SBV-LawGraph: A Hybrid RAG Approach Integrating Knowledge Graph for the State Bank of Vietnam Legal Documents

Khoa Phan[1,2], Xuan-Bach Le[1,2*], and Tho Quan[1,2]

[1] Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Vietnam
[2] Vietnam National University Ho Chi Minh City, Ho Chi Minh City, Vietnam
{pqkhoa.sdh231, lexuanbach, qttho}@hcmut.edu.vn

Abstract. While Retrieval-Augmented Generation (RAG) pipelines often demonstrate strong performance in general settings, they often struggle with legal texts, where interpreting the structure and relationships between laws is crucial. To address this, we introduce SBV-LawGraph – a dual-retrieval framework designed specifically for Vietnamese legal documents. It combines semantic retrieval with graph-based reasoning by integrating two modules: a Legal Retrieval module that uses sparse–dense reranking for textual accuracy, and a Relationship Retrieval module that traverses a curated Legal Knowledge Graph to capture links like amendments, citations, and definitions. This design enables SBV-LawGraph to generate responses that are not only relevant but also structurally grounded, addressing the limitations of standard RAG systems. Evaluations on the ALQAC2025 and SBV Legal Questions datasets show it consistently outperforms strong baselines, highlighting its effectiveness for precise and explainable legal QA.

Keywords: Retrieval-Augmented Generation, Knowledge Graph, Natural Language Processing, Question-Answering Systems.

1 Introduction

Large Language Models (LLMs) have powerful generative capabilities but often make factual errors in specialized domains due to limited external knowledge access [21,18]. Retrieval-Augmented Generation (RAG) addresses this by using vector retrieval to retrieve relevant information before answering [25,14]. However, traditional RAG methods, with basic chunking and retrieval, often lack precision and miss document relationships [23,30]. Newer methods such as AdvancedRAG [26,23,16] and GraphRAG [44,40,14] were proposed to improve performance – AdvancedRAG uses semantic chunking, hybrid search and reranking, while GraphRAG builds knowledge graphs to capture deeper context and connections in the documents.

* Corresponding author.

However, neither AdvancedRAG nor GraphRAG alone is sufficiently robust to handle Vietnamese legal documents, which are known to be inherently structured and densely interlinked [36,35]. While AdvancedRAG excels at retrieval, it lacks reasoning capability over complex legal references. Conversely, GraphRAG captures relational structures, but often fails to account for critical legal semantics. To overcome these challenges, we propose **SBV-LawGraph**, a RAG-based framework that unifies the strengths of both AdvancedRAG and GraphRAG, to handle Vietnamese legal documents. Our framework employs a dual-retrieval mechanism composed of two concurrent components:

1. **Legal Retrieval Mechanism (SBV-LR).** This module uses techniques such as semantic chunking, hybrid search, and re-ranking to improve semantic clarity. Semantic chunking segments legal texts by meaning, hybrid search integrates dense and sparse retrieval (e.g., BM25 [48]) to balance recall and precision, and re-ranking refines top results using contextual relevance. These steps help to ensure that the information found is both relevant to the topic and appropriate from a legal point of view.
2. **Relationship Retrieval Mechanism (SBV-RR).** This module models the dependencies among legal articles using a purpose-built *Legal Knowledge Graph (LKG)*. The LKG was constructed using zero-shot and few-shot prompting with reasoning LLMs. This approach helps to uncover key relationships between legal provisions, such as how certain articles amend current law, introduce new regulations, repeal outdated statutes, or refer to legal interpretation. By traversing this graph, SBV-RR can retrieve related articles that provide a deeper and more meaningful legal context.

Finally, our framework combines the retrieved semantic and structural information to create a comprehensive and context-aware prompt for the LLMs.

Our main contributions are as follows.

1. We introduce SBV-LawGraph, a unified dual-retrieval RAG framework that integrates semantic and relational reasoning to enhance legal question answering in the Vietnamese domain.
2. We build an LKG for the State Bank of Vietnam's legal texts using zero- and few-shot prompting with LLMs. The LKG captures the structure and links between provisions, showing how they amend, repeal, or clarify laws.
3. We evaluated SBV-LawGraph on two Vietnamese legal benchmarks: AI-LegalQA 2025 and an internal SBV dataset. The results demonstrate that SBV-LawGraph outperforms existing RAG-based and state-of-the-art LLMs baselines in both retrieval precision and contextual accuracy.

The remainder of this paper is structured as follows. Section 2 reviews related work on RAG pipelines and legal LLMs. Section 3 formulates the problem while Section 4 details the proposed SBV-LawGraph framework. Section 5 presents the experimental setup and Section 6 reports and discusses the results. Finally, Section 7 concludes the paper and outlines future research directions.

2 Related Work

This section reviews two key research directions that lay the foundation for our framework: RAG pipelines – from basic to advanced, and LLM-based legal AI.

2.1 Retrieval-Augmented Generation Pipelines

RAG combines retrieval and generation to ground responses in evidence [25], improving accuracy in QA, dialogue, and fact-checking [20]. But early RAG models use static retrievers and basic chunking, leading to incomplete or redundant results [56]. These shortcomings highlight the need for more adaptive and semantically aware retrieval–generation integration. Recent advances such as agentic and multimodal RAG architectures address this gap by enabling more dynamic, context-aware reasoning [49,17,19].

To address these limitations, recent work has split into two paths: AdvancedRAG for better semantic matching, and GraphRAG for structured, graph-based reasoning. AdvancedRAG boosts semantic accuracy with co-training, adaptive retrievers, and re-ranking [22,28,10]. GraphRAG builds document graphs to model entities and links, enabling structured reasoning [6,15,37]. While AdvancedRAG improves lexical and contextual matching, GraphRAG excels in modeling hierarchical and referential dependencies. These findings suggest that the combination of semantic and structural reasoning is essential for complex domains such as legal text.

2.2 LLM-based Legal AI Systems

Recent legal AI focuses on adapting LLMs through domain-specific pretraining [11,32] and task fine-tuning [7]. Methods range from LegalBERT-style encoders [8] to instruction-tuned LLMs [53,39], targeting tasks like legal QA, classification, and case prediction [2,57].

Dense retrieval uses neural embeddings to match queries and documents by semantics [23,12]. It has been effective for case law and statute retrieval [4,13], but struggles with structured legal reasoning, such as linking provisions or tracking amendments [29,27,41].

Domain-adapted models like PhoBERT [32,36] and benchmarks such as the Zalo AI Challenge [55] and ALQAC [3] have advanced Vietnamese legal NLP. Yet, text-only approaches still face challenges with word segmentation, diacritics, and regional variation [46], which degrade embedding quality. SBV-LawGraph bridges this gap by combining a legal knowledge graph with a semantic retriever tailored to Vietnamese law, enabling stronger evidence aggregation. To the best of our knowledge, it is the first to integrate linguistic adaptation and graph-based retrieval for statutory reasoning.

3 Framework Formulation for Legal QA

Our approach frames the Legal QA task as a sequence of two interdependent sub-tasks: *retrieval* and *generation* [5,50,24,58]. Let $\mathbb{T}$ denote the space of all textual representations, e.g. user queries and legal documents expressed in natural

language form. Given a natural language query $q \in \mathbb{T}$, a legal corpus $\mathcal{C} \subset \mathbb{T}$, and an LKG $G_{\text{LKG}} = (\mathcal{V}, \mathcal{E})$—where $\mathcal{V}$ represents legal entities (e.g., laws, articles, or clauses) and $\mathcal{E}$ captures their interconnections (such as amendments, repeals, or interpretive links)–the system begins by identifying relevant documents.

First, a hybrid sparse–dense retrieval function $\mathcal{R}_D$ retrieves a ranked list of the top-k candidate documents with their corresponding relevance scores:

$$\hat{D}_q = \mathcal{R}_D(q, \mathcal{C}) = \{(d_1, s_1), \ldots, (d_k, s_k)\}.$$

Next, this set is filtered using a relevance threshold τ to form the final document context D_q. This logic ensures that only documents with a sufficiently high relevance score ($s_i \geq \tau$) are retained for the generation step:

$$D_q = \{d_i \mid (d_i, s_i) \in \hat{D}_q \wedge s_i \geq \tau\}.$$

Once this high-relevance document set D_q is established, the system's subsequent actions are conditional. If no relevant documents are found ($D_q = \emptyset$), the system bypasses context enrichment and generation, returning a predefined fallback response a_{fallback}. Otherwise (if $D_q \neq \emptyset$), the system proceeds to enrich the context by extracting a task-specific subgraph:

$$G_q = \mathcal{R}_{\text{LKG}}(D_q, G_{\text{LKG}}),$$

where $\mathcal{R}_{\text{LKG}}$ identifies entities and relationships in G_{LKG} that are semantically or legally linked to the retrieved documents in D_q.

Finally, the generation module synthesizes the query, textual evidence, and relational context. The complete answer generation a_q is formally defined as:

$$a_q = \begin{cases} a_{\text{fallback}} & \text{if } D_q = \emptyset \\ \mathcal{G}_f(q, D_q, G_q) & \text{if } D_q \neq \emptyset \end{cases}$$

where $\mathcal{G}_f$ is a generation function that combines both retrieved content and graph-based legal context to ensure the answer is accurate, well-supported, and sensitive to legal nuance.

4 The SBV-LawGraph Framework

We present *SBV-LawGraph*, a two-stage system that integrates semantic retrieval with legal knowledge graph reasoning to generate verifiable answers. This section covers the system architecture (Section 4.1), data indexing (Section 4.2), and the retrieval-to-generation pipeline (Section 4.3).

4.1 System Overview

The SBV-LawGraph framework operates on a dual-retrieval architecture – textual contents and graph-based relationships – as illustrated in Figure 1.

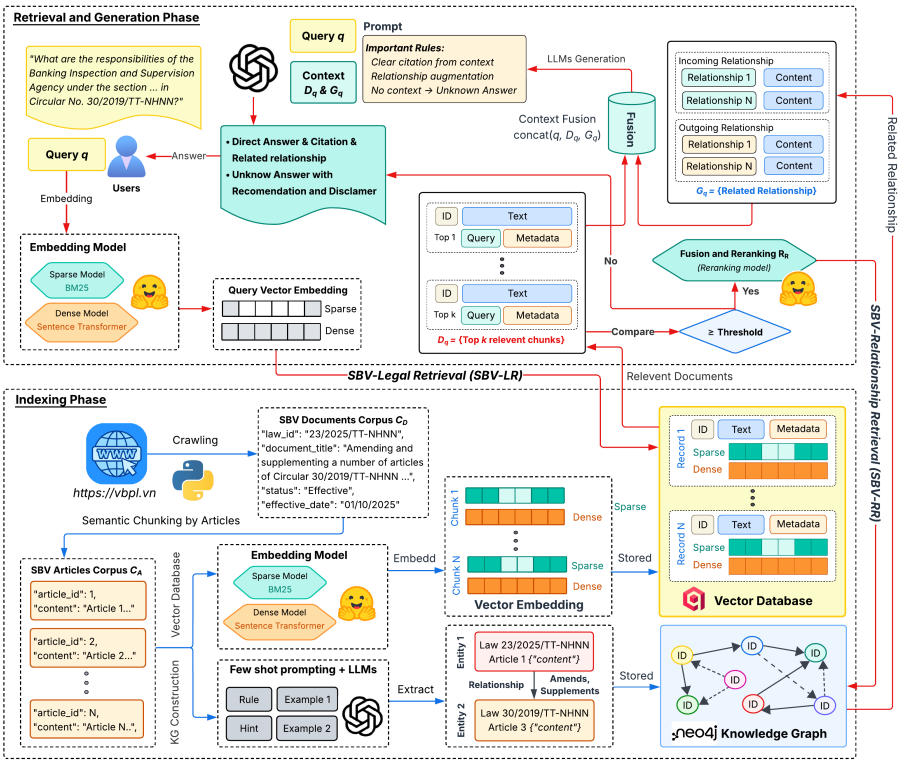


Fig. 1: **SBV-LawGraph's two phases**: (i) offline indexing builds an LKG from legal texts, and (ii) online retrieval–generation combines vector and graph-based evidence to generate citation-backed answers.

Indexing Phase. The procedure starts with offline indexing, which involves crawling, cleaning, and dividing the legal corpus into individual article units. We utilize two parallel pipelines to prepare the data:

1. **Vector Database:** Dense and sparse embeddings are generated for each segment, capturing both lexical and semantic features for retrieval.
2. **Legal Knowledge Graph (LKG):** Relationship extraction using LLMs identifies legal entities and their connections to build a knowledge graph.

Retrieval Phase. When a query is issued, a dual-retrieval mechanism collects complementary evidence:

1. **SBV-Legal Retrieval (SBV-LR):** Textual content is retrieved through a hybrid sparse–dense search, refined through reranking.
2. **SBV-Relationship Retrieval (SBV-RR):** The LKG is queried to surface related legal entities and their immediate relationships.

Retrieved text and graph-based context are combined into a unified prompt to produce responses that are based on credible sources.

Algorithm 1 SBV-LawGraph: Offline Indexing

Input: SBV raw corpus $\mathcal{D}_{\text{raw}}$
Output: Vector DB $\mathcal{V}$, Legal Knowledge Graph G_{LKG}

$\mathcal{D} \leftarrow \textsc{CrawlAndClean}(\mathcal{D}_{\text{raw}})$
$\mathcal{A} \leftarrow \textsc{SegmentByArticle}(\mathcal{D})$ ▷ Rule-based keyword splitting
for all $a \in \mathcal{A}$ **do**
    $v_{\text{sparse}}(a) \leftarrow \text{BM25}(a)$
    $v_{\text{dense}}(a) \leftarrow \textsc{Embed}(a;$ `paraphrase-vietnamese-law`$)$
    $\textsc{UpsertToVectorDB}(\mathcal{V}, a, v_{\text{sparse}}(a), v_{\text{dense}}(a))$
for all $d \in \mathcal{D}$ **do**
    $E(d), R(d) \leftarrow \text{LLM}\textsc{ExtractEntitiesRels}(d;$ `gpt-oss-120b`$)$
    for all $e \in E(d)$ **do**
        $\textsc{AddNode}(G_{\text{LKG}}, \textsc{MakeNodeID}(e))$
    for all $r = (e_i \xrightarrow{\rho} e_j) \in R(d)$ **do**
        if $\rho \in \{\textsc{Amend}, \textsc{Repeal}, \textsc{Replace}, \textsc{Guide}\}$ **then**
            $\textsc{AddEdge}(G_{\text{LKG}}, e_i, \rho, e_j)$
return $(\mathcal{V}, G_{\text{LKG}})$

4.2 Indexing Phase

Offline indexing turns raw legal texts into a structured, multi-modal knowledge base (Algorithm 1). Two core components are constructed in this step: a vector-based retriever and an LKG. We collect legal documents from the official SBV repository and segment them into article-level units using a rule-based keyword strategy. From there, the data are processed along two parallel paths:

1. **Vector Embedding Generation.** Each text chunk is processed through a hybrid embedding model that generates both sparse and dense vectors. Sparse representations are generated with the BM25 algorithm [48], while dense embeddings are obtained from the `paraphrase-vietnamese-law` Sentence Transformer [42], fine-tuned from `paraphrase-multilingual-mpnet-base-v2` [47] on Vietnamese legal QA datasets (ViLQA, ALQAC2024). The model maps text into a 768-dimensional space to represent its lexical and semantic features. These embeddings are stored in Qdrant vector database [45].
2. **Knowledge Graph Construction.** We use the `gpt-oss-120b` model [38] with few-shot prompting to extract key legal entities and their relationships. Our framework utilizes four common relationship types that are prevalent in legal documents: "Amend, Supplement", "Repeal", "Replace", and "Guidance, Regulation". The extracted entities become nodes, and their links form edges in the LKG. The graph structure is stored in Neo4j, a native graph database management system [52].

The two components – vector-based retrieval and structured legal graph – form the SBV-Legal Knowledge Base for context-aware legal question answering.

Algorithm 2 SBV-LawGraph: Online Retrieval and Generation

Input: Query q, Vector DB $\mathcal{V}$, LKG G_{LKG}, thresholds $(k,\ \tau)$
Output: Answer a_q

SBV-LR (Hybrid Retrieval + Reranking)

$u_{\text{dense}} \leftarrow \textsc{Embed}(q;\ \texttt{paraphrase-vietnamese-law})$
$\mathcal{C}_{\text{sparse}} \leftarrow \textsc{SearchBM25}(\mathcal{V}, q)$
$\mathcal{C}_{\text{dense}} \leftarrow \textsc{SearchDense}(\mathcal{V}, u_{\text{dense}})$
$\hat{\mathcal{C}} \leftarrow \textsc{Union}(\mathcal{C}_{\text{sparse}}, \mathcal{C}_{\text{dense}})$
$\hat{\mathcal{C}} \leftarrow \text{RRF}(\hat{\mathcal{C}})$
$\hat{\mathcal{C}} \leftarrow \textsc{ReRank}(\hat{\mathcal{C}};\ \texttt{ViRanker},\ \texttt{bge-reranker-v2-m3})$
$D_q \leftarrow \{d \in \hat{\mathcal{C}} \text{ top-}k \ \wedge\ \textsc{Score}(d) \geq \tau\}$
if $D_q = \emptyset$ **then**
    return `Unknown Answer` ▷ Controlled fallback

SBV-RR (Graph Expansion, 1-Hop)

$\mathcal{E}_q \leftarrow \textsc{DetectEntities}(q)$ ▷ Lightweight NER over query
$\mathcal{E}_D \leftarrow \textsc{MapDocsToEntities}(D_q)$
$\mathcal{S} \leftarrow \mathcal{E}_q \cup \mathcal{E}_D$ ▷ Anchor set
$G_q \leftarrow \emptyset$
for all $e \in \mathcal{S}$ **do**
    $N_{\text{in}}(e) \leftarrow \{x \mid (x \xrightarrow{\rho} e) \in G_{\text{LKG}},\ \rho \in \{\textsc{Amend}, \textsc{Repeal}, \textsc{Replace}, \textsc{Guide}\}\}$
    $N_{\text{out}}(e) \leftarrow \{y \mid (e \xrightarrow{\rho} y) \in G_{\text{LKG}},\ \rho \in \{\textsc{Amend}, \textsc{Repeal}, \textsc{Replace}, \textsc{Guide}\}\}$
    $G_q \leftarrow G_q \cup \textsc{InducedSubgraph}(\{e\} \cup N_{\text{in}}(e) \cup N_{\text{out}}(e))$ ▷ 1-hop

Fusion and Generation

$\Pi \leftarrow \textsc{FusePrompt}(q, D_q, G_q)$ ▷ Unified, citation-ready prompt
$a_q \leftarrow \mathcal{G}_f(\Pi)$ ▷ LLM generation constrained by evidence
if $\neg\textsc{HasCitations}(a_q)$ **or** $\textsc{EvidenceMismatch}(a_q, D_q, G_q)$ **then**
    return `Unknown Answer` ▷ Refuse unsupported claims
else
    return a_q ▷ Citation-grounded answer

4.3 Retrieval and Generation Phase

After constructing the SBV-Legal Knowledge Base, the system proceeds to the *Retrieval and Generation Phase* to handle user queries (Algorithm 2). This phase focuses on finding the most relevant legal information – both from the document index and the LKG – to generate reliable and well-supported answers. It is driven by three main components: SBV-LR, SBV-RR, and the Fusion and Generation module, which brings everything together into a coherent response.

SBV-LR. This component is responsible for finding the most relevant text in the legal corpus based on a user's query. It uses a hybrid search approach that combines sparse (keyword-based) and dense (semantic) retrieval. We follow the top-performing method from ALQAC 2024 [42] by embedding the user query the `paraphrase-vietnamese-law` model. The resulting vector is then compared against precomputed embeddings in Qdrant [45], enabling retrieval based on both exact terms and deeper semantic similarity.

We introduce a dynamic reranking stage to improve retrieval accuracy. We reorder the candidate documents using Reciprocal Rank Fusion (RRF), the ViRanker cross-encoder [43], and the multilingual `bge-reranker-v2-m3` model [9]. These rerankers collectively enhance result quality by optimizing both semantic relevance and contextual alignment. The final output is a ranked list of the top-k most relevant documents $D_q = \{d_i\}_{i=1}^{k} \subset \mathcal{C}$ for the input query q.

SBV-RR. Although SBV-LR captures which legal documents are related to a query, it does not account for the formal relationships between those documents. The SBV-RR component fills this gap by using the LKG to introduce a structured legal context. It uses a lightweight NER model [54] to detect legal entities mentioned in the query, such as `Circular23/2025/TT-NHNN`. These entities serve as reference points for making queries in the Neo4j graph database [52].

The LKG defines two types of directional relationships: *incoming* links (e.g. laws or decisions that amend the target document) and *outgoing* links (e.g. documents issued under the authority of a given law). To keep results focused and relevant, we limit the graph traversal to one hop in either direction, ensuring that only the most direct and meaningful connections are retrieved, such as amendments, repeals, or enabling regulations.

The final output is a subgraph $G_q \subset G_{\text{LKG}}$ that captures the most relevant legal relationships related to the query. This relational view complements the semantic retrieval from SBV-LR, giving the LLM a more complete legal picture to generate accurate and well-grounded responses.

Fusion and Generation. The retrieved text segments – D_q from SBV-LR – and the relational context – G_q from SBV-RR – are combined with the original user query q to form a unified prompt for the LLM. This fusion gives the model access to both the content of the law and the connections between legal documents – essential for accurate legal reasoning. The generation function $\mathcal{G}_f(q, D_q, G_q)$ produces the final answer a_q, guided by two key principles:

1. **Citation-grounded output:** Each part of the answer must be directly supported by retrieved evidence. Clear citations to source documents or legal relationships are included to ensure transparency and verifiability.
2. **Controlled fallback:** If there is not enough evidence (i.e., little or no D_q or G_q), the system avoids making unsupported claims. Instead, it returns an *"Unknown Answer"*, often with a suggestion on how to refine the query.

By combining both semantic content and legal structure – and refusing to generate unsupported claims – this stage ensures that the generated answers are not only accurate but also trustworthy and explainable.

5 Experimental Setup

This section covers our data, setup, and baselines. We start with Vietnam's banking legal corpus and datasets (Section 5.1), then explain our implementation and models (Section 5.2), and finally describe the evaluation metrics (Section 5.3).

Table 1: Composition and validity of SBV legal documents.

Type	**Effective**	**Repealed**	**Partial-Repealed**	**Yet-Effective**	**Total**
Law	2	6	4	0	12
Ordinance	2	3	0	0	5
Resolution	1	0	0	0	1
Decree	31	36	16	1	84
Decision	276	485	16	0	777
Circular	291	309	154	9	763
Joint Circular	36	24	1	0	61
Total	**639**	**863**	**191**	**10**	**1703**

Table 2: Statistics of the ALQAC2025 and SBV Legal datasets.

Dataset	**Category**	**Quantity**
ALQAC2025 Corpus	Number of documents	15
	Number of question–answer pairs	729
	Questions with one relevant document	718
	Questions with multiple relevant documents	11
SBV Legal Corpus	Number of documents	840
	Number of qumestion–answer pairs	100
	Questions with one relevant document	89
	Questions with multiple relevant documents	11

5.1 SBV Legal Corpus and Evaluation Datasets

We use the Legal Corpus of the State Bank of Vietnam (SBV) consisting of 1,703 regulatory documents sourced from the Vietnamese Database of Legal Documents . These include Laws, Ordinances, Decrees, Decisions, and Circulars, with Decisions and Circulars making up the majority of Vietnam's banking regulations. As shown in Table 1, the corpus reflects a fast-changing legal environment: 863 documents have been fully repealed and 191 partially repealed.

We built an LKG to capture document dependencies, mapping relationships such as amendments or repeals. For example, Circular `23/2025/TT-NHNN` updates Circular `30/2019/TT-NHNN` on compulsory reserves. This helps trace the evolution of legal norms and ensures responses reflect current laws.

For evaluation, we used two datasets (Table 2). A subset of ALQAC2025 Dataset [3] includes 729 QA pairs from 15 documents and measures retrieval performance. The SBV Legal Dataset contains 840 currently effective or partially effective documents, segmented into 9,661 articles and indexed in the LKG, which holds 6,019 relationships among 5,221 nodes. We also collected 100 real-world QA pairs from the Legal Library portal, manually linking each to relevant documents and articles.

5.2 Implementation and Baselines

Implementation Setup. SBV-LawGraph combines sparse and dense retrieval with structured graph reasoning. Dense embeddings are generated using the `minhqu`

`an6203/paraphrase-vietnamese-law-embedding` model, while sparse retrieval is handled by BM25. Retrieved results are fused and re-ranked using the cross-encoder `BAAI/bge-reranker-v2-m3` to improve semantic relevance.

The 840 documents from the SBV Legal Corpus are segmented, embedded, and stored in a Qdrant vector database. An LKG is constructed in Neo4j, with relationships extracted using the `gpt-oss-120b` model and queried using Cipher. This model also supports the answer generation component through OpenAI's API. All components are implemented in Python using the Haystack framework. We set top-$k = 5$ and a cosine similarity threshold of 0.9, which yielded the best balance between recall and precision in our experiments.

Baselines. We benchmarked SBV-LawGraph against four systems. **BM25** serves as a keyword-based lexical retrieval baseline. **NaiveRAG** uses dense embeddings from `paraphrase-vietnamese-law` to compute cosine similarity between queries and documents, without re-ranking. **AdvancedRAG** combines BM25 with dense retrieval via a weighted score (75% BM25, 25% semantic). **GPT-5** and **Gemini 2.5 Pro** are evaluated in zero-shot mode through official APIs to reflect the capabilities of state-of-the-art LLMs with web-enabled retrieval.

5.3 Evaluation Metrics

We evaluate using standard information retrieval metrics –**Recall@k**, **Precision@k**, **Mean Reciprocal Rank (MRR@k)**, and **F2@k** [33,1,59,34] – along with a task-specific measure **Correctness** [31,51], tailored for legal QA.

Retrieval Metrics. Given a query set Q, let R_i be the set of relevant documents for query i, and $\hat{R}_i^k$ the top-k retrieved results. We use the standard recall and precision metrics:

$$\text{Recall@k} = |\hat{R}_i^k \cap R_i|/|R_i|, \qquad \text{Precision@k} = |\hat{R}_i^k \cap R_i|/k$$

To capture how early a correct result appears, we use:

$$\text{MRR@k} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$

Here, rank_i is the position of the first relevant result for query i. To balance precision and recall while emphasizing completeness, we report the **F2-Score**:

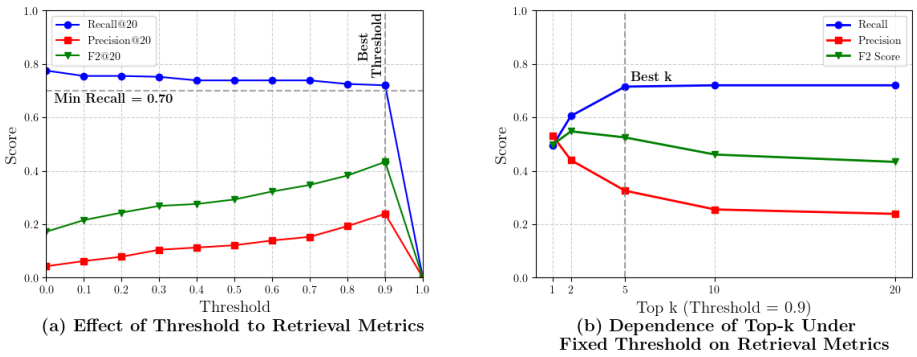
$$\text{F2@k} = 5 \times \frac{\text{Precision@k} \times \text{Recall@k}}{4 \times \text{Precision@k} + \text{Recall@k}}$$

Answer Correctness. We present a **Correctness** score that indicates if an answer is both semantically correct and legally grounded. Let a_i be the generated answer and g_i the gold (reference) answer for question q_i:

$$\text{Correctness} = \frac{1}{N} \sum_{i=1}^{N} \text{Correct}(a_i, g_i)$$

Table 3: Results on the ALQAC2025 dataset and SBV Legal Questions.

Dataset	Model	R@1	R@2	R@5	R@10	R@20	MRR@2	P@2	F2@2
ALQAC Dataset Questions	BM25	0.57	0.65	0.70	0.74	0.75	0.61	0.33	0.54
	Naive RAG	0.36	0.44	0.53	0.58	0.65	0.40	0.22	0.37
	Advanced RAG	0.57	0.65	0.71	0.74	0.76	0.61	0.33	0.54
	SBV-LawGraph	**0.69**	**0.73**	**0.76**	**0.77**	**0.78**	**0.71**	**0.37**	**0.61**
SBV Legal Questions	BM25	0.38	0.51	0.62	0.65	0.70	0.47	0.28	0.43
	Naive RAG	0.32	0.39	0.53	0.61	0.68	0.38	0.21	0.33
	Advanced RAG	0.40	0.52	0.62	0.67	0.70	0.49	0.28	0.44
	SBV-LawGraph	**0.49**	**0.62**	**0.74**	**0.76**	**0.78**	**0.73**	**0.39**	**0.60**



(a) Effect of Threshold to Retrieval Metrics

(b) Dependence of Top-k Under Fixed Threshold on Retrieval Metrics

Fig. 2: Evaluation of Retrieval Performance Across Thresholds and Top-k Values.

To ensure rigor beyond a coarse binary definition, each answer was independently evaluated by two annotators with experience in Vietnamese legal NLP and the fintech domain. Here, An answer was marked correct $\text{Correct}(a_i, g_i) = 1$ only if it met all of the following conditions: (i) Semantic equivalence — the response preserves the core legal meaning without contradictions; (ii) Citation presence — the response includes at least one relevant legal article, clause, or statute; (iii) Citation validity — cited references match actual corpus provisions and are relevant to the question. Otherwise $\text{Correct}(a_i, g_i) = 0$.

6 Results and Discussion

As shown in Table 3, **SBV-LawGraph** achieves state-of-the-art performance across all metrics and datasets, clearly outperforming both sparse and dense retrieval baselines. On the *ALQAC2025* benchmark, it delivers the highest recall at all cutoffs (R@1–R@20), with gains of up to +12% over BM25 and +8% over AdvancedRAG. It also records an MRR@2 of 0.71, indicating strong ranking precision, and an F2@2 of 0.61, reflecting a balanced trade-off between recall and precision. Similar results are seen on the *SBV Legal Questions* dataset, where recall@10 rises from 0.67 (AdvancedRAG) to 0.76, and P@2 improves by 39%, demonstrating the model's robustness across domains.

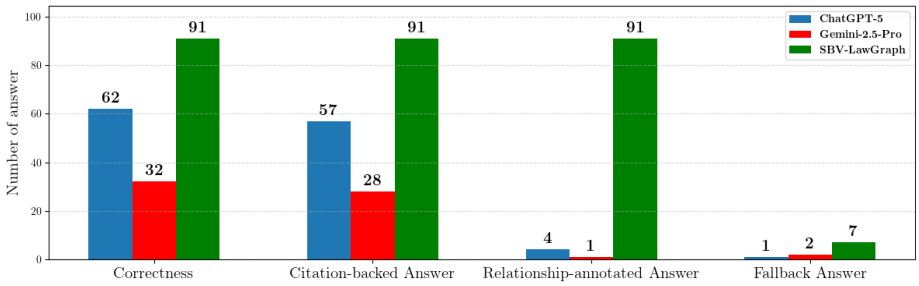


Fig. 3: Comparative performance of ChatGPT-5, Gemini-1.5-Pro, and SBV-LawGraph across various answer types on the SBV Legal Questions dataset.

In our opinion, these results are driven by SBV-LawGraph's hybrid design, which combines BM25-based sparse retrieval, dense semantic embeddings, a reranking model, and graph-based legal reasoning that captures inter-article dependencies and hierarchical relationships between statutes. This integration helps the system retrieve both direct legal references and contextually linked provisions, improving factual grounding and interpretability.

Hyperparameter tuning confirms the effectiveness of this setup. As seen in Fig. 2(a), the optimal cosine similarity threshold is 0.9, maximizing precision and F2 while preserving coverage. With this threshold fixed, top-k values from 1 to 20 were tested (Fig. 2(b)), and $k = 5$ provided the best balance between retrieval accuracy and efficiency, minimizing unnecessary token overhead.

In the generative evaluation (Fig. 3), SBV-LawGraph consistently outperforms GPT-5 and Gemini-2.5-Pro across correctness, citation inclusion, and legal consistency. Thanks to its graph reasoning layer, the model can identify statutory dependencies – like definitions, amendments, and exceptions – enabling it to generate answers that are not only accurate but legally coherent. When evidence is lacking, it falls back to a neutral response to avoid unsupported claims.

7 Conclusion and Future Work

We introduced **SBV-LawGraph**, a unified framework that combines sparse and dense retrieval with knowledge graph reasoning to improve legal question answering. By capturing both lexical and semantic signals and grounding responses in structured legal relationships, SBV-LawGraph delivers accurate, citation-supported answers. Experiments on ALQAC2025 and SBV Legal datasets show consistent improvements over strong baselines across all metrics.

Future work will focus on improving the adaptability, reliability, and legal transparency of SBV-LawGraph: (i) expanding and documenting benchmark datasets with clearer selection criteria, reasoning distribution, and larger scale to reduce bias; (ii) refining evaluation protocols through detailed annotator guidelines, agreement analysis, and controlled baseline tuning, alongside ablation studies isolating SBV-LR and SBV-RR contributions; (iii) assessing knowledge graph quality via manual relation auditing and precision–recall reporting; and (iv) con-

ducting user-centered studies with domain practitioners to evaluate real-world usefulness and trust.

Acknowledgments

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

References

1. Abdallah, A., Piryani, B., Jatowt, A.: Exploring the state of the art in legal qa systems. Journal of Big Data **10**(127) (2023)
2. Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., Lampos, V.: Predicting judicial decisions of the european court of human rights: A natural language processing perspective. PeerJ Comput. Sci. **2**, e93 (2016)
3. ALQAC Organizing Committee: Automated legal question answering competition (alqac 2025). `https://sites.google.com/view/ALQAC-2025` (2025)
4. Althammer, S., Faggioli, G., Savelka, J., Spanakis, G., Dinu, G.: Dossier@coliee 2021: Leveraging dense retrieval for case law retrieval. In: COLIEE (2021)
5. Barron, R.C., Eren, M.E., Serafimova, O.M.: Bridging legal knowledge and ai: Retrieval-augmented generation with vector stores, knowledge graphs, and hierarchical non-negative matrix factorization. arXiv preprint arXiv:2502.20364 (2025)
6. Bommireddy, R., Kancheepuram, I., Akansha, S.: Graph-based agentic retrieval-augmented generation: A comprehensive survey. ResearchGate Preprint (2025)
7. Chalkidis, I., Fergadiotis, M., Androutsopoulos, I., Aletras, N., Tsarapatsanis, D., Malakasiotis, P.: Lexglue: A benchmark dataset for legal language understanding in english. In: Findings of ACL: EMNLP. pp. 119–136 (2021)
8. Chalkidis, I., Firooz, H., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: Legal-bert: The muppets straight out of law school. arXiv preprint (2020)
9. Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., Liu, Z.: Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation (2024)
10. Cheng, M., Luo, Y., Ouyang, J., Liu, Q., Liu, H., Li, L.: A survey on knowledge-oriented retrieval-augmented generation. arXiv preprint arXiv:2503.10677 (2025)
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proc. NAACL-HLT. vol. 1, pp. 4171–4186. Association for Computational Linguistics (2019)
12. Gao, T., Yao, X., Chen, D.: Simcse: Simple contrastive learning of sentence embeddings. In: Proc. EMNLP. pp. 6894–6910. ACL (2021)
13. Gao, W., Kim, J., Ng, V.: Enhancing legal case retrieval via scaling high-quality data. In: Proc. EMNLP (2024)
14. Gao, Y., Xiong, Y., Gao, X., Tang, J., Xie, X.: Retrieval-augmented generation for large language models: A survey. arXiv preprint (2024)
15. Gao, Z., Cao, Y., Wang, H., Ke, A., Feng, Y., Xie, X.: Frag: A flexible modular framework for retrieval-augmented generation based on knowledge graphs. arXiv preprint arXiv:2501.09957 (2025)
16. Guo, Z., Zhao, H., Xu, L., Chen, S., Xu, T.: Lightrag: Simple and fast retrieval-augmented generation. arXiv preprint (2024)
17. Gupta, S., Ranjan, R., Singh, S.N.: A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions. arXiv preprint arXiv:2410.12837 (2024)

18. Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., Liu, T.: A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. arXiv preprint (2023)
19. Huang, Y., Huang, J.: A survey on retrieval-augmented text generation for large language models. arXiv preprint arXiv:2404.10981 (2024)
20. Islam, S., Rahman, M., Hossain, K., Hoque, E.: Open-rag: Enhanced retrieval-augmented reasoning with open-source large language models. arXiv preprint arXiv:2410.01782 (2024)
21. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Madotto, A., Fung, P.: Survey of hallucination in natural language generation. ACM Comput. Surv. **55**(12), 1–38 (2023)
22. Jiang, P., Ouyang, S., Jiao, Y., Zhong, M., Tian, R.: Retrieval and structuring augmented generation with large language models. In: Proc. CIKM (2025)
23. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense passage retrieval for open-domain question answering. In: Proc. EMNLP. pp. 6769–6781 (2020)
24. Lai, J., Conghui, Z., Xiaohan, Z., Fuhui, S.: Leveraging llm-based retrieval-augmented generation for legal knowledge graph completion. In: BigData (2024)
25. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive nlp tasks. In: Adv. Neural Inf. Process. Syst. vol. 33, pp. 9459–9474 (2020)
26. Li, S., Stenzel, L., Eickhoff, C., Bahrainian, S.A.: Enhancing retrieval-augmented generation: A study of best practices. In: ACL (2025)
27. Lima, R.A.: Poly-vector retrieval: Reference and content embeddings for legal documents. arXiv preprint arXiv:2504.10508 (2025)
28. Lin, T., Zhu, Y., Luo, Y., Tang, N.: Srag: Structured retrieval-augmented generation for multi-entity question answering over wikipedia graph. arXiv preprint arXiv:2503.01346 (2025)
29. Louis, J., Kim, H., Kim, S., Yoon, S.: Finding the law: Enhancing statutory article retrieval via graph neural networks. In: EACL. pp. 2754–2769 (2023)
30. Luan, Y., Eisenstein, J., Toutanova, K., Collins, M.: Sparse, dense, and attentional representations for text retrieval. Trans. Assoc. Comput. Linguist. **9**, 329–345 (2021)
31. Martínez-Gil, J.: A survey on legal question answering systems. Computers & Security Review **48**, 100552 (2023)
32. Nguyen, D.Q., Nguyen, A.T.: Phobert: Pre-trained language models for vietnamese. In: Findings of ACL: EMNLP. pp. 1037–1042 (2020)
33. Nguyen, D., Bui, T., Nguyen, L.M., Nguyen, H.T.: Vlqa: The first comprehensive, large, and high-quality vietnamese dataset for legal question answering. arXiv preprint arXiv:2507.19995 (2025)
34. Nguyen, H.T., Tran, V.M., Tu, M.P., Nguyen, L.M.: A legal information retrieval system for statute law. In: COLIEE (2022)
35. Nguyen, M.D., Le, T.H., Tran, V.B.: Hierarchical structure and regulatory complexity in vietnamese legal documents: Challenges for automated processing. Vietnam J. Legal Sci. **15**(3), 45–62 (2023)
36. Nguyen Ba, T., Pham, D.T., Nguyen, T.M., et al.: Vietnamese legal information retrieval in question answering system. arXiv preprint (2024)
37. Nie, P., Lin, M., Ma, W., Cong, L., Yin, R., Zu, M., Wu, Q.: Kgsrag: Retrieval-augmented generation system for biomedical information retrieval and reasoning based on knowledge graphs and statements. TechRxiv Preprint (2025)

38. OpenAI: gpt-oss-120b and gpt-oss-20b model card (2025), https://arxiv.org/abs/2508.10925
39. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Leike, J.: Training language models to follow instructions with human feedback. Adv. Neural Inf. Process. Syst. **35**, 27730–27744 (2022)
40. Peng, B., Zhu, Y., Liu, Y., Zhang, Z., Li, Z.: Graph retrieval-augmented generation: A survey. J. ACM **37**(4) (2024)
41. Peng, C., Xia, F., Naseriparsa, M., Osborne, F.: Knowledge graphs: Opportunities and challenges. Artif. Intell. Rev. **56**, 13071–13102 (2023)
42. Pham, H.Q., Van Nguyen, Q., Tran, D.Q., Nguyen, T.K.B., Van Nguyen, K.: Top 2 at alqac 2024: Large language models (llms) for legal question answering. International Journal of Asian Language Processing (2024)
43. Phuong, N.D.: Viranker: A cross-encoder model for vietnamese text ranking (2024)
44. Procko, T.T., Ochoa, O.: Graph retrieval-augmented generation for large language models: A survey. In: Proc. AIxSET. pp. 166–169 (2024)
45. Qdrant Team: Qdrant: Open-source vector database (2024), https://github.com/qdrant/qdrant
46. Quan, T.T.: Modern approaches in natural language processing. VNU Journal of Science: Computer Science and Communication Engineering **39**(1), 31–55 (2021)
47. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proc. EMNLP. Association for Computational Linguistics (11 2019)
48. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M.: Okapi at trec-3. In: Proc. TREC-3. pp. 109–126. NIST (1994)
49. Singh, A., Ehtesham, A., Kumar, S., Khoei, T.T.: Agentic retrieval-augmented generation: A survey on agentic rag. arXiv preprint arXiv:2501.09136 (2025)
50. Sovrano, F., Palmirani, M., Vitali, F.: Legal knowledge extraction for knowledge graph based question-answering. In: FAIA. pp. 183–190 (2020)
51. Trautmann, D., Tkachuk, Y., Risch, J.: Measuring the groundedness of legal question-answering systems. In: NLLP (2024)
52. Webber, J.: A programmatic introduction to neo4j. In: SPLASH. pp. 217–218 (2012)
53. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q., Zhou, D., Li, X., Song, D.: Finetuned language models are zero-shot learners. In: Proc. ICML. vol. 162, pp. 18477–18494 (2022)
54. Wolf, T., Debut, L., Sanh, V., Chaumond, J., et al.: Transformers: State-of-the-art natural language processing. In: Proc. EMNLP. pp. 38–45. ACL (2020)
55. Zalo AI Challenge: Zalo ai challenge 2021: Vietnamese legal text processing and question answering. In: Proceedings of the 2021 Zalo AI Challenge. VNG Corporation, Ho Chi Minh City, Vietnam (2021), https://www.kaggle.com/datasets/hariwh0/zaloai2021-legal-text-retrieval
56. Zhao, P., Zhang, H., Yu, Q., Wang, Z., Geng, Y., Fu, F.: Retrieval-augmented generation for ai-generated content: A survey. arXiv:2402.19473 (2024)
57. Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., Sun, M.: How does nlp benefit the legal system: A summary of legal artificial intelligence. In: Proc. EMNLP. pp. 5218–5230 (2020)
58. Zhu, G., Hao, M., Zheng, C., Wang, L.: Design of knowledge graph retrieval system for legal and regulatory framework of multilevel latent semantic indexing. J. Math. **2022**, 1–12 (2022)
59. Zhu, J., Wu, J., Luo, X., Liu, J.: Semantic matching based legal information retrieval system for covid-19 pandemic. Artif. Intell. Law **32**, 397–426 (2024)