Rethinking Legal Retrieval as Generative Article Identifier Prediction

Anonymous ACL submission

Abstract

Legal article retrieval requires mapping concrete factual disputes to abstract statutory norms, not merely matching semantically similar texts. This fact-to-norm gap makes lexical or embedding-based retrieval unreliable. We propose a simple alternative: generative statutory article identifier prediction. Given a legal question, we use a large language model (LLM) to predict canonical article identifiers, such as *Civil Code Article 243*, by leveraging its parametric legal memory and reasoning ability. The predicted identifiers are then bound to authoritative statutory texts as retrieval results. Moreover, we introduce a self-supervised statute knowledge injection strategy that teaches LLMs to associate statutes with their identifiers, without annotated query-document pairs or positive-negative samples required by common embedding-based retriever training. Experiments show that LLMs are strong zero-shot generative retrievers for legal articles, and that knowledge injection further improves recall and ranking quality.[1]

1 Introduction

Legal article retrieval is a core task for evidence-grounded legal reasoning. Given a legal question, a system needs to retrieve several statutory provisions that support subsequent legal analysis. This task is important because legal assistance cannot rely only on free-form generation: even strong large language models (LLMs) may generate legal provisions that deviate from authoritative statutes (Ji et al., 2023; Maynez et al., 2020; Shuster et al., 2021). Recent legal reasoning and retrieval-augmented generation (RAG) benchmarks also show that reliable legal systems require accurate retrieval of supporting legal sources (Fei et al., 2024; Pipitone and Alami, 2024; Li et al., 2025).

[1] https://anonymous.4open.science/r/GenArtID-review-D5D0.

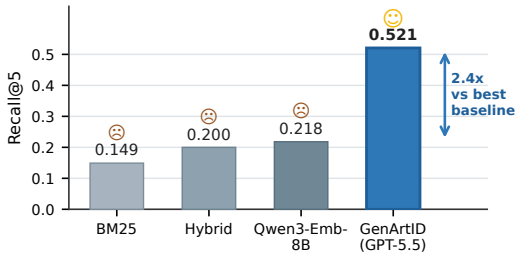

Figure 1: Legal retrieval performance on LexRAG.

Existing retrieval methods mainly follow the conventional RAG paradigm, where a query is matched against the legal corpus by lexical or representation-space similarity (Lewis et al., 2020; Robertson and Zaragoza, 2009; Karpukhin et al., 2020; Chen et al., 2024; Zhang et al., 2025). However, legal article retrieval has a special gap: users describe concrete facts, while statutes express abstract legal norms. For example, a user may describe a neighbor's renovation causing water leakage and compensation loss, while the relevant provision may be written as a general tort-liability rule. Such fact-to-norm mismatch makes both keyword retrieval and dense retrieval unreliable.

In this paper, we propose **GenArtID**, a simple generative paradigm for legal article retrieval. Instead of directly searching the legal corpus, we ask the LLM to generate legal article identifiers from the input question. The generated identifiers are then parsed, normalized, and linked to authoritative statutory texts through a legal database. Surprisingly, this simple paradigm is highly effective. As shown in Figure 1, we find GenArtID substantially outperforms conventional retrievers. This result suggests that LLMs can use parametric legal knowledge and implicit reasoning to bridge the fact-to-norm gap effectively.

We further introduce self-supervised statute knowledge injection to make article-level legal knowledge more easily activated during generation. Legal statutes naturally provide identifier–text

pairs: each article has a canonical identifier and an official text. We train the model to map identifiers to article texts and article texts back to identifiers with parameter-efficient tuning (Hu et al., 2022; Dettmers et al., 2023). Different from dense retriever training, this process does not require manually annotated query–document pairs or positive–negative sample construction. Experiments show that this lightweight self-supervised training further improves retrieval performance.

Our highlights are summarized as follows: 1) We propose a simple generative paradigm for legal article retrieval. 2) We introduce a self-supervised learning approach to further enhance article-level statutory knowledge activation. 3) Extensive experiments demonstrate the effectiveness of our method and the benefit of knowledge injection.

2 Related Work

Legal retrieval and RAG have been widely studied (Chalkidis et al., 2020; Zhong et al., 2020; Li et al., 2024). Conventional RAG systems usually retrieve external evidence before generation (Lewis et al., 2020), using sparse lexical methods such as BM25 (Robertson and Zaragoza, 2009), dense retrievers (Karpukhin et al., 2020), late-interaction models (Khattab and Zaharia, 2020), or stronger embedding models such as BGE-M3 and Qwen3-Embedding (Chen et al., 2024; Zhang et al., 2025). However, training effective embedding-based retrievers often depends on carefully constructed positive and negative pairs for contrastive learning, which introduces substantial data construction and annotation challenges (Karpukhin et al., 2020; Xiong et al., 2021; Qu et al., 2021). Hierarchical and graph-based methods further organize long corpora or relational paths for retrieval (Sarthi et al., 2024; Chen et al., 2026). Late-interaction and reranking methods provide stronger retrieval or ranking models (Santhanam et al., 2022; Nogueira and Cho, 2019). Another related line is generative retrieval, where models retrieve items by generating identifiers, such as entity names or document IDs (De Cao et al., 2021; Tay et al., 2022). Autoregressive search engines also generate substrings or identifiers for retrieval (Bevilacqua et al., 2022).

3 Method

3.1 Task Formulation

Let the legal corpus be $\mathcal{L} = \{(a_i, t_i)\}_{i=1}^{N}$, where a_i denotes the identifier of the i-th legal article and t_i denotes its authoritative text. Given a legal question q, the target evidence set is $G(q) \subseteq \{a_1, \ldots, a_N\}$. The task is to retrieve a ranked top-k candidate set

$$C_k(q) = \{(a_{q,1}, t_{q,1}), (a_{q,2}, t_{q,2}), \ldots, (a_{q,k}, t_{q,k})\}$$

such that the candidates cover as many gold articles in $G(q)$ as possible.

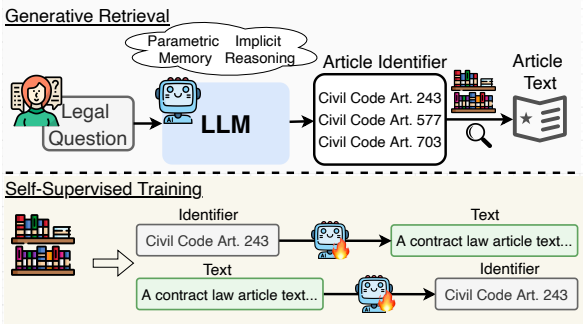

Figure 2: Overview of GenArtID.

3.2 Generative Identifier Prediction

GenArtID performs retrieval in two steps. First, the LLM $\mathcal{M}$ generates a ranked list of article identifiers from the question, with the full prompt provided in Appendix A:

$$\{a_{q,1}, a_{q,2}, \ldots, a_{q,k}\} = \mathcal{M}_\theta(q, k)$$

Second, the generated identifiers are parsed and linked to the legal database:

$$T_k(q) = \{t_i \mid a_i \in C_k(q), (a_i, t_i) \in \mathcal{L}\}$$

This design separates candidate selection from textual verification. The LLM is responsible for producing legal article pointers, while the database provides authoritative statutory texts.

3.3 Self-Supervised Knowledge Injection

To further strengthen the model's callable memory of legal articles, we train GenArtID-SSL directly on statute texts. For each article (a_i, t_i), we create self-supervised instruction examples that ask the model to produce the official text t_i given a_i and to produce the canonical identifier a_i given t_i. These examples form the training set $\mathcal{D}$. The model is trained with the causal language modeling loss:

$$\mathcal{L} = -\sum_{(x,y)\in\mathcal{D}} \sum_{m=1}^{|y|} \log p_\theta(y_m \mid x, y_{<m})$$

where x and y denote the input and target of the model. We also apply a lightweight format alignment stage, following instruction-tuning practice,

Method	LexRAG P@5	LexRAG R@5	LexRAG F1@5	LawBench P@5	LawBench R@5	LawBench F1@5
BM25	0.0338	0.1495	0.0542	0.0383	0.1311	0.0565
RAPTOR (BGE-M3)	0.0419	0.1809	0.0667	0.0569	0.1836	0.0825
Hybrid (BM25 + Qwen3-Emb-8B)	0.0457	0.1995	0.0731	0.0601	0.2032	0.0888
LinearRAG (BGE-M3)	0.0459	0.1993	0.0732	0.0647	0.2096	0.0942
Embedding Retrieval (Qwen3-Emb-8B)	0.0492	0.2178	0.0789	0.0779	0.2338	0.1120
GenArtID (Qwen3-8B)	0.0830	0.3516	0.1316	0.0979	0.3096	0.1413
GenArtID (DeepSeek V4 Flash)	0.1171	0.5004	0.1860	0.1420	0.4303	0.2029
GenArtID (Claude Opus 4.6)	0.1209	0.5187	0.1922	0.1479	0.4527	0.2119
GenArtID (GPT-5.5)	**0.1228**	**0.5207**	**0.1946**	**0.1490**	**0.4546**	**0.2130**

Table 1: Performance comparison on LexRAG and LawBench.

Legal Text	Model	P@5↑	R@5↑	F1@5↑	Avg. Rank↓	Sample Hit@1↑	Sample Hit@3↑	Sample Hit@5↑
Civil Code	GenArtID (Qwen3-8B)	0.0853	0.3631	0.1353	1.78	0.235	0.355	0.404
Civil Code	GenArtID-SSL (Qwen3-8B)	**0.1001** +0.0148	**0.4100** +0.0469	**0.1560** +0.0207	**1.70** -0.08	**0.277** +0.042	**0.413** +0.058	**0.446** +0.042
Crim. Proc. Law	GenArtID (Qwen3-8B)	0.0708	0.3125	0.1141	**1.12**	**0.292**	0.333	0.333
Crim. Proc. Law	GenArtID-SSL (Qwen3-8B)	**0.0969** +0.0261	**0.4167** +0.1042	**0.1552** +0.0411	1.78 +0.66	0.229 -0.063	**0.438** +0.105	**0.479** +0.146

Table 2: Training effectiveness of self-supervised knowledge injection.

where the target output is a JSON-formatted identifier list (Wei et al., 2022; Ouyang et al., 2022). Our training uses only natural identifier–text pairs and requires no manually constructed retrieval pairs.

4 Experimental Setup

Dataset The legal database used for article-text binding is the open-source LexRAG legal corpus[2]. It contains 222 Chinese statutes and regulations and provides a structured retrieval corpus of 17,228 legal articles. We conduct experiments on LexRAG (Li et al., 2025) and LawBench (Fei et al., 2024). LexRAG contains 1,013 legal consultation questions. From the legal-question subset of LawBench, 439 items are retained after matching their supporting articles against our legal database.

Metrics We report Precision@5 (P@5), Recall@5 (R@5), and F1@5, which are macro-averaged over samples. For knowledge-injection experiments, we also report Average Correct Rank (Avg. Rank) and Sample Hit@1/3/5 to measure ranking quality.

Baselines 1) BM25, a sparse lexical retriever based on the probabilistic relevance framework (Robertson and Zaragoza, 2009). 2) RAPTOR, which retrieves from a recursively clustered and summarized hierarchy of the legal corpus (Sarthi et al., 2024). 3) LinearRAG, a linear graph-based retrieval pipeline with BGE-M3 embeddings (Zhuang et al., 2026; Chen et al., 2024). 4) Embedding Retrieval, which retrieves nearest articles with Qwen3-Embedding-8B (Zhang et al., 2025). 5) Hybrid Retrieval, which linearly combines BM25 and Qwen3-Embedding-8B scores with $\alpha = 0.5$.

Implementation Details For knowledge injection, we fine-tune Qwen3-8B (Yang et al., 2025) with QLoRA (Dettmers et al., 2023), using rank $r = 32$ and $\alpha = 64$ on a single NVIDIA RTX 5090 GPU. We construct 3,136 training examples in total. More implementation details are provided in Appendix B.

5 Results and Discussion

Our Generative article prediction substantially outperforms conventional retrieval. As shown in Table 1, GenArtID consistently outperforms retrieval-based baselines on both LexRAG and LawBench. Dense retrieval is the strongest conventional baseline, but its performance remains far below our method. Among LLM backends, stronger closed models achieve the best results, while Qwen3-8B still outperforms all retrieval baselines, showing that the method is not tied to a single proprietary model.

Knowledge injection further improve the performance of Generative Retrieval. Table 2 compares GenArtID based on Qwen3-8B with GenArtID-SSL, the LoRA-tuned model after knowledge injection. The results show that knowledge injection improves retrieval coverage by generating more correct article identifiers within the top-5 candidates, and generally improves ranking

[2] https://github.com/CSHaitao/LexRAG

Model	Mode	P@5	R@5	F1@5	Corr.	Part.	Wrong
DeepSeek	ID-Only	**0.1171**	**0.5004**	**0.1860**	–	–	–
DeepSeek	ID+Content	0.1148	0.4842	0.1816	0.906	0.075	0.019
Qwen3-8B	ID-Only	**0.0830**	**0.3516**	**0.1316**	–	–	–
Qwen3-8B	ID+Content	0.0824	0.3513	0.1308	0.401	0.473	0.125

Table 3: Ablation of generating article IDs only versus generating both article IDs and statute text on LexRAG.

Civil Code Case 250	
Question	Goods damaged during carrier transport; compensation only 1,000 RMB for loss of 50,000 RMB.
Gold	Civil Code Article 832: cargo damage liability.
GenArtID	Art. 803–807: carrier-related provisions, but none matches the gold article. ✗
GenArtID-SSL	**Art. 832** (rank 1), Art. 833, 584, 585, 586. ✓

Table 4: Representative case study of knowledge injection. An additional case study is provided in Appendix C.2.

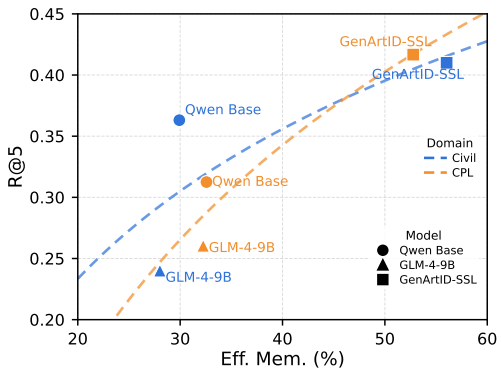

Figure 3: Relationship between effective parametric memory and retrieval recall. Dashed curves denote logarithmic auxiliary fits by legal domain.

quality by placing relevant articles at higher positions.

Stronger parametric memory leads to better retrieval. Figure 3 visualizes the relationship between effective statute memory and Recall@5 using the parametric-memory results in Table 7. We measure effective memory by asking the model to recite the statutory text given an article identifier, and count it as effective when the output matches the target article above a fixed cosine-similarity threshold. Each point denotes a model–legal-text pair. GenArtID-SSL shifts the Qwen3-8B points toward higher memory and higher R@5. The overall trend supports our motivation: GenArtID benefits from parametric legal knowledge, and self-supervised statute training helps activate such knowledge as canonical article identifiers.

Generating identifiers is more effective and safer than generating legal content. Table 3 compares two prompting modes on LexRAG: ID-Only generates only canonical article identifiers, while ID+Content generates both identifiers and statutory texts. For ID+Content, we further measure content fidelity against official article texts using cosine similarity, where Corr., Part., and Wrong denote correct, partially correct, and incorrect generations. ID-Only achieves slightly better retrieval performance for both DeepSeek and Qwen3-8B. More importantly, ID+Content is not fully reliable: even when the identifier is useful, the generated statutory text may be incomplete or deviate from the official provision, especially for Qwen3-8B. This supports our design choice: GenArtID uses the LLM to predict article identifiers, but obtains final legal evidence through database binding rather than free-form legal content generation.

Knowledge injection helps distinguish neighboring articles. Table 4 shows a representative case where the base model misses the gold article but GenArtID-SSL retrieves the correct article at rank 1. For the cargo-damage dispute, the base model generates thematically related carriage-contract provisions, but none matches the exact applicable article. After knowledge injection, GenArtID-SSL recovers Civil Code Article 832, suggesting that knowledge injection improves article-level discrimination and helps the model select the precise statutory provision from neighboring articles within the same topic. An additional Criminal Procedure Law case is provided in Appendix C.2, where GenArtID-SSL similarly recovers the gold reporting-and-filing article from neighboring detention and investigation provisions.

6 Conclusion

We propose GenArtID, a simple generative framework for legal article retrieval. Instead of retrieving legal texts by lexical or embedding similarity, GenArtID first generates statutory article identifiers and then links them to authoritative legal texts. Experiments show that GenArtID consistently outperforms a series of baselines. We further show that self-supervised knowledge injection strengthens the activation of article-level statutory knowledge and improves both retrieval coverage and ranking quality. Overall, our results suggest that generative article identifier prediction is a simple and effective alternative to conventional retrieval for bridging the fact-to-norm gap in legal retrieval.

Limitations

This work has several limitations. First, our experiments are limited to Chinese legal corpora, mainly LexRAG and LawBench. Due to space constraints, we leave evaluation on more languages, jurisdictions, and legal datasets to future work. Second, GenArtID is most suitable for stable and identifier-rich corpora. For private, newly updated, or rarely exposed corpora, LLMs may lack sufficient parametric knowledge to generate reliable identifiers, so sparse, dense, or hybrid retrieval may still be needed. Third, generated identifiers can still be invalid, outdated, or mismatched, making database validation and fallback retrieval necessary in practical systems.

References

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Wen-tau Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. In *Advances in Neural Information Processing Systems*, volume 35, pages 31668–31683.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Ion Androutsopoulos, and Nikolaos Aletras. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904. Association for Computational Linguistics.

Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. 2026. PathRAG: Pruning graph-based retrieval augmented generation with relational paths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 30183–30191.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-Embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Alan Huang, Songyang Zhang, Kai Chen, Zhixin Yin, Zongwen Shen, Jidong Ge, and Vincent Ng. 2024. LawBench: Benchmarking legal knowledge of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7933–7962. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Haitao Li, Yifan Chen, Yiran Hu, Qingyao Ai, Junjie Chen, Xiaoyu Yang, Jianhui Yang, Yueyue Wu, Zeyang Liu, and Yiqun Liu. 2025. LexRAG: Benchmarking retrieval-augmented generation in multi-turn legal consultation conversation. *arXiv preprint arXiv:2502.20640*.

Haitao Li, Yunqiu Shao, Yueyue Wu, Qingyao Ai, Yixiao Ma, and Yiqun Liu. 2024. LeCaRDv2: A large-scale chinese legal case retrieval dataset. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2251–2260.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919. Association for Computational Linguistics.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

Nicholas Pipitone and Ghita Houir Alami. 2024. LegalBench-RAG: A benchmark for retrieval-augmented generation in the legal domain. *arXiv preprint arXiv:2408.10343*.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 5835–5847. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Efficient and effective retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 3715–3734. Association for Computational Linguistics.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803. Association for Computational Linguistics.

Yi Tay, Vinh Q. Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, and Donald Metzler. 2022. Transformer memory as a differentiable search index. In *Advances in Neural Information Processing Systems*, volume 35, pages 21831–21843.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does NLP benefit legal system: A summary of legal artificial intelligence. *arXiv preprint arXiv:2004.12158*.

Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. 2026. LinearRAG: Linear graph retrieval augmented generation on large-scale corpora. In *International Conference on Learning Representations*.

A Full Prompts and Implementation Details

A.1 Prompt Format

All GenArtID experiments use the prompts shown below under deterministic decoding (temperature 0). The ID-Only setting asks the model to return exactly five legal article identifiers without accompanying text. The ID+Content setting additionally asks for the full article content, but retrieval evaluation is still conducted at the article-identifier level; content is only used for the fidelity analysis in Section 4.

A.1.1 ID-Only Prompt

```
You are a professional Chinese
legal retrieval expert.

Your task is not to answer,
but to generate candidate
legal articles for subsequent
database retrieval.  Given
the user's case description,
output the 5 most relevant legal
article identifiers, ordered
by relevance from highest to
lowest.
```

```
Strict requirements: 1. Must output exactly 5 articles; 2. Each must be in the format "Law Name Article XXX"; 3. Do not output explanations, analysis, reasoning, numbering, or any other text; 4. Even if uncertain, you must provide the 5 most likely candidate articles; 5. Prioritize the most directly applicable articles; 6. Output only JSON, nothing else.

Output format: {"articles": ["Law Name Article XXX", ...]}
```

A.1.2 ID+Content Prompt

```
You are a professional Chinese legal retrieval expert.

Your task is not to answer, but to generate candidate legal articles for subsequent database retrieval. Given the user's case description, output the 5 most relevant legal article identifiers together with their full text, ordered by relevance from highest to lowest.

Strict requirements: 1. Must output exactly 5 articles; 2. Each must be in the format "Law Name Article XXX: content"; 3. Do not output explanations, analysis, reasoning, numbering, or any other text; 4. Even if uncertain, you must provide the 5 most likely candidate articles; 5. Prioritize the most directly applicable articles; 6. Output only JSON, nothing else.

Output format: { "articles": [ "Law Name Article XXX: content", ... ] }
```

For domain-specific knowledge injection on the Civil Code and Criminal Procedure Law, the system prompt instructs the model to restrict its output to the target legal corpus. Decoding uses temperature 0 for all LLM backends.

A.2 Parsing and Normalization

Model outputs are processed in two stages. We first apply strict JSON parsing and extract the `articles` field. If JSON parsing fails, we use a regular-expression fallback to extract article identifiers. Each identifier is normalized by removing spaces, normalizing parentheses and full-width digits, converting Chinese numerals to Arabic numerals, and removing duplicate identifiers while preserving their first occurrence. Only the first $k = 5$ valid identifiers are retained.

Model	Correct	Partial	Wrong
DeepSeek V4 Flash	90.6%	7.5%	1.9%
Qwen3-8B	40.1%	47.3%	12.5%

Table 5: Content fidelity of ID+Content outputs on LexRAG (cosine similarity, same thresholds as the memory evaluation).

A.3 Content Fidelity Evaluation

To assess whether generated article text can replace database verification, we evaluate the content fidelity of ID+Content outputs. For each predicted article, we look up its authoritative text in the legal database and compute cosine similarity between the generated text and the ground-truth text using `shibing624/text2vec-base-chinese`, the same model used in the memory evaluation.

- Correct: cosine similarity ≥ 0.85.
- Partial: $0.60 \leq$ cosine similarity < 0.85.
- Wrong: cosine similarity < 0.60.

Even for DeepSeek V4 Flash, 1.9% of generated article texts fall below the correct threshold, confirming that generated legal text is not a substitute for authoritative database binding. For Qwen3-8B, only 40.1% of outputs reach the correct threshold, further indicating that smaller models cannot reliably generate full statutory text.

A.4 Recitation Prompt (Parametric Memory)

The parametric memory evaluation uses the following prompt to ask the model to recite a specific statute article from its identifier. The resulting memory coverage is reported in Table 7.The same prompt format is used for all models and legal texts.

```
[System] You are now to recite the full text of each article from the {law_name}. Output only the body text of the specified article. Do not output titles, tables of contents, annotations, explanations, sources, or any extra text. Follow official punctuation and line-breaking conventions where possible.

[User] Recite the full text of Article {article_no} of the {law_name}. Output only the body text.
```

A.5 Knowledge Injection Training Templates

Stage-1 knowledge injection (Section 3.3 in the main text) uses SFT data constructed from balanced bidirectional mappings. For each statute article (a_i, t_i) in the target legal corpus, we create examples from the templates below. The forward templates train the model to produce article content from an article identifier; the reverse templates train the model to map natural-language legal descriptions and article content back to canonical article identifiers.

A.5.1 Forward Templates (Article ID → Content)

- `Accurately reproduce the content of {name}.`
- `What does {name} stipulate?`
- `Retrieve the statute: what is the full content of {name}?`
- `Recite {name}.`

A.5.2 Reverse Templates (Content → Article ID)

- `Given the statute text: "{content}". Which article does this come from?`
- `Which law and article does the following provision belong to? Provision: {content}`
- `Given the content, provide the corresponding legal article identifier: {content}`
- `Which article of which law does the content "{content}" belong to?`
- `Which article stipulates "{clause}"?`
- `Given the following legal excerpt, provide the corresponding article identifier: {clause}`

The system prompt for all training examples is:

```
/no_think

You are a professional Chinese
legal recitation assistant.
Output the statute content
exactly as requested by the
user. Do not add explanations.
```

A.6 Format Alignment Prompt

After Stage-1 knowledge injection, a small set of 10 legal consultation questions with manually completed top-5 article annotations is used for Stage-2 format alignment (REPEAT_S2 = 2). The prompt below teaches the fine-tuned model to produce structured JSON output under the retrieval protocol without distorting the injected legal knowledge.

```
You are a professional Chinese
legal retrieval expert.

Your task is not to answer,
but to generate candidate
legal articles for subsequent
database retrieval. Given
the user's case description,
output the 5 most relevant legal
article identifiers, ordered
by relevance from highest to
lowest.

Strict requirements: 1. Must
output exactly 5 articles; 2.
Each must be in the format
"Law Name Article XXX"; 3.
Do not output explanations,
analysis, reasoning, numbering,
or any other text; 4. Even
if uncertain, you must provide
the 5 most likely candidate
articles; 5. Prioritize
the most directly applicable
articles; 6. Output only JSON,
nothing else.

Output format: {"articles":
["Law Name Article XXX", ...]}
```

B Implementation Details

For LLM inference, we use deterministic decoding with temperature 0 and a unified JSON-format prompt. Outputs are parsed by strict JSON parsing first and regular-expression extraction second; Chinese and Arabic numerals are normalized before matching. For knowledge injection, Qwen3-8B is fine-tuned with QLoRA (Dettmers et al., 2023) using 4-bit NF4 quantization, rank $r = 32$, and $\alpha = 64$ on a single NVIDIA RTX 5090 GPU. Training consists of article-level knowledge injection followed by lightweight format alignment. The injection data use balanced bidirectional pairs: one article-ID-to-content example and one content-to-article-ID example for each statute article. Thus, the Civil Code has 1,260 articles and yields $1{,}260 \times 2 = 2{,}520$ examples, while the Criminal Procedure Law has 308 articles and yields $308 \times 2 = 616$ examples. This 1:1 construction teaches both recognition of article content and normalization into canonical article identifiers.

Knowledge Injection Hyperparameters . For each legal article (a_i, t_i), we construct two self-supervised mappings: identifier-to-text $(a_i \rightarrow t_i)$ and text-to-identifier $(t_i \rightarrow a_i)$. The first mapping encourages statutory content memorization,

Parameter	Value
Base model	Qwen3-8B
Quantization	4-bit NF4
LoRA rank	32
LoRA alpha	64
LoRA dropout	0.05
Target modules	q_proj, v_proj, o_proj
Learning rate	1×10^{-4}
Batch size	2
Gradient accumulation	8
Stage-1 epochs	5
Stage-2 epochs	1
Max sequence length	768
Scheduler	cosine
Optimizer	AdamW (8-bit)
Precision	bfloat16
GPU	1 NVIDIA RTX 5090

Table 6: QLoRA knowledge injection hyperparameters.

while the second directly trains the model to recover canonical article identifiers from legal semantics. After this stage, a small format-alignment set teaches the model to output a JSON-formatted top-5 identifier list.

C More Results

C.1 Domain-Specific Retrieval Results

Table 7 reports the domain-specific memory coverage and top-5 retrieval results for the Civil Code and Criminal Procedure Law subsets. These supplementary results show that knowledge injection improves effective memory and retrieval quality in both domains.

Legal Text	Model	Eff.	Dup.	Err.	P@5	R@5	F1@5
Civil Code	Qwen3-8B Base (4-bit)	0.2992	0.4667	0.2341	0.0853	0.3631	0.1353
Civil Code	GLM-4-9B	0.2802	0.4944	0.2254	0.0593	0.2401	0.0923
Civil Code	Qwen3-8B LoRA (ckpt-280)	0.5603	0.2587	0.1810	0.1001	0.4100	0.1560
Crim. Proc. Law	Qwen3-8B Base (4-bit)	0.3257	0.4528	0.2215	0.0708	0.3125	0.1141
Crim. Proc. Law	GLM-4-9B	0.3225	0.4007	0.2769	0.0583	0.2604	0.0942
Crim. Proc. Law	Qwen3-8B LoRA (ckpt-180)	0.5277	0.3941	0.0782	0.0969	0.4167	0.1552

Table 7: Parametric memory coverage and retrieval performance. Retrieval columns are evaluated on the corresponding legal-domain subsets and are not directly comparable to the full LexRAG results in Table 1.

C.2 Additional Case Study

We provide an additional case study to further support the qualitative observation in the main text. The case shows that knowledge injection is also effective in the Criminal Procedure Law domain.

Criminal Procedure Law Case 207	
Question	Suspected drunk driving reported; the suspect was not caught at the scene. What procedures follow?
Gold	Criminal Procedure Law Article 85: criminal reporting and case filing.
GenArtID	Art. 82, 91, 113, 114, and Criminal Law Art. 133: related to detention or investigation, but miss the reporting article. ✗
GenArtID-SSL	**Art. 85** (rank 1), Art. 108, 92, 93, 116. ✓

Table 8: Additional case study of knowledge injection in the Criminal Procedure Law domain.

D The Use of Large Language Models

Using an LLM to help with paper writing During the preparation of this work, the authors utilized Large Language Models for language polishing, improving the structural clarity of the manuscript, and refining the formal expression of individual sentences. The use of Large Language Models did not influence the substantive content of the study and served solely as a writing aid.