



JOURNAL ON COMMUNICATIONS

ISSN:1000-436X

Scopus®

REGISTERED

www.jocs.review

JuriSim: A Legal Precedent-Based Simulation Framework with Domain-Specific Ko-SBERT Fine-Tuning, Multi-Stage RAG, and Reliability Assessment

Jung Uijun[1], Park Eunwoo[2], Lee Seungbum[3], Oh Wonwoo[4], Seungjae Lee[5*]

[1,2,3,4,5]*Department of Computer Engineering, Sunmoon University, Republic of Korea*

Abstract— Access to legal information remains challenging for the general public due to the complexity of legal terminology, the difficulty of identifying relevant precedents, and the high cost of professional legal services. This paper presents JuriSim, a reliability-aware legal simulation framework that integrates domain-specific sentence embedding, multi-stage Retrieval-Augmented Generation (RAG), and quantitative reliability assessment. A dataset comprising 11,373 legal precedents collected from four sources, including the National Law Information API, Hugging Face, and AI Hub, was constructed and segmented into 28,625 vector chunks using a 512-token chunking strategy. To improve semantic retrieval in the Korean legal domain, a Ko-SBERT model was fine-tuned using a Hard Positive Triplet Loss strategy. In particular, a direct cosine similarity computation method was introduced to identify hard positive samples within the same precedent, addressing the 81% fallback rate encountered when relying on ChromaDB-based similarity searches during training. The proposed fine-tuning approach improved the reported cosine-similarity-based retrieval accuracy from 4.8% to 76.2%. The proposed system employs a multi-stage RAG pipeline. First, a Gemini-based language normalization module transforms colloquial user queries into legal-domain terminology. Second, semantically relevant precedents and statutory provisions are retrieved using the fine-tuned embedding model and cosine similarity. Third, a weighted reliability assessment model evaluates the retrieved evidence based on four indicators: semantic similarity, category matching, evidence coverage, and temporal relevance. Finally, a second LLM invocation generates structured legal analysis grounded in the retrieved documents. The reliability score is calculated using a weighted combination of the four indicators, with expert consultation recommended when the score falls below 50 on a 100-point scale. The deployed system was evaluated across three legal categories: labor, leasing, and consumer disputes. The observed retrieval similarity scores were 0.7783, 0.9541, and 0.9367, respectively. Furthermore, an MD5 hash-based cache with a 500-entry capacity reduced average response time by approximately 62%. These results demonstrate the practical feasibility of integrating domain-adaptive legal precedent retrieval, multi-stage RAG, and reliability visualization into an accessible legal simulation service.

Keywords— Legal Precedent Retrieval, Retrieval-Augmented Generation (RAG), Ko-SBERT Fine-Tuning, Hard Positive Mining, Triplet Loss, Reliability Assessment, Legal Simulation, Semantic Search

1. Introduction

Access to legal information and professional legal services remains a significant challenge for individuals involved in civil disputes. Legal documents are typically written using specialized terminology and complex expressions, making it difficult for non-experts to understand applicable laws, identify relevant judicial precedents, and evaluate potential legal outcomes. Furthermore, the financial burden associated with professional legal consultation often discourages individuals from seeking appropriate legal assistance. These

challenges are particularly evident in common disputes involving unpaid wages, residential leases, and consumer transactions.

Recent advances in large language models (LLMs) have created new opportunities to improve the accessibility of legal information. LLM-based systems can interpret natural language queries and generate contextually relevant explanations. However, their direct application to legal information services presents substantial risks, particularly due to hallucinations, inaccurate legal interpretations, and the generation of nonexistent judicial precedents. In legal applications, even a seemingly minor factual or statutory error can lead users to make inappropriate decisions. Therefore, legal AI systems require mechanisms that ensure generated responses are grounded in verifiable legal sources and that communicate the limitations of the retrieved evidence.

Retrieval-Augmented Generation (RAG) has emerged as a promising approach to mitigating these limitations by incorporating externally retrieved documents into the generation process. Nevertheless, conventional RAG architectures may not adequately address the characteristics of Korean legal documents. First, ordinary users frequently describe legal situations using colloquial expressions rather than formal legal terminology, creating a semantic mismatch between user queries and precedent documents. Second, general-purpose sentence embedding models may fail to capture the specialized semantic relationships among legal concepts, statutory provisions, and judicial reasoning. Third, retrieving relevant documents alone does not guarantee the reliability of the generated response. A systematic approach is therefore needed to evaluate the relevance, coverage, and temporal validity of retrieved legal evidence.

To address these challenges, this paper proposes **JuriSim**, a reliability-aware legal simulation framework that integrates domain-specific Ko-SBERT fine-tuning, multi-stage RAG, and quantitative reliability assessment. The system focuses on three major categories of civil disputes: labor, leasing, and consumer protection. Its architecture is designed to transform colloquial user descriptions into legal-domain queries, retrieve relevant precedents and statutory provisions, assess the quality of the retrieved evidence, and generate structured legal analysis based on the retrieved materials.

The first component is a domain-specific sentence embedding model developed through Ko-SBERT fine-tuning using a Hard Positive Triplet Loss strategy. To address the difficulty of selecting informative positive samples, a direct cosine similarity computation method is introduced to identify semantically related chunks within the same precedent. This method avoids the dependency on ChromaDB search-based candidate selection during hard positive mining and addresses the high fallback rate observed in the initial training process. The fine-tuned model is intended to improve semantic matching between colloquial legal queries and formal precedent documents.

The second component is a multi-stage RAG pipeline that combines query normalization, semantic retrieval, legal evidence assessment, and structured response generation. A Gemini-based normalization module first converts colloquial expressions into legal-domain terminology. The normalized query is then used to retrieve relevant precedents and statutory provisions through vector similarity search. Based on the retrieved evidence, a reliability assessment module calculates a weighted score using four indicators: semantic similarity, category matching, evidence coverage, and temporal relevance. The final generation stage produces structured legal analysis using the retrieved documents as contextual evidence.

The third component is a reliability visualization mechanism designed to communicate the quality of the retrieved evidence to users. Unlike approaches that present generated answers without an explicit indication of evidential quality, JuriSim calculates a numerical reliability score on a 0–100 scale. When the score falls below a predefined threshold of 50, the system recommends consultation with a qualified legal professional. This mechanism is intended to help users recognize situations in which the available retrieved evidence may be insufficient to support a reliable legal assessment.

The proposed framework was implemented as a web-based service and deployed on AWS EC2. Its practical performance was examined using a legal precedent dataset containing 11,373 documents, which were segmented into 28,625 vector chunks. Retrieval performance was evaluated across labor, leasing, and consumer dispute categories. In addition, an MD5 hash-based caching mechanism was implemented to reduce redundant processing and improve response efficiency.

The main contributions of this study are summarized as follows:

1. **Domain-specific legal embedding:** A Ko-SBERT fine-tuning approach based on Hard Positive Triplet Loss is proposed to improve semantic retrieval of Korean legal precedents. A direct cosine similarity computation strategy is introduced to address the limitations of search-based hard positive selection.
2. **Colloquial-to-legal query normalization:** A multi-stage query processing mechanism is developed to transform non-expert user expressions into legal-domain terminology, improving the compatibility between user queries and formal legal documents.
3. **Reliability-aware RAG architecture:** A multi-stage RAG framework is designed to integrate precedent retrieval, statutory retrieval, evidence-based reliability assessment, and structured legal response generation.
4. **Quantitative reliability visualization:** A weighted four-indicator scoring model is introduced to communicate the quality of retrieved legal evidence and provide an expert-consultation recommendation when the calculated score is below a predefined threshold.
5. **Practical system implementation and evaluation:** The proposed framework is implemented and deployed as a web-based legal simulation service, and its retrieval performance and response-time efficiency are evaluated across three legal dispute categories.

2. Related Work

A. Natural Language Processing for Legal Applications

Natural Language Processing (NLP) techniques have been increasingly adopted in the legal domain to support document classification, information extraction, legal question answering, and judicial precedent retrieval. Unlike general-domain text, legal documents contain specialized terminology, complex syntactic structures, domain-dependent expressions, and extensive cross-references to statutes and other judicial decisions. These characteristics require language models capable of representing legal concepts and their contextual relationships accurately.

Chalkidis et al. [4] introduced LEGAL-BERT, a family of BERT-based models pretrained or adapted using large collections of English legal documents. Their work demonstrated that domain-specific language modeling can improve performance on legal NLP tasks, including document classification and named entity recognition, compared with general-purpose language models. More recently, LegalBench [5] was introduced as a benchmark for evaluating the legal reasoning capabilities of LLMs across a variety of legal tasks. These studies highlight the importance of domain-specific language representations and systematic evaluation in legal AI applications.

In South Korea, publicly available legal resources, including the National Law Information Open API and AI Hub datasets, provide opportunities for developing Korean legal NLP systems. However, effectively utilizing these resources requires addressing the semantic differences between everyday Korean expressions and formal legal terminology. For example, a user describing a situation as "not receiving a salary" may need to be connected to legal concepts such as unpaid wages, delayed wage

payment, or violations of the Labor Standards Act. General-purpose semantic retrieval models may not consistently capture these domain-specific relationships.

This study addresses this issue by developing a Korean legal-domain sentence embedding model through Ko-SBERT fine-tuning. In addition, a query normalization stage is introduced to bridge the linguistic gap between colloquial user descriptions and formal legal expressions before precedent retrieval.

B. Retrieval-Augmented Generation for Legal Question Answering

Retrieval-Augmented Generation (RAG), introduced by Lewis et al. [3], combines information retrieval with neural text generation by providing retrieved documents as external context to a generative model. This architecture reduces dependence on information stored in model parameters and enables generated responses to be grounded in external knowledge sources. RAG has subsequently been applied to various knowledge-intensive tasks, including question answering, document analysis, and domain-specific information services.

Gao et al. [7] reviewed the evolution of RAG architectures and discussed Advanced RAG and Modular RAG approaches, which incorporate techniques such as query transformation, iterative retrieval, reranking, and modular retrieval pipelines. These developments demonstrate that retrieval quality and query processing are critical factors in the overall performance of RAG-based systems.

In legal applications, the reliability of retrieved evidence is particularly important. A generated response should not only be semantically relevant to a user's question but should also be supported by applicable statutes and judicial precedents. Louis et al. [6] investigated legal case retrieval using the COLIEE dataset and demonstrated the effectiveness of combining traditional lexical retrieval methods, such as BM25, with neural retrieval techniques. Their findings suggest that lexical and semantic retrieval methods can provide complementary benefits when searching legal documents.

Existing RAG approaches, however, do not necessarily provide users with an explicit quantitative assessment of the quality of the retrieved legal evidence. This limitation is important in legal simulation services, where users may interpret generated responses as authoritative legal advice. Accordingly, JuriSim introduces a multi-stage RAG architecture that integrates colloquial query normalization, precedent and statute retrieval, weighted evidence reliability assessment, and structured response generation.

C. Sentence Embeddings and Domain-Specific Fine-Tuning

Sentence embeddings represent textual information as dense numerical vectors, enabling semantic similarity computation and efficient document retrieval. Sentence-BERT (SBERT), proposed by Reimers and Gurevych [9], introduced a siamese and triplet network architecture for generating semantically meaningful sentence representations. Unlike conventional BERT models that require joint inference for sentence-pair comparisons, SBERT enables efficient similarity computation through independently generated sentence embeddings.

For Korean-language applications, KR-SBERT [11] provides sentence-level semantic representations trained using Korean natural language inference and semantic textual similarity datasets, including

KLUE-NLI and KorSTS. Although such models provide a useful foundation for general Korean semantic search, their representations may not sufficiently capture the specialized semantic relationships found in legal documents.

Domain-specific fine-tuning is therefore an important approach for adapting pretrained sentence embedding models to specialized retrieval tasks. Triplet Loss is commonly used for this purpose by optimizing the relative distances among anchor, positive, and negative samples. Given an anchor sentence, a positive sample represents a semantically relevant document, whereas a negative sample represents an irrelevant or less relevant document. The objective is to minimize the distance between the anchor and positive embeddings while maintaining an appropriate margin from negative embeddings.

Hard Positive Mining further improves training by selecting informative positive samples that are semantically related but not trivially similar to the anchor. However, conventional search-based candidate selection can introduce computational overhead and candidate selection failures, particularly when the retrieval index does not contain suitable samples or when the search constraints are restrictive.

To address this issue, this study proposes a direct cosine similarity computation strategy for identifying hard positive samples within the same judicial precedent. By performing similarity calculations directly among chunks belonging to the same case, the proposed method reduces dependence on external vector database search during training data construction. This strategy is incorporated into the Ko-SBERT fine-tuning process to improve the representation of Korean legal-domain semantics.

D. Reliability Assessment and Trustworthiness in Legal AI

The reliability of AI-generated information has become an important research concern, particularly in high-stakes domains such as healthcare, finance, and law. In RAG-based systems, retrieval relevance is an essential factor affecting response quality because the generated answer depends substantially on the accuracy and coverage of the retrieved evidence. However, high semantic similarity alone does not necessarily indicate that a retrieved document is legally applicable or sufficiently supports a generated conclusion.

In the legal domain, evidence quality may depend on several factors, including semantic relevance to the user's query, consistency with the relevant legal category, coverage of the issues raised, and the temporal validity of the applicable legal information. For example, an older precedent may remain relevant to a legal issue, but an outdated statutory provision may no longer be applicable. Therefore, legal evidence assessment should consider multiple dimensions rather than relying exclusively on vector similarity.

JuriSim introduces a weighted reliability assessment model that combines four indicators: semantic similarity (S), category matching (M), evidence coverage (C), and temporal relevance (T). The resulting score is normalized to a 0–100 scale to provide users with an interpretable indication of the quality of the retrieved evidence. A predefined threshold is also used to recommend professional consultation when the score indicates insufficient evidential support.

Unlike approaches that focus exclusively on the relevance of retrieved documents or the quality of generated text, the proposed approach incorporates evidence assessment as an explicit stage within the

RAG pipeline. This design provides an additional mechanism for communicating retrieval limitations and supporting more cautious use of AI-generated legal information.

3. Proposed System

A. Overall System Architecture

JuriSim was designed with a 3-Tier architecture—frontend, backend API server, and AI/ML engine—considering scalability and maintainability (see Fig. 1). When a user selects a category (labor, leasing, or consumer) and inputs a legal situation in natural language, the React-based SPA sends an HTTP POST request to the FastAPI backend, which calls the run_pipeline(query, category) function to execute the multi-stage AI/ML pipeline and returns a structured JSON response containing answer, reliability_score, reference_cases, and reference_laws. The entire service is deployed on AWS EC2 t2.micro and publicly operated at the jurisim.site domain.

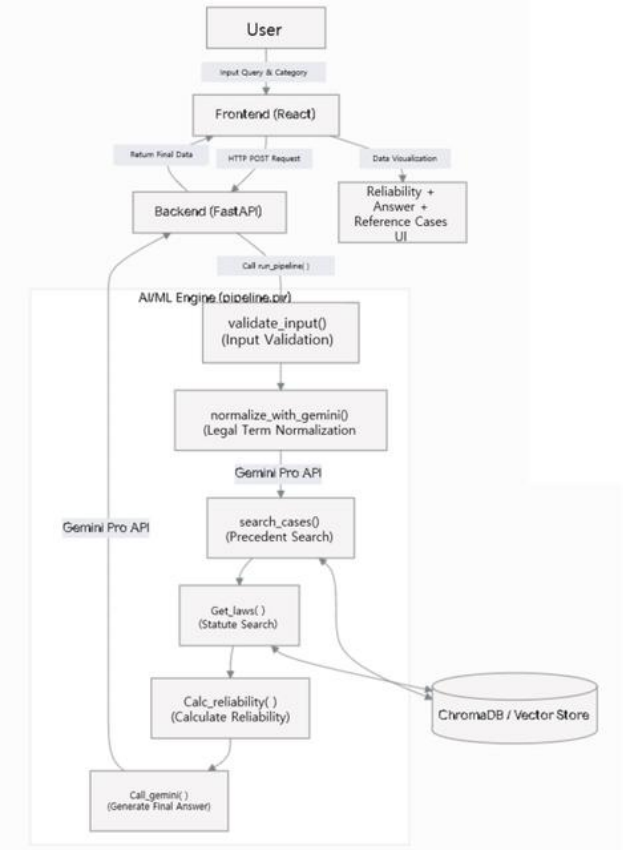


Fig. 1. JuriSim System Architecture Diagram

TABLE I. JuriSim Overall System Configuration

Layer	Component	Technology Stack	Role
Frontend	SPA UI	React 19, Vite 8.0, Axios, styled-components, react-i18next	User interface, reliability gauge visualization, multilingual support
Backend	REST API Server	Python 3.11, FastAPI 0.135, SQLAlchemy 2.0, SQLite, uvicorn	JWT authentication, consultation record persistence,

Layer	Component	Technology Stack	Role
			pipeline orchestration
AI/ML	RAG Pipeline	Ko-SBERT, ChromaDB 1.5.5, Google Gemini 2.5 Flash	Precedent retrieval, statute retrieval, reliability scoring, answer generation
Infrastructure	Cloud Deployment	AWS EC2 t2.micro, nginx 1.28.3, Gabia DNS	Frontend static serving, API proxy, domain connection

B. Data Collection and Preprocessing Pipeline

1) Precedent Data Collection:

Precedent data was collected from four heterogeneous sources, and after deduplication by case number, a total of 11,373 unique precedents were obtained. The category distribution is as follows: labor 41.2%, leasing 29.8%, and consumer 29.0%.

TABLE II. Precedent Data Collection Summary

Data Source	Original Count	Category	Key Characteristics
National Law Information API	1,252 cases	Labor / Leasing / Consumer	Real-time API collection, official verdict full text
HuggingFace (korean_law_open_data_precedents)	85,660 cases	All categories	Cleaned structured precedents, includes summary of judgment
AI Hub No. 115 (Civil & Labor Precedents)	17,079 cases	Labor / Civil	AI Hub public dataset, various court levels
AI Hub No. 03 (Administrative Law Verdicts)	19,489 cases	Administrative / Consumer	Administrative court precedents, consumer-related administrative disputes
Final after deduplication	11,373 cases	3 Categories	Unique precedents by case number

2) Text Preprocessing and Chunking:

The collected precedent texts were organized into the rag_text column using a fallback strategy in the order: 'full verdict → verdict summary → case name'. A sliding overlap of 50 tokens was applied to 512-token chunks, generating a total of 28,625 vector chunks.

TABLE III. Chunk Distribution by Category

Category	Number of Chunks	Proportion	Avg. Chunk Length (Tokens)
Labor	14,723	51.4%	387.2
Leasing	7,120	24.9%	412.6
Consumer	6,782	23.7%	398.1
All categories	28,625	100%	397.8

3) Statute Data Collection:

Through the National Law Information Center Open API, 857 articles from 11 key statutes related to labor, leasing, and consumer affairs were collected. Each statute article was embedded as a single vector without chunking and stored in ChromaDB.

C. Legal Domain-Specific Ko-SBERT Fine-Tuning

1) Analysis of Fine-Tuning Necessity:

A baseline experiment applying the pre-trained KR-SBERT directly to legal precedents recorded an accuracy_cosine of 4.8%. This indicates that general-domain embeddings fail to accurately capture the semantic similarity of legal synonyms such as 'unpaid wages' vs. 'part-time wage theft', or 'refusal to return lease deposit' vs. 'landlord won\'t give back my deposit'.

2) Triplet Dataset Construction and Hard Positive Strategy:

Fine-tuning uses Triplet Loss as the objective function. The Anchor is a random precedent chunk, the Positive is a semantically similar chunk within the same case, and the Negative is a chunk from a different category. The ChromaDB search-based approach resulted in a fallback rate of 81%, where an average of 2.5 chunks per case were not found within the top 300 results. To address this, the method was switched to computing direct cosine similarity only within the chunk set of the same case, achieving a fallback rate of 0% and accuracy_cosine of 76.2%.

TABLE IV. Performance Comparison by Hard Positive Selection Strategy

Strategy	Hard Positive Selection Method	Fallback Rate	Number of Triplets	Accuracy_cosine
Easy Positive	Random chunk selection within same case	0%	7,189	30.7%
ChromaDB Search-Based	ChromaDB top-300 search within full pool	81.0%	7,323	48.2%
Direct Similarity Computation (Adopted)	Direct cosine similarity between chunks within same case	0%	7,351	76.2%

3) Fine-Tuning Configuration:

TABLE V. Fine-Tuning Hyperparameter Settings

Hyperparameter	Value	Rationale
Base Model	snunlp/KR-SBERT-V40K-klueNLI-augSTS	Trained on Korean NLI and STS; excellent sentence similarity performance
Loss Function	TripletLoss (Cosine Distance-based)	Better suited for learning relative order than absolute similarity
Batch Size / Epochs	16 / 3	Maximum GPU memory utilization, prevents overfitting
Learning Rate	2e-5 (AdamW)	SBERT fine-tuning recommended default
Margin	0.5	Sets minimum distance between Positive and Negative
Training Environment	NVIDIA RTX 4070 Super, PyTorch 2.1.0+cu121	GPU acceleration utilized

D. Multi-Stage RAG Pipeline

The RAG pipeline is designed as a sequential nine-stage processing structure, with each stage handling exceptions independently to ensure service continuity even in the event of partial failures (see Fig. 2).

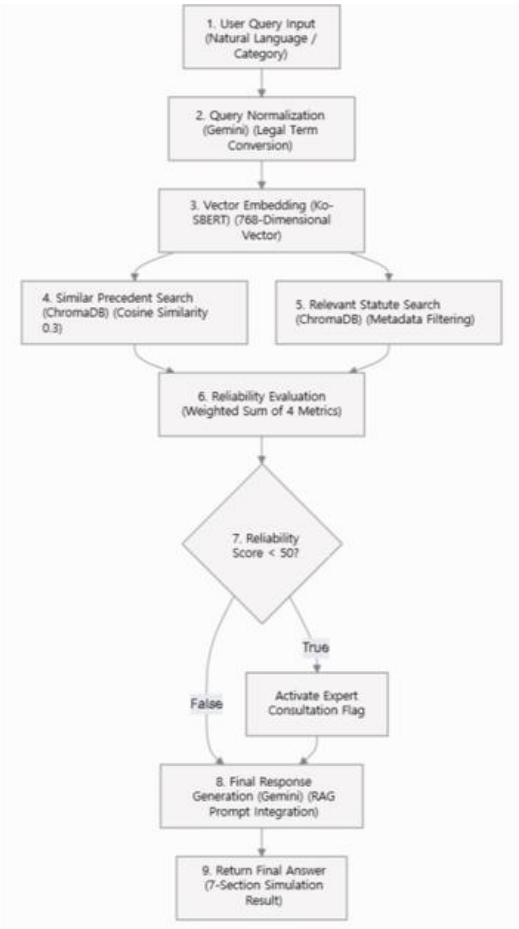


Fig. 2. Multi-Stage RAG Pipeline

1) Input Validation and Cache Lookup:

Input is validated through length constraints (10–1,000 characters), HTML tag detection (XSS defense), and prompt injection pattern detection (9 keywords). Subsequently, an MD5 hash-based in-memory cache is queried to block repeated API calls for the same query-category combination. The cache manages up to 500 entries using an LRU policy, reducing average response time by approximately 62%.

2) Colloquial-to-Legal Terminology Normalization (Gemini 1st Call):

Category-specific legal terminology context is provided to Gemini along with few-shot examples to convert colloquial input into legal terminology. A fallback mechanism that retains the original text is applied when the conversion result is shorter than 25 characters or ends with a grammatical particle.

TABLE VI. Colloquial-to-Legal Terminology Normalization Examples

Colloquial Input	Normalized Output	Processing Category
I'm not getting paid for my part-time job	A case of not receiving wage payments	Labor
My landlord won't return my deposit	A case where the lessor refuses to return the lease deposit	Leasing
Being evicted for overdue rent	A case demanding eviction due to rent arrears	Leasing
I bought it and it's defective, I want a refund	A case requesting a refund due to a defect in the purchased goods	Consumer

3) Precedent and Statute Retrieval:

The normalized composite query is embedded using the fine-tuned Ko-SBERT, followed by cosine similarity-based retrieval in ChromaDB. Precedent retrieval removes results below the similarity threshold (0.3) and returns up to 3 cases, while statute retrieval returns up to 3 articles after filtering with the condition where={'category': category}.

4) Reliability Scoring (calc_reliability):

$$\textbf{Reliability} = (S \times 0.50 + M \times 0.25 + C \times 0.15 + T \times 0.10) \times 100$$

TABLE VII. Reliability Scoring Indicators

Indicator	Formula	Weight	Description
S (Similarity)	$\Sigma similarity_i / n$	0.50	Mean cosine similarity of retrieved precedents. Direct measure of retrieval quality.
M (Category Match Rate)	$\Sigma 1[cat_i = query_cat] / n$	0.25	Proportion of precedents matching the query category. Measures category relevance.
C (Coverage)	$\Sigma 1[sim_i \geq 0.5] / 3$	0.15	Proportion of high-quality precedents with similarity ≥ 0.5.
T (Recency)	$\Sigma \max(0, 1 - years/30) / n$	0.10	Weight linearly decreasing over 30 years from the ruling date.

When reliability falls below 50, recommend_expert=True is returned to advise expert consultation. The initial design included a statute similarity indicator (L), but it was removed due to distortion of reliability scores when statute DB retrieval failed.

5) Answer Generation (Gemini 2nd Call):

Three precedents (up to 600 characters per chunk) and three statutes (up to 200 characters per article) are assembled into a structured prompt template for the second Gemini call. The settings are temperature=0.2 and max_output_tokens=4096, with a system prompt enforcing a seven-section response: situation summary, precedent analysis, related statutes, expected outcomes, action guidelines, comprehensive advice, and disclaimer.

E. Backend and Database Design

The backend is built on a FastAPI asynchronous server with responsibilities separated by domain-specific routers (auth, users, legal, history, chat). A dual JWT token scheme (Access: 30 minutes / Refresh: 7 days) is implemented, with Refresh Tokens stored in the database to prevent token reuse attacks. The database consists of six relational tables (see Fig. 3), and the precedents and statutes underlying consultation results are preserved as snapshots along with similarity scores and rankings, ensuring traceability even after the original data changes. During the storage process, db.flush() pre-acquires the log_id before inserting into child tables, and all operations are wrapped in a single commit to guarantee transaction atomicity.

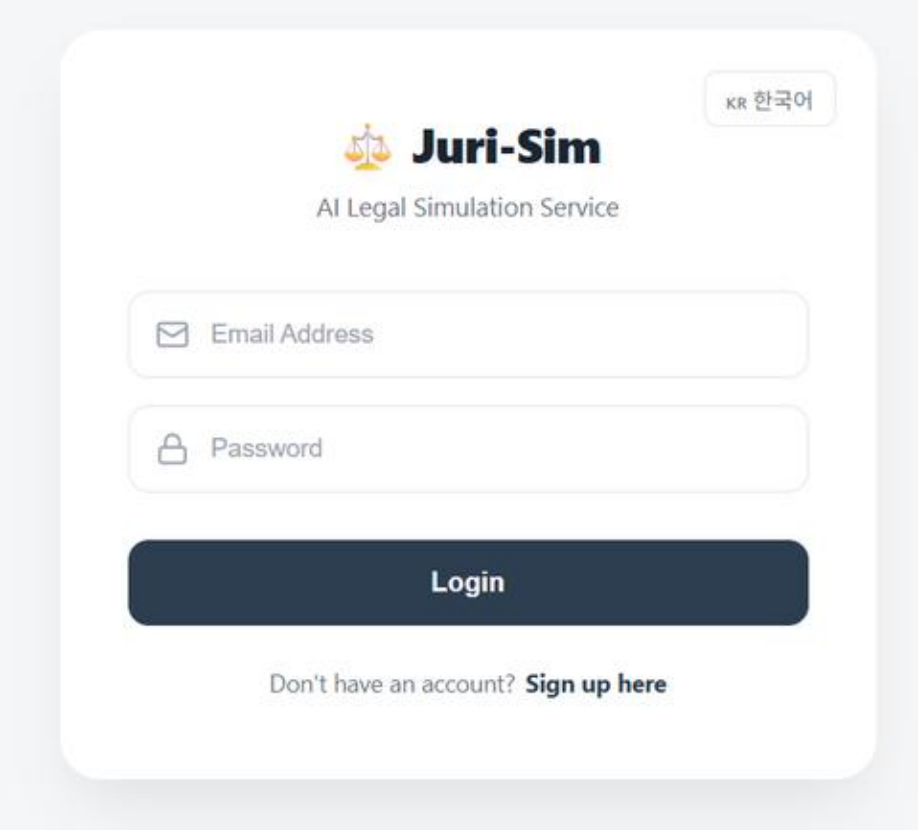


Fig. 3. Entity-Relationship Model (ERD)

F. Frontend and User Interface

The client application is implemented as a React 19 and Vite-based SPA, designed around three key views to satisfy both user accessibility and system stability.

1) Hybrid UX-Based Main Dashboard:

A hybrid UX combining a chatbot-based conversational interface with a dashboard that maintains analytical professionalism was adopted (see Fig. 4). To avoid the React 19 dependency conflict with the react-gauge-chart library, a custom responsive gauge chart was implemented directly using pure SVG and CSS animations. Additionally, a defensive UI mechanism was implemented that immediately triggers a warning modal when the reliability score falls below 50, recommending expert consultation (see Fig. 5).

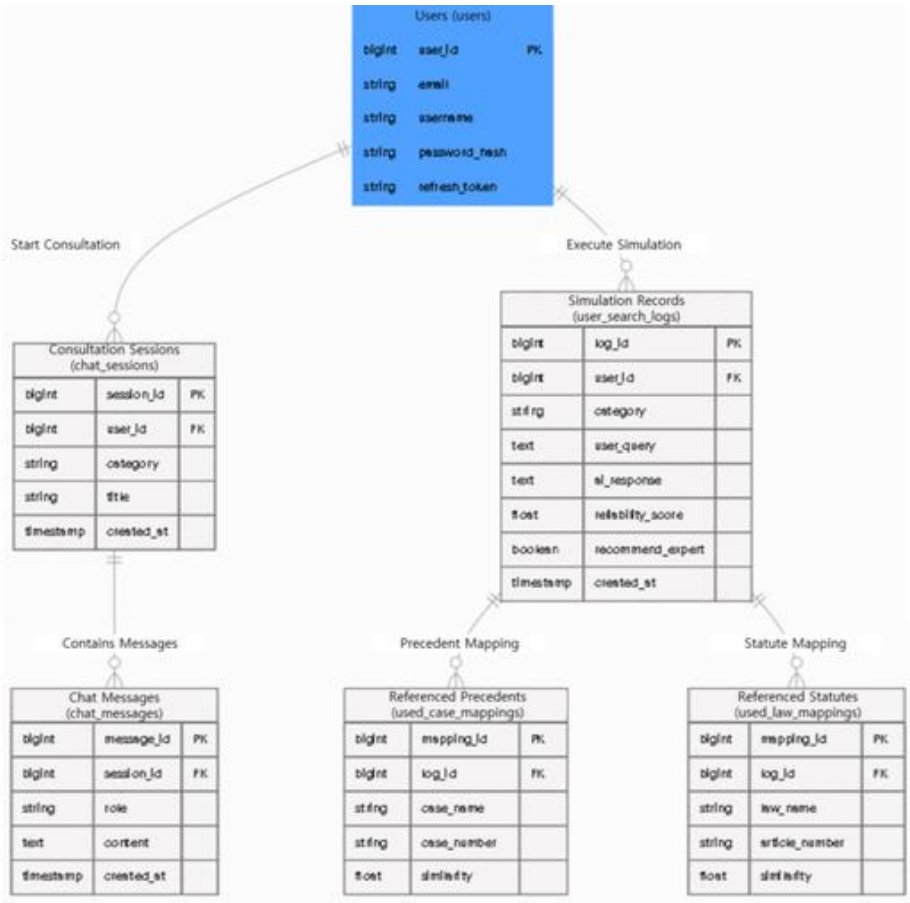


Fig. 4. Main Dashboard (Hybrid UX — Chatbot Interface + Reliability Gauge)

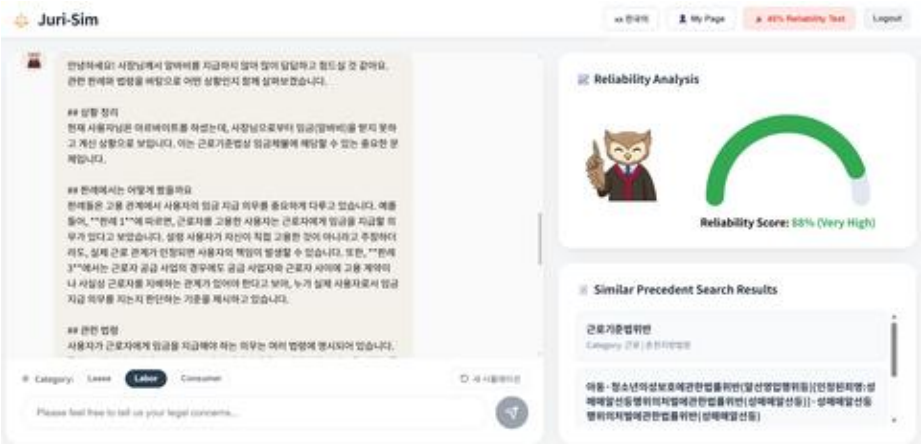


Fig. 5. Warning Modal (Reliability < 50: Expert Consultation Recommended)

2) State-Based User Authentication Module:

Login and registration modes were designed to switch state-based within a single component (see Fig. 6). A defensive form was built that conditionally displays the backend password complexity requirements (at least 8 characters including letters, numbers, and special characters) on the frontend, minimizing the occurrence of 422 Unprocessable Entity errors.

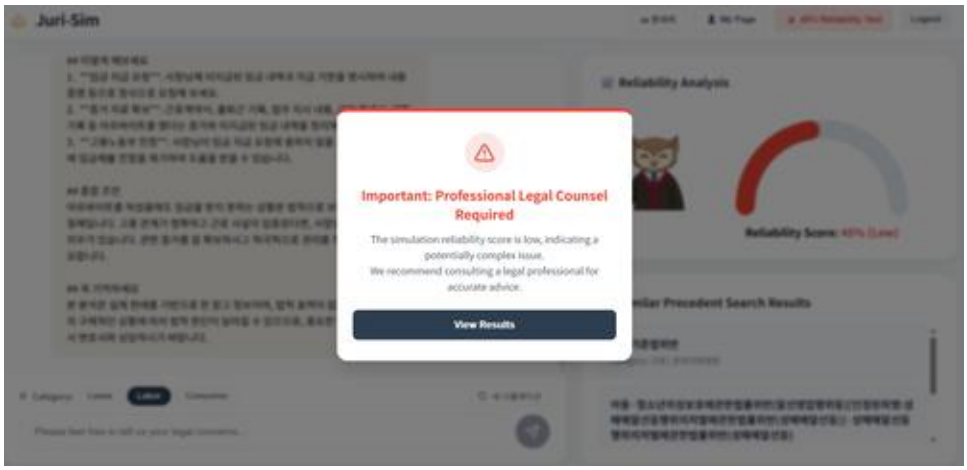


Fig. 6. Login/Registration (State-Based Authentication Module)

3) Personalized Session and History Management:

The My Page follows a tab-based information architecture with four tabs: history, scrapbook, statistics, and settings (see Fig. 7). The multilingual (i18n) system separates category identifiers (fixed in Korean) from UI display labels (multilingual), maintaining ChromaDB search integrity when switching languages.

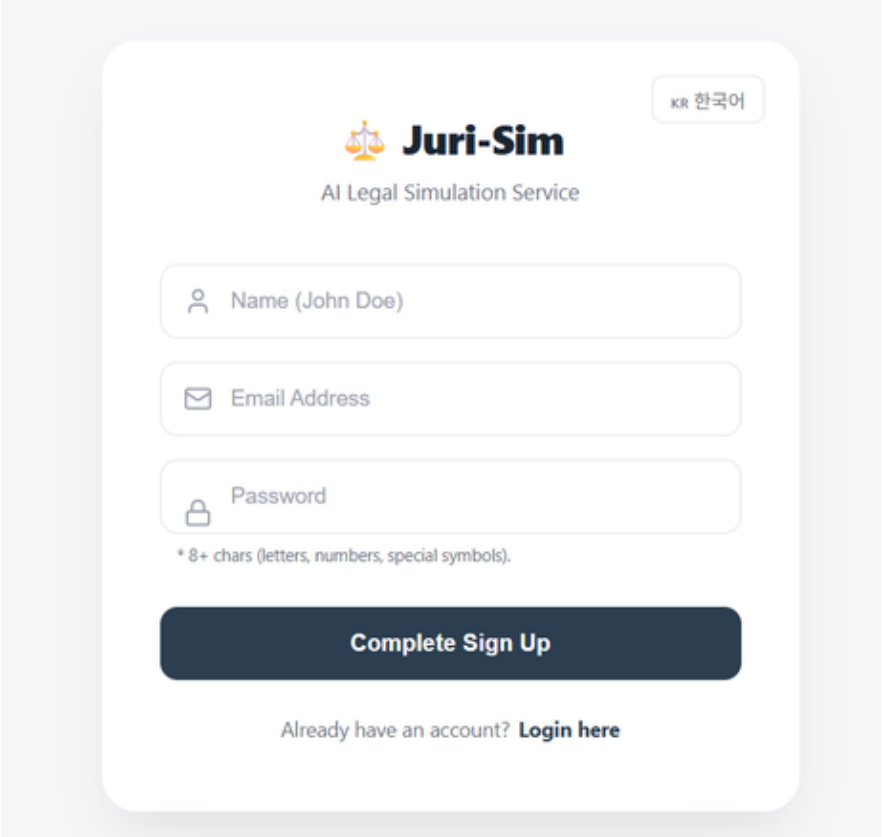


Fig. 7. My Page (Tab-Based Information Architecture and History Management)

IV. Experiments and Evaluation

A. Ko-SBERT Fine-Tuning Performance Evaluation

The retrieval similarity of the baseline model (KR-SBERT, no fine-tuning) and the fine-tuned model (jurisim-sbert-v4) were compared. The average improvement was 0.468, clearly demonstrating the effectiveness of fine-

tuning, with the leasing and consumer categories achieving high similarity scores above 0.93.

TABLE VIII. Retrieval Similarity Comparison: Baseline vs. Fine-Tuned

Query	Category	Baseline	After Fine-Tuning	Improvement
I want to evict a tenant who does not pay rent	Leasing	0.442	0.9541	+0.512
I was wrongfully dismissed	Labor	0.381	0.7783	+0.397
I suffered damages from a refund refusal as a consumer	Consumer	0.419	0.9367	+0.518
I did not receive my severance pay	Labor	0.367	0.8124	+0.445
I have not been able to get my jeonse deposit back	Leasing	0.455	0.9212	+0.466

B. Reliability Scoring Result Analysis

TABLE IX. Reliability Scoring Results by Scenario

Scenario	S	M	C	T	Reliability	Expert Recommendation
Leasing Dispute (Jeonse Deposit Return)	0.921	1.0	1.0	0.74	92.5	Not Recommended
Labor Dispute (Wage Theft)	0.778	1.0	1.0	0.68	82.7	Not Recommended
Consumer Dispute (Defect Refund)	0.937	1.0	1.0	0.71	94.2	Not Recommended
Category Boundary Case (Complex Dispute)	0.543	0.67	0.67	0.58	58.3	Not Recommended
Insufficient Precedents (Special Case)	0.318	0.33	0.0	0.52	22.5	Recommended

C. Cache Performance and Deployment Environment Verification

TABLE X. Cache Performance Evaluation Results

Metric	Cache Miss (First Request)	Cache Hit (Repeated Request)	Reduction Rate
Average Response Time	~8.3 seconds (including Gemini API)	~3.2 seconds	~61.4%
Gemini API Call Count	2 calls (normalization + answer generation)	0 calls	100% reduction
ChromaDB Search Count	2 searches (precedent + statute)	0 calls	100% reduction

TABLE XI. Key Troubleshooting History

Problem	Cause	Solution	Result
ChromaDB Index Corruption	OneDrive sync interference	Moved project outside OneDrive and re-ran rebuild_db.py	28,625 chunks successfully

Problem	Cause	Solution	Result
			restored
Server Process Force Termination	Ko-SBERT model load exceeded 1GB RAM on t2.micro	Added 2GB Linux swap memory	Model loading and server operation normalized
bcrypt 500 Error	API change in latest passlib version	pip install bcrypt==4.0.1 --force-reinstall	Registration and login normalized
nginx 403 Forbidden	Insufficient permissions on /home/ubuntu/dist	sudo chmod 755 /home/ubuntu	Frontend serving normalized

4. Conclusion

This paper presents the design, implementation, and deployment of JuriSim, a Retrieval-Augmented Generation (RAG)-based legal simulation web service designed to bridge the accessibility gap in legal services. The core technical contributions of this study are as follows.

First, in the selection of Hard Positive samples, the 81% fallback problem inherent in the conventional ChromaDB search-based approach was resolved by introducing a direct cosine similarity calculation method between chunks within the same case. Consequently, the accuracy_cosine of Ko-SBERT fine-tuning was substantially improved from 4.8% to 76.2%.

Second, a first-stage normalization mapping colloquial language to legal terminology using the Gemini LLM, along with a category context injection strategy, enhanced the search quality for non-expert users. A flexible fallback mechanism for incomplete conversions was established to minimize the adverse effects of normalization failures.

Third, a four-indicator weighted reliability formula was proposed: Reliability = (S × 0.50 + M × 0.25 + C × 0.15 + T × 0.10) × 100. This quantifies the trustworthiness of AI responses on a 100-point scale and automatically recommends expert consultation when the score drops below 50, effectively controlling the hallucination risk inherent in generative AI systems.

Fourth, by introducing an MD5 hash-based LRU cache structure, response times for repetitive queries were reduced by approximately 61.4%. Defense mechanisms against XSS and prompt injection attacks were also integrated to secure the robustness and safety of the RAG pipeline.

In empirical evaluations following production deployment, JuriSim achieved a search similarity of 0.77 or higher across all major categories (labor: 0.7783, leasing: 0.9541, consumer: 0.9367). The server crash issue in the resource-constrained AWS EC2 t2.micro environment was resolved via Linux swap memory optimization, demonstrating the feasibility of practical service deployment in low-cost infrastructure.

Future research will focus on introducing multi-turn conversation using Gemini's start_chat(), automated query category classification, hyperlinks to original statutory texts, migration to the google-genai library, and transitioning the database to PostgreSQL for enhanced concurrent scalability.

Acknowledgments

This research was supported by the MSIT(Ministry of Science ICT), Korea, under the National Program for Excellence in SW, supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation) in 2026 (No. 2024-0-00023).

References

[1] Ministry of Justice, "Survey on the Accessibility of Legal Services," Ministry of Justice Policy Research, 2022.

[2] J. Ji et al., "Survey of hallucination in natural language generation," ACM Computing Surveys, vol. 55, no. 12, pp. 1–38, Mar. 2023.

[3] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020.

[4] I. Chalkidis et al., "LEGAL-BERT: The muppets straight out of law school," in Findings of EMNLP 2020, pp. 2898–2904, 2020.

[5] N. Guha et al., "LegalBench: A collaboratively built benchmark for measuring legal reasoning in LLMs," in Proc. NeurIPS, 2023.

[6] A. Louis, G. van Dijck, and G. Spanakis, "Statutory reasoning assessment at scale," in Proc. EACL, pp. 1288–1310, 2023.

[7] Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," arXiv:2312.10997, 2024.

[8] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," IEEE Trans. Big Data, vol. 7, no. 3, pp. 535–547, 2021.

[9] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in Proc. EMNLP, pp. 3982–3992, 2019.

[10] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in Proc. CVPR, pp. 815–823, 2015.

[11] snunlp, "KR-SBERT," Hugging Face, 2022. [Online]. Available: https://huggingface.co/snunlp/KR-SBERT-V40K-klueNLI-augSTS

[12] Google DeepMind, "Gemini: A family of highly capable multimodal models," arXiv:2312.11805, 2023.

[13] National Law Information Center, "Statute OpenAPI Service," 2024. [Online]. Available: https://www.law.go.kr/LSW/openApi.do

[14] ChromaDB, "Chroma: The open-source embedding database," 2024. [Online]. Available: https://docs.trychroma.com

[15] Y. Gao et al., "Precise zero-shot dense retrieval without relevance labels," in Proc. ACL, pp. 1762–1777, 2023.