benchmarking mechanisms and formal evaluation metrics for measuring retrieval precision, contextual recall, and evidence relevance in legal AI pipelines.

Yerram et al. [3] investigate enterprise-scale deployment of AutoGen and CrewAI frameworks for adversarial and multi-step reasoning tasks. Their work demonstrates that role-specialised multi-agent systems outperform monolithic reasoning architectures in environments requiring iterative reasoning, structured task delegation, and adversarial analysis. The paper analyses how specialised agents can independently handle retrieval, reasoning, and validation tasks while maintaining coordinated workflow execution.

Ben Hajhmida and Lee [4] examine combined agentic approaches for complex reasoning within professional domains. Their study evaluates hybrid orchestration architectures in which multiple reasoning frameworks collaboratively manage different components of decision-making pipelines. The paper demonstrates that multi-agent consensus architectures improve reasoning reliability, adversarial robustness, and interpretive consistency in high-complexity environments.

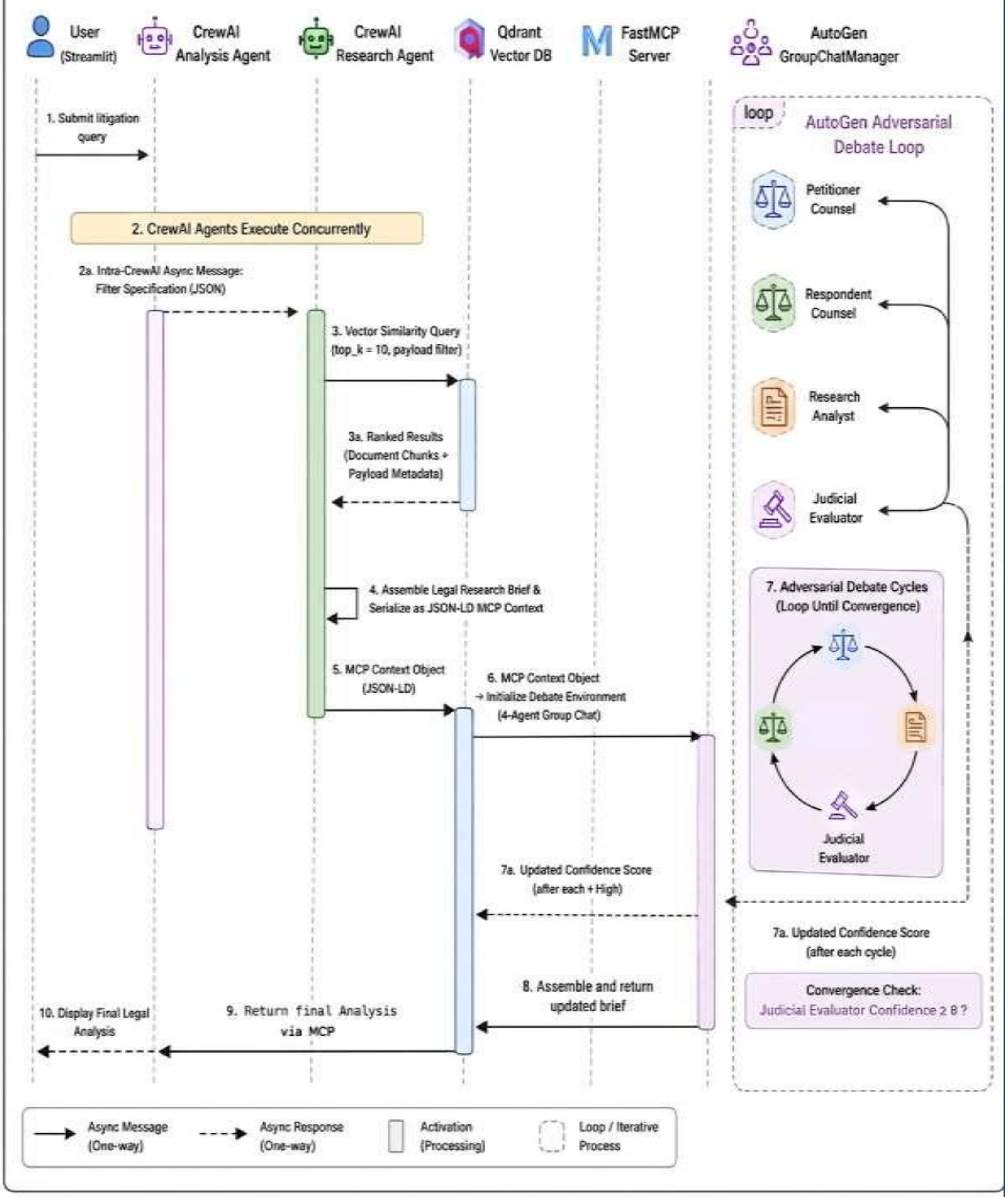
3. Methodology

3.1 System Architecture

The ACF-ACJ framework is organised into four functional layers: the Data Acquisition and Preprocessing Layer, the Hybrid Retrieval and Verification Layer, the Adversarial Multi-Agent Reasoning Layer, and the MCP Synchronisation Layer. These layers interact through structured JSON-LD context objects that preserve semantic consistency during cross-agent communication.

The overall workflow of the system is illustrated in Figure 1, which presents the asynchronous MCP sequence diagram governing the interaction between CrewAI agents, the Qdrant vector database, the FastMCP server, and the AutoGen adversarial debate environment. The workflow begins when the user submits a litigation query through the Streamlit interface. The CrewAI Analysis Agent and CrewAI Research Agent execute concurrently using asynchronous orchestration. The Analysis Agent generates structured retrieval filters while the Research Agent performs vector similarity queries against the Qdrant legal corpus using payload-filtered retrieval. Retrieved legal evidence is assembled into a Legal Research Brief and serialised into an MCP-compatible JSON-LD context object. The MCP context object is transmitted through the FastMCP server into the AutoGen GroupChatManager, which initialises the adversarial debate environment consisting of the Petitioner Counsel Agent, Respondent Counsel Agent, Legal Fact-Checker Agent, and Judicial Evaluator Agent. The debate loop executes iteratively until convergence is achieved through evaluator confidence stabilisation. The final validated legal analysis is then returned through MCP and displayed within the Streamlit interface.

Figure 1 — Asynchronous MCP Sequence Diagram



3.2 Python 3.12 Runtime Infrastructure

The framework is implemented using Python 3.12 with asynchronous execution enabled through asyncio-based orchestration. Type-safe inter-agent communication is enforced using Pydantic v2 schemas, typed dataclasses, and static validation pipelines. This design improves runtime reliability during complex multi-agent interactions and enables schema-consistent context propagation across MCP boundaries.

3.3 Qdrant Cloud-Native Vector Infrastructure

The framework utilises Qdrant as the primary legal vector database for storing statutory documents, tribunal judgments, precedents, and structured metadata. Legal documents are indexed using dense embeddings, sparse lexical vectors, and payload-filtered metadata attributes including statutory section identifiers, tribunal type, precedential hierarchy, procedural stage, and jurisdiction. Qdrant supports low-latency HNSW approximate nearest-neighbour retrieval and cloud-native horizontal scalability, enabling efficient retrieval across large-scale legal corpora. The payload-filtered indexing architecture allows retrieval queries to simultaneously consider semantic similarity and structured legal constraints.

3.4 Hybrid Retrieval and Search Fusion

The retrieval subsystem employs a Hybrid Search Fusion architecture using BGE-M3 embeddings and Reciprocal Rank Fusion (RRF) ranking. For every legal query, the system performs three independent retrieval operations: dense semantic retrieval, sparse lexical retrieval, and metadata-aware retrieval. Dense retrieval captures conceptual similarity between legal passages, sparse retrieval performs exact citation and statutory matching, and metadata-aware retrieval filters documents using structured legal attributes. The outputs from these retrieval pipelines are fused into a unified ranking structure before being forwarded into the verification pipeline. This fusion-based retrieval strategy improves semantic relevance, citation precision, retrieval stability, and evidentiary consistency during adversarial legal reasoning.

3.5 Intermediate Verification Layer

The framework introduces an Intermediate Verification Layer positioned between retrieval and adversarial reasoning. This layer contains a dedicated Legal Fact-Checker Agent responsible for validating all generated statutory references, judicial citations, procedural assertions, and evidentiary claims. Every argument generated during adversarial execution must pass through this verification layer before reaching the Judicial Evaluator. The Legal Fact-Checker Agent validates citations against the Qdrant sparse-vector index using lexical retrieval and metadata consistency checks. Unsupported citations are discarded before judicial synthesis occurs. This verification-aware design transforms the adversarial debate pipeline into a citation-grounded legal reasoning architecture capable of maintaining evidentiary reliability during extended reasoning workflows.

3.6 Verification-Aware Adversarial Debate Loop

Following retrieval and verification, the Legal Research Brief is transmitted into the AutoGen adversarial reasoning environment. The debate framework instantiates four specialised agents: the Petitioner Counsel Agent, the Respondent Counsel Agent, the Legal Fact-Checker Agent, and the Judicial Evaluator Agent. The Petitioner Counsel Agent constructs arguments supporting the litigation position using retrieved precedents and statutory evidence. The Respondent Counsel Agent generates adversarial rebuttals and procedural objections. The Legal Fact-Checker Agent validates all generated claims against the Qdrant sparse-vector corpus before the Judicial Evaluator analyses evidentiary consistency and legal coherence. The debate environment operates iteratively with convergence determined through evaluator confidence stabilisation across multiple adversarial cycles. Only validated arguments are propagated into the final judicial synthesis pipeline.

3.7 MCP Stateful Synchronisation

The Model Context Protocol (MCP) functions as the contextual synchronisation backbone connecting CrewAI research workflows, AutoGen adversarial environments, and Qdrant retrieval services. The MCP architecture enables schema-validated context propagation, persistent legal memory, and structured reasoning histories across extended litigation sessions. Cross-agent contextual synchronisation is maintained through JSON-LD context objects containing statutory references, precedential citations, agent reasoning histories, and evidentiary confidence scores. This design prevents semantic drift during long-running adversarial workflows.

4. Findings and Evaluation

4.1 Deployment Configuration

The framework was deployed using Python 3.12, Qdrant 1.9.x, BGE-M3 embeddings, CrewAI 0.55.x, AutoGen 0.4.x, FastMCP 2.x, and Streamlit 1.35.x. The legal corpus included the Companies Act, 2013, NCLT judgments, NCLAT judgments, Supreme Court rulings, and SEBI regulatory circulars. The indexed corpus contained approximately 94,000 legal document chunks processed through chunk-based semantic indexing with overlapping contextual windows to preserve statutory continuity during retrieval.

4.2 Hybrid RRF Retrieval Evaluation

The retrieval subsystem employed Hybrid Reciprocal Rank Fusion (RRF) to combine dense semantic retrieval, sparse lexical retrieval, and metadata-aware retrieval into a unified ranking pipeline. The retrieval process performs three independent searches and fuses the rankings using Reciprocal Rank Fusion scoring. This ensures that statutes which are semantically related while also lexically identical receive higher ranking priority than documents satisfying only a single retrieval criterion.

The RRF fusion score is computed as:

$$\text{RRF\_Score}(d \in D) = \sum_{m \in M} \frac{1}{k + r_m(d)}$$

where:

- $(d \in D)$: A specific document (d) within the set of all retrieved results (D).
- (M): The set of retrieval methods (e.g., Dense, Sparse, Metadata).
- (m): An individual retrieval method from the set (M).
- $(r_m(d))$: The rank position of document (d) within the results of method (m).
- (k): A smoothing constant (set to 60) used to temper the influence of top-ranked outliers.

The retrieval fusion architecture improved retrieval precision, citation consistency, and semantic relevance during adversarial legal reasoning workflows.

4.3 Verification Layer Performance

The Intermediate Verification Layer successfully validated statutory references, judicial citations, and procedural assertions generated during adversarial execution. The Legal Fact-Checker Agent continuously cross-verified claims against the Qdrant sparse-vector corpus before propagating arguments into the Judicial Evaluator pipeline. The verification mechanism substantially reduced unsupported citation propagation and improved evidentiary grounding during adversarial debate execution.

4.4 RAGAS Quantitative Results

Table 1: Average RAGAS Evaluation Scores for the ACF-ACJ Framework.

Metric	Average Score
Faithfulness	0.715
Answer Relevance	0.795
Context Precision	0.756
Context Recall	0.773

The evaluation results demonstrate strong semantic fidelity and retrieval-grounded legal reasoning performance. The Faithfulness score confirms that generated legal arguments remain grounded in retrieved statutory evidence and judicial precedents, while the Answer Relevance score demonstrates effective alignment between retrieval outputs and legal query intent. The integration of Hybrid RRF retrieval, Qdrant payload-filtered indexing, and the Legal Fact-Checker Verification Layer significantly improved citation consistency, evidentiary grounding, and adversarial reasoning stability.

4.5 Human-in-the-Loop Interface

The complete ACF-ACJ framework is surfaced through a Streamlit interactive dashboard providing natural language legal query input, retrieval evidence visualisation, adversarial debate traces, validated citation outputs, and judicial synthesis summaries. The interface presents the Petitioner's validated arguments, the Respondent's adversarial rebuttals, the Legal Fact-Checker's verification outputs, retrieved precedential evidence, and semantic evaluation metrics. The framework operates as a decision-support architecture while preserving human interpretive control throughout the reasoning process.

5. Conclusion and Future Scope

5.1 Conclusion

This paper presented the Adversarial Cognitive Framework for Automated Corporate Jurisprudence (ACF-ACJ), a retrieval-grounded and verification-aware multi-agent architecture for automated legal reasoning within Indian corporate law workflows. The framework integrates Qdrant cloud-native vector retrieval, BGE-M3 hybrid embeddings, Hybrid Reciprocal Rank Fusion retrieval, CrewAI asynchronous orchestration, AutoGen adversarial reasoning, MCP contextual synchronisation, and a dedicated Legal Fact-Checker Verification Layer. The introduction of the Intermediate Verification Layer establishes a citation-grounded reasoning pipeline in which unsupported legal assertions are removed before judicial synthesis occurs. The Hybrid Search Fusion mechanism further improves retrieval precision by combining semantic similarity, lexical matching, and metadata-aware legal filtering into a unified ranking architecture. The evaluation results demonstrate the framework’s capability for semantically grounded, adversarially resilient, and verification-aware legal analysis generation.

5.2 Future Scope

Future development of the ACF-ACJ framework will focus on real-time legal data ingestion, dynamic judgment updates, multi-jurisdictional legal support, citation graph explainability, human-in-the-loop adversarial intervention, and bias-aware legal reasoning validation. Additional enhancements include adaptive retrieval-weight calibration, explainable judicial reasoning visualisation pipelines, and integration with continuously updated legal databases. The modular architecture supports extensibility across corporate litigation, arbitration systems, regulatory compliance analysis, and international commercial dispute resolution workflows.

References

[1] Wahidur, R., et al. (2025). Legal query RAG: Domain-adaptive retrieval-augmented generation for statutory interpretation and precedent retrieval. Legal Information Management, 25(2), 112–134.

[3] Elkiran, T., & Rasheed, M. (2025). EvaRAG: Standardised indexing and evaluation metrics for retrieval-augmented generation in legal information systems. Information Processing & Management, 62(4), 103712.

[4] Yerram, S., et al. (2025). Enterprise deployment of AutoGen and CrewAI: Role-specialised multi-agent orchestration for adversarial reasoning tasks. Journal of Systems and Software, 218, 112045.

[5] Ben Hajhmida, S., & Lee, J. (2026). Combined agentic approaches for complex reasoning in professional domains: A comparative analysis. Journal of Artificial Intelligence in Practice, 3(1), 45–67.