When the Correct Law Still Fails: A Diagnostic Benchmark for Law-in-Force Retrieval in Versioned Legal RAG

Yuchen Yao[1] **Yiran Hu**[2] **Kaixiang Zhang**[2]
Yixin Su[2] **Jiani Ge**[2] **Yun Liu**[3]*
[1]Department of Computer Science and Technology
[2]School of Law [3]Institute for Studies on AI and Law
Tsinghua University, Beijing, China
yaoyc23@mails.tsinghua.edu.cn liuyun89@tsinghua.edu.cn

Abstract

Legal RAG systems often retrieve passages by semantic relevance, lexical score, or citation match. In versioned legal corpora, however, a passage can be topically relevant and citation-matched while being legally invalid for the operative date. We study *law-in-force retrieval*: retrieving evidence that matches the authority, is valid at the operative date, and supports the answer. We build LawInForce-eCFR from 805 human-validated eCFR/CFR amendment pairs and 3,220 controlled answer-flipping yes/no questions. We also present AuthorityFirst-RAG, a transparent diagnostic retriever that decomposes retrieval into authority identification, temporal scoping, and rule-support ranking. AuthorityFirst-RAG improves Recall@1 from 2.80% for BM25-date-filter to 30.90%, but generation remains difficult even with GoldOracle evidence. These results show a dual bottleneck in versioned legal RAG: retrieving the law in force and applying numeric legal rules.

1 Introduction

Retrieval-augmented generation (RAG) grounds language-model outputs in retrieved evidence (Lewis et al., 2020) and has become a common architecture for legal question answering, retrieval, and regulatory analysis (Ma et al., 2021; Hou et al., 2025; Kalra et al., 2024). Existing legal NLP and legal retrieval work often evaluates broad legal understanding, topical relevance, citation-like authority, or evidence retrieval (Chalkidis et al., 2022; Guha et al., 2023; van Opijnen and Santos, 2017). These objectives are important, but they do not by themselves establish that a retrieved passage is legally applicable.

Versioned law exposes a stricter requirement. A passage may be semantically relevant, citation-matched, and located in the correct section, yet still be legally invalid. Legal rules are amended over time, while old and current versions often coexist in the same corpus or index. For a legal question, the operative date is not ancillary metadata; it determines whether a rule can govern the facts at all. The central claim of this paper is therefore that versioned legal RAG requires retrieving legally applicable authority, not merely relevant text.

This distinction matters because wrong-version evidence can have near-zero utility. In many domains, a topically related document may still provide background information. In legal application, however, an inapplicable version can support the opposite answer with high surface plausibility. A retrieved passage can look correct to both the retriever and the generator because it shares the same citation, heading, vocabulary, and paragraph structure as the valid rule. Its defect is not topical irrelevance; its defect is legal invalidity.

For example, suppose an older regulation required an application to be filed within 60 days, while a later amendment changed the period to 90 days. If the applicant filed on day 75, the answer is *no* under the old rule and *yes* under the new rule. Both versions may mention the same application, use the same section number, and differ only in a local numeric threshold. A retriever that returns the later version for an older operative date has found the right topic and citation, but the wrong legally applicable evidence. The example is diagnostic precisely because lexical, semantic, and citation signals point toward both versions while only one is authority for the facts.

This paper studies that failure mode directly. Legal time is not merely a factual timestamp; it is a normative applicability condition. We define **law-in-force retrieval** as retrieving evidence that belongs to the relevant authority, is valid at the operative date, and contains the rule support needed to answer the question. The retrieval target is therefore authority-validity-support evidence rather than

*Corresponding author.

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 879–893
November 6-10, 2026 ©2026 Association for Computational Linguistics

semantic relevance alone.

We construct LawInForce-eCFR, a diagnostic benchmark derived from real amendments in versioned eCFR/CFR regulations. The benchmark starts from legal changes rather than naturally collected user questions. This is intentional: by aligning old and new versions of the same rule, extracting numeric threshold changes, and placing fact values between the old and new thresholds, we create controlled answer flips. The design isolates the wrong-version failure mode: an incorrect passage can be topically close and citation-wise close while still being invalid for the facts.

LawInForce-eCFR contains 805 human-validated amendment pairs and 3,220 yes/no questions across cited and uncited formulations. We also implement AuthorityFirst-RAG, a transparent diagnostic retriever that separates authority identification, temporal scoping, section-first retrieval, and rule-support ranking. Retrieval improves sharply at the top rank, but generation improves much less, revealing a dual bottleneck: systems must retrieve the law in force and apply its numeric rule.

This setting is distinct from neighboring problems. Temporal QA asks time-specific factual questions, whereas law-in-force retrieval asks whether a legal norm was valid authority for a fact pattern. Citation-based legal retrieval can still fail because the same citation may refer to multiple historical versions. Numeric reasoning benchmarks test arithmetic or comparison in isolation; our setting requires applying a numeric comparison only after the valid legal rule has been selected.

Our contributions are:

1. We formalize law-in-force retrieval as authority-validity-support retrieval: the evidence must match the authority, be valid for the operative date, and contain answer-supporting rule text.

2. We build LawInForce-eCFR from real eCFR/CFR amendments, yielding 805 human-validated amendment pairs and 3,220 controlled answer-flipping QA instances.

3. We introduce AuthorityFirst-RAG, a transparent diagnostic retriever that separates authority identification, temporal scoping, and rule-support ranking.

4. We show empirically that retrieval improves substantially but generation remains hard, revealing a dual bottleneck in versioned legal RAG.

2 Task Definition

2.1 Problem setting

Let C be a versioned legal corpus containing passages from multiple versions of legal authorities. A passage $d \in C$ is represented as

$$d = (text, citation, section, paragraph, valid_from, valid_to).$$

The fields $citation$ and $section$ identify the legal authority. The interval $[valid_from, valid_to)$ represents the period during which the passage was in force, with an open-ended $valid_to$ indicating continued validity.

A query contains a legal question q and an operative date t. The operative date is the date at which the facts are evaluated. In a regulatory filing question, for example, it may be the filing date, deadline date, conduct date, or effective-date-relevant factual date specified by the prompt. A passage is temporally valid for the query if

$$valid_from(d) \leq t < valid_to(d).$$

This condition is not a convenience filter. It determines whether the passage can serve as valid legal authority.

2.2 Legally applicable evidence

We define a passage as *legally applicable evidence* for a question when it satisfies three conditions:

1. **Authority match:** it belongs to the relevant legal authority, such as the correct citation, section, or regulatory provision.

2. **Temporal validity:** the operative date falls within the passage validity interval.

3. **Rule support:** the text contains the threshold, comparator, unit, exception, or local rule language needed to answer the question.

These conditions are jointly necessary. Authority match without temporal validity can retrieve the wrong historical rule. Temporal validity without authority match can retrieve a contemporaneous but irrelevant provision. Authority and validity without rule support can retrieve a nearby paragraph that does not answer the question.

We focus on yes/no legal questions. Each question has a gold passage and a gold answer. Retrieval evaluation asks whether the correct in-force passage is found. Generation evaluation asks whether

an LLM produces the correct yes/no answer from supplied evidence. Separating the two lets us distinguish wrong-law retrieval from rule-application failures.

2.3 Why citation match is insufficient

Citation is an important legal signal, but it is not equivalent to law in force. The same citation can refer to multiple historical versions of the same section, and legal corpora often contain both obsolete and current text. If a question asks about conduct in an earlier period, the current version may be the easiest passage to retrieve while being legally inapplicable. Conversely, a superseded version may govern an older fact pattern even though it is not the current rule. The user's factual date determines which version is authoritative.

2.4 Scope

LawInForce-eCFR focuses on a foundational case: real legal amendments that change numeric thresholds and induce answer flips. Examples include deadlines, monetary caps, percentages, employee counts, penalties, ages, and frequencies. More complex issues such as retroactivity, transition clauses, tolling, and multi-rule statutory interpretation are left outside the benchmark. This restricted scope is intentional because it yields a controlled setting where wrong-version retrieval and numeric rule application can be diagnosed cleanly.

3 The LawInForce-eCFR Benchmark

LawInForce-eCFR is built from versioned U.S. federal regulations in eCFR/CFR.[1] It is not intended to cover all jurisdictions or legal sources; it targets the retrieval problem created when a corpus stores multiple versions of regulatory text. Table 1 summarizes the benchmark.

3.1 Source corpus and amendment mining

The benchmark begins from legal changes rather than user questions. We mine amendment pairs in which two versions of the same eCFR/CFR authority contain a local numeric change. Each pair links an old version and a new version of the same rule, together with their validity intervals and authority metadata. Starting from amendments gives the benchmark a clear target: the old and new passages are both plausible retrieval results, but only one is in force for a given operative date.

This construction differs from collecting natural legal questions and then searching for answers. Natural questions often mix ambiguity, missing facts, broad legal interpretation, and retrieval failure. By starting from verified amendments, LawInForce-eCFR isolates a narrower but important failure mode: the retriever can find the right legal neighborhood while selecting the wrong version.

Statistic	Value
Legal collection	eCFR/CFR
Jurisdiction	U.S. federal regulations
Human-validated amendment pairs	805
QA instances	3,220
QA per pair	4
Acceptance rate after filtering	94%

Table 1: Compact statistics for LawInForce-eCFR. The benchmark contains 805 human-validated eCFR/CFR amendment pairs and 3,220 controlled QA instances.

3.2 Numeric threshold alignment

For each candidate pair, we align old and new rule sentences, extract the changed numeric values and units, and infer the comparator relation expressed by the rule. The comparator may be an upper bound, lower bound, deadline, range, cap, minimum, maximum, or frequency constraint. Deterministic checks require that the relevant values appear in the legal text, that the units are compatible, and that the old and new sentences express the same underlying rule apart from the changed threshold.

Numeric threshold changes are useful for diagnosis because they allow answer flips to be constructed without inventing a new legal issue. If the old threshold is 60 days and the new threshold is 90 days, a fact value of 75 days lies between the thresholds. The same fact pattern therefore receives opposite yes/no answers depending on which version is legally applicable. The benchmark uses these controlled facts to make wrong-version retrieval observable.

3.3 Controlled answer-flipping QA generation

Each validated amendment pair yields four QA instances by crossing the operative version with whether the authority is explicitly cited. The old-date questions use an operative date during the

[1] Official sources: eCFR API documentation and GovInfo bulk data (Office of the Federal Register, n.d.; U.S. Government Publishing Office, n.d.).

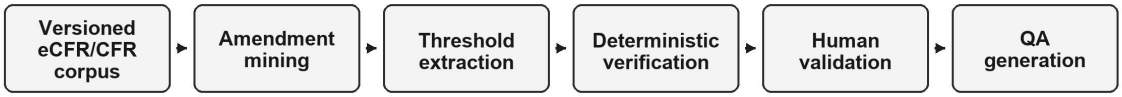


Figure 1: LawInForce-eCFR is constructed from versioned eCFR/CFR text by mining real amendments, extracting threshold changes, verifying deterministic answer flips, validating pairs by humans, and generating controlled QA instances.

QA type	Operative date	Citation given
Old cited	old-date	Yes
New cited	new-date	Yes
Old uncited	old-date	No
New uncited	new-date	No

Table 2: Each amendment pair yields four QA instances, crossing the applicable version date with whether the question explicitly cites the legal authority.

old version's validity interval; the new-date questions use an operative date during the new version's validity interval. The fact value is held constant across the old and new variants so that the legal answer flips because the governing rule changes, not because the facts change.

Cited variants include an authority cue such as a citation or section reference. They test whether a system can select the correct version when the legal neighborhood is mostly known. Uncited variants omit that cue and require the retriever to infer the authority from the factual description and legal context. Together, the two forms distinguish authority identification from version selection within a known authority.

3.4 Human validation

Human validation removes cases where deterministic matching is not enough. Our checklist verifies rule equivalence between old and new text, a single numeric threshold change, unit compatibility, comparator clarity, and deterministic answer flips, and rejects transition clauses, retroactivity, tolling, exceptions, and non-deterministic drafting changes. The 94% acceptance rate reflects that rule-based filters remove most unsuitable candidates before human review. Appendix A gives the checklist and field schema.

All 805 amendment pairs were independently reviewed by two annotators using the validation checklist. The annotators achieved 95.28% agreement; disagreements were discussed, with unresolved cases adjudicated by a third experienced annotator. The same dual-annotation and disagreement-resolution procedure was used for error-taxonomy annotation.

3.5 Benchmark purpose

LawInForce-eCFR is a diagnostic benchmark, not a comprehensive model of legal advice. Its purpose is to stress a failure mode that ordinary relevance evaluation can hide. A system receives credit only when it retrieves evidence that is authoritative, temporally valid, and answer-supporting. This makes the benchmark useful for measuring whether retrieval systems can distinguish legally applicable text from plausible but invalid nearby versions.

4 AuthorityFirst-RAG: A Diagnostic Retriever

4.1 Method overview

AuthorityFirst-RAG is a transparent authority-first retrieval procedure. We do not present it as a universal neural retriever or as a deployable legal search engine. Instead, it is designed to test which legal-structural signals matter for law-in-force retrieval. The pipeline first extracts query structure, then identifies candidate authorities, applies temporal scoping, retrieves within the candidate authority scope, and finally ranks passages by rule-support signals.

AuthorityFirst-RAG is implemented with deterministic rule-based parsing and heuristic feature-based ranking rather than a learned retriever. Citation cues, operative dates, fact values, units, and comparator signals are extracted using regular expressions and rule-based normalization. After temporal filtering, ranking combines authority/citation match, BM25 relevance, lexical overlap, threshold/unit compatibility, and comparator cues; no learned parameters are used.

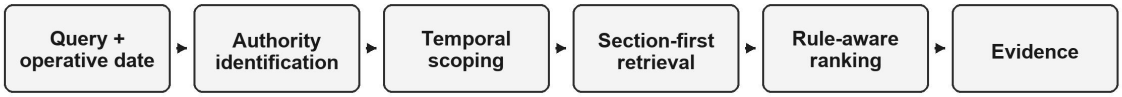


Figure 2: AuthorityFirst-RAG first narrows candidates by legal authority and temporal validity, then ranks paragraphs by rule-support signals such as threshold, unit, and comparator match.

4.2 Query parsing

The query parser extracts four types of information. First, it identifies citation cues when the question explicitly mentions a title, part, section, or other authority marker. Second, it extracts the operative date that determines which version is in force. Third, it extracts the fact value and unit, such as a number of days, dollars, employees, percent, or years. Fourth, it detects whether the question is cited or uncited. This distinction affects how much weight the retriever gives to exact citation matching versus broader lexical and legal-context cues.

4.3 Authority identification

When citation cues are present, AuthorityFirst-RAG prioritizes exact or normalized matches to the cited authority and section. When the question is uncited, it uses lexical overlap and legal-context cues to identify candidate sections. This stage is deliberately section-first: it tries to locate the legal authority before ranking individual paragraphs. The design reflects the structure of legal retrieval, where the correct paragraph is often meaningful only within a particular authority.

4.4 Temporal scoping and section-first retrieval

After candidate authorities are identified, AuthorityFirst-RAG filters or prioritizes passages whose validity intervals cover the operative date. Temporal scoping is applied before fine-grained rule ranking so that the system does not treat obsolete and current versions as interchangeable. Section-first retrieval then ranks paragraphs within the authority scope rather than across the entire corpus. This reduces the chance that a globally similar passage from the wrong version displaces the legally applicable passage.

Stage	Operation
1	Parse citation cues, operative date, fact value/unit, and cited/uncited status.
2	Identify candidate authorities, using exact citation matches or legal-context cues.
3	Scope candidates to validity intervals covering the operative date.
4	Retrieve within candidate sections before paragraph ranking.
5	Rank by authority, date validity, lexical overlap, threshold/unit match, and comparator compatibility.
6	Return the highest-scoring legally applicable evidence.

Table 3: Compact retrieval procedure for AuthorityFirst-RAG. The method is a transparent diagnostic decomposition of law-in-force retrieval.

4.5 Rule-aware ranking

The final ranking stage combines lexical relevance with rule-support features. These features include threshold match, unit compatibility, comparator compatibility, citation or section match, and date validity. Comparator compatibility is crucial because expressions such as *within*, *not more than*, *at least*, *shall not exceed*, and *between* encode different comparison directions. A passage that contains the right number but reverses the legal comparison does not support the answer.

Table 3 summarizes the procedure. Its key assumption is that legal applicability is structured: a retriever should first ask whether a passage comes from the right authority and was in force, then ask whether the local rule text supports the answer. This decomposition lets the experiments separate failures caused by authority identification, temporal validity, and rule support.

5 Experimental Setup

5.1 Retrieval evaluation

Retrieval uses Recall@1, Recall@3, Recall@5, Recall@10, and MRR. Table 4 reports the 644-instance test split aligned with generation and er-

ror analysis; Appendix C reports all 3,220 QA instances. Recall@1 is especially important for RAG (Lewis et al., 2020) because downstream answer generation often relies heavily on the top evidence.

We compare lexical, citation-assisted, ColBERT-style, and authority-first retrieval, alongside an oracle evidence condition. BM25-only uses Okapi BM25 scoring (Robertson and Zaragoza, 2009) as version-blind lexical retrieval. BM25-date-filter is BM25 retrieval with temporal filtering. citation-hybrid date filter is a metadata-assisted diagnostic baseline: it uses citation/authority cues and date filtering, with gold title/section fallback metadata for uncited queries to test version selection under partially oracle authority information. The ColBERT-style late-interaction baseline ranks passages without validity filtering in its version-blind condition; the post-date-filter condition removes temporally invalid passages after retrieval, retaining the remaining ranking order. GoldOracle is an oracle evidence condition with 100% retrieval and is used as an upper bound for generation, not as a deployable retriever.

5.2 Generation evaluation

Generation is evaluated by feeding a question and retrieved evidence to open-weight LLMs and parsing a yes/no answer. The prompt supplies the question and evidence; the model must decide whether the fact pattern satisfies the applicable rule. This setup reflects a common legal RAG pipeline while keeping retrieval and answer generation analytically separable.

Generation models receive only the top-1 retrieved paragraph-level passage rather than the full CFR section. Citation and heading information are embedded in the retrieved text; validity intervals are used for retrieval filtering but are not separately provided to the generator.

Generation evaluation is reported for completed model-condition runs. The reported evaluations contain 1,200 examples for Qwen2.5-32B-Instruct (Yang et al., 2025b); 96 examples each for Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), Qwen3-8B, and Qwen3-14B (Yang et al., 2025a); and 2,576 examples each for Qwen2.5-14B-Instruct, Qwen2.5-7B-Instruct (Yang et al., 2025b), Zephyr-7B-beta (Tunstall et al., 2024), Mistral-Nemo-Instruct-2407 (Mistral AI, 2024), and DeepSeek-R1-Distill-Qwen-7B/14B (Guo et al., 2025). Because sample sizes differ, cross-model comparisons should be interpreted cautiously. We emphasize evidence-condition and failure-mode comparisons rather than model ranking.

6 Results

6.1 Authority-first ranking improves top-rank legal applicability

Table 4 shows that authority-first ranking substantially improves top-rank legal applicability. AuthorityFirst-RAG reaches 30.90% Recall@1, compared with 2.80% for BM25-date-filter, 9.16% for citation-hybrid date filter, and 0.31% for BM25-only. The improvement reflects modeling legal authority and temporal validity before ranking local rule text. On the full 3,220-instance benchmark, AuthorityFirst-RAG preserves the strongest overall ranking pattern but with lower absolute Recall@1, indicating that the full benchmark is harder than the 644-instance test split.

The contrast with citation-hybrid date filter is informative: it reaches 77.17% Recall@10 but much lower Recall@1, so citation/date signals often find the right candidate without ranking it first. In RAG, wrong top evidence can dominate the answer; law-in-force retrieval therefore requires top-rank ranking of legally applicable authority.

ColBERT-style retrieval remains weak even with post-retrieval date filtering, suggesting that semantic matching alone does not resolve authority/version selection.

On a balanced 100-instance subset, a tool-using LLM agent reaches 15.0% R@1 versus 23.0% for AuthorityFirst-RAG; both obtain 0% R@1 on uncited queries. Agent traces indicate that candidate generation/authority discovery is the main bottleneck (Appendix D).

6.2 Gold evidence is not enough for legal numeric QA

Generation gains are much smaller than retrieval gains. Table 5 reports sample-weighted averages across completed condition-level runs and shows that AuthorityFirst-RAG improves average generation accuracy over BM25-date-filter, but by far less than the Recall@1 gain in Table 4. Correct evidence is necessary because an invalid version may support the opposite answer. It is not sufficient because the model must still interpret the operative date, threshold, unit, and comparator.

The GoldOracle evidence condition confirms the same pattern. In that setting, the best reported open-

System	Recall@1	Recall@3	Recall@5	Recall@10	MRR
AuthorityFirst-RAG	0.3090	0.4363	0.4783	0.4845	0.3772
citation-hybrid date filter	0.0916	0.3602	0.5590	0.7717	0.2779
BM25-date-filter	0.0280	0.0807	0.1273	0.2096	0.0709
BM25-only	0.0031	0.0078	0.0140	0.0264	0.0081
ColBERT-version-blind	0.0016	0.0047	0.0062	0.0078	0.0035
ColBERT-post-date-filter	0.0155	0.0404	0.0621	0.1040	0.0349
GoldOracle	1.0000	1.0000	1.0000	1.0000	1.0000

Table 4: Main retrieval on the 644-instance test split aligned with generation/error analysis. AuthorityFirst-RAG gives the strongest top-rank law-in-force retrieval, improving Recall@1 from 2.80% for BM25-date-filter to 30.90%.

Evidence condition	Accuracy (%)
GoldOracle	53.8
AuthorityFirst-RAG	52.1
BM25-date-filter	50.5
citation-hybrid date filter	48.6

Table 5: Sample-weighted average generation accuracy by evidence condition across completed condition-level runs. Better evidence helps modestly, but even GoldOracle evidence does not solve legal numeric rule application.

weight model is Qwen2.5-32B at 63.3% accuracy ($N = 300$). Among 644-example GoldOracle runs, Qwen2.5-14B reaches 60.4%, while the remaining reported models range from 49.2% to 53.3%. For Qwen2.5-32B-Instruct, BM25-date-filter yields 50.00%, citation-hybrid date filter yields 50.33%, AuthorityFirst-RAG yields 56.00%, and GoldOracle reaches 63.33%. The gap between AuthorityFirst-RAG and GoldOracle indicates remaining retrieval error; the gap between GoldOracle and perfect accuracy indicates model-side rule-application error.

These results motivate the benchmark's diagnostic separation. A system can improve retrieval without solving generation, and a model can receive the correct law while still misapplying it. Treating the task as a single end-to-end accuracy number would hide which bottleneck is responsible.

6.3 Better evidence changes the structure of generation errors

We classify erroneous generations using checklist-based manual review with heuristic consistency checks; the plotting script tabulates category shares. **Rule application errors** occur when the evidence is correct or near-correct but the model misapplies a comparator, deadline, interval, or numeric condition, or contradicts its own reasoning. **Semantic grounding errors** occur when the model fails to use the applicable legal rule, often because the evidence is mismatched or the model latches onto non-applicable text. **Output stability errors** occur when the model fails to produce a parseable yes/no answer. Appendix B gives fine-grained definitions and counts. Appendix B also reports model-wise error composition, showing that the shift from semantic-grounding to rule-application errors is not driven by a single generator.

Figure 3 shows the dual bottleneck at the error level. Under GoldOracle and AuthorityFirst-RAG evidence, most errors are rule-application errors: 91.1% and 83.0%, respectively. Under BM25-date-filter, semantic-grounding errors become dominant at 60.3%. Under citation-hybrid date filter, semantic-grounding errors are 40.9% and output-stability errors rise to 9.9%. Better evidence therefore changes the structure of failure rather than eliminating failure.

The shift is important. Weak evidence pushes models toward grounding mistakes: they use the wrong version, copy non-applicable text, or miss the controlling rule. Strong evidence leaves failures concentrated in legal rule application, so retrieval quality changes the likely error type.

6.4 Upper-bound comparator failures persist under gold evidence

A prominent source of rule-application error is upper-bound comparator failure. Legal expressions such as *not more than*, *shall not exceed*, *maximum penalty*, *capped at*, and *within X days* define upper-bound constraints. The fact must be less than or equal to the threshold, or must occur before the deadline. Models often reverse the direction and treat the threshold as a lower bound. This can turn a compliant fact into a violation, or an untimely filing into a timely one.

Figure 4 shows that this failure persists under GoldOracle evidence. The model is not simply missing the relevant passage; it is misreading the legal comparison after the correct passage is supplied.

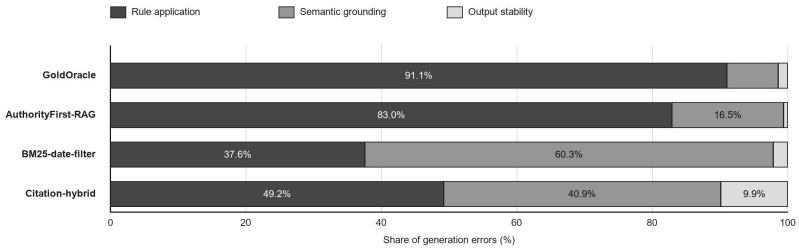


Figure 3: Generation error taxonomy across evidence conditions. High-quality evidence shifts remaining failures toward rule application, while weaker evidence increases semantic-grounding and output-stability failures.

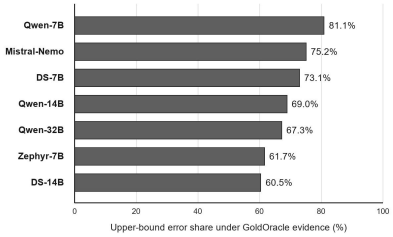


Figure 4: Upper-bound comparator failures persist even under GoldOracle evidence, indicating that some errors come from applying comparator semantics rather than from retrieving the wrong passage.

This pattern is consistent with prior numerical and legal mathematical reasoning benchmarks showing difficulty with quantities, comparison, and rule application (Dua et al., 2019; Chen et al., 2021b; Mishra et al., 2022; Zhang et al., 2025). Versioned legal RAG must solve both the retrieval problem and the direction-sensitive rule-application problem.

7 Discussion

7.1 Law in force is different from semantic relevance

A relevant passage is not necessarily legally applicable. Semantic and citation-aware retrieval reward topical and authority-like signals; law-in-force retrieval adds validity and rule support. Wrong versions are hard because they often share section numbers, headings, definitions, and most wording; in LawInForce-eCFR, the decisive difference may be a single threshold or deadline.

7.2 Wrong versions have near-zero utility

Wrong versions can be more harmful than irrelevant text because they appear authoritative and answer-bearing. This is why Recall@1 matters: if the first passage is the wrong version, the system may rely on authority that cannot govern the facts.

7.3 The dual bottleneck

AuthorityFirst-RAG improves top-rank retrieval, but downstream answer gains remain modest. Versioned legal RAG therefore has a dual bottleneck: selecting the law in force and applying its numeric rule. Better evidence shifts failures from semantic grounding toward rule application, showing that retrieval and rule application are distinct failure modes. Controlled numeric amendments make this split diagnosable because old/new versions, fact values, answer flips, and gold evidence are verifiable.

8 Related Work

Legal NLP benchmarks. LexGLUE (Chalkidis et al., 2022), LEGAL-BERT (Chalkidis et al., 2020), ContractNLI (Koreeda and Manning, 2021), and LegalBench (Guha et al., 2023) evaluate broad legal-language tasks; they do not test whether retrieved law was in force for an operative date.

Legal retrieval and legal RAG. BM25 is a standard lexical baseline (Robertson and Zaragoza, 2009), and RAG combines retrieval with generation (Lewis et al., 2020). LeCaRD (Ma et al., 2021), CLERC (Hou et al., 2025), and Hypa-RAG (Kalra et al., 2024) study legal relevance and evidence retrieval, while legal IR work emphasizes structured relevance (van Opijnen and Santos, 2017). They do not directly test operative-date validity when multiple versions share the same authority.

Temporal QA and temporal retrieval. TimeQA (Chen et al., 2021a), CronQuestions (Saxena et al., 2021), and temporal information retrieval surveys (Kanhabua and Anand, 2016) motivate time-aware

retrieval. Here, the date selects authoritative legal text, not merely a time-indexed fact.

Legal updating and versioned law. LeKUBE (Wang et al., 2024), LexTime (Barale et al., 2025), and ChronosLex (T.Y.S.S et al., 2024) address legal updating and temporal generalization. LawInForce-eCFR instead tests whether retrieval selects the version legally in force for a fact pattern's operative date.

Numerical, rule, and model reasoning. DROP (Dua et al., 2019), FinQA (Chen et al., 2021b), and NumGLUE (Mishra et al., 2022) show that models struggle with quantities and comparison; LexNum (Zhang et al., 2025) is complementary. LawInForce-eCFR couples comparison with retrieval of the law in force. The evaluated generators are Qwen, Mistral, Zephyr, and DeepSeek-R1 releases (Yang et al., 2025b,a; Jiang et al., 2023; Mistral AI, 2024; Tunstall et al., 2024; Guo et al., 2025); model ranking is not our main claim.

9 Conclusion

We introduced LawInForce-eCFR, a diagnostic benchmark for law-in-force retrieval in versioned legal RAG, together with AuthorityFirst-RAG, a transparent retriever for authority, temporal validity, and rule support. The central lesson is that topically relevant or citation-matched law can still be legally wrong when it is not in force.

The results show a dual bottleneck: retrieve applicable authority, then apply its numeric rule. Wrong versions can support the opposite answer, while correct evidence can still fail when models misapply thresholds, deadlines, or comparator direction. Versioned legal RAG must therefore be both applicability-aware and rule-faithful.

10 Limitations

LawInForce-eCFR covers U.S. federal eCFR/CFR regulations and focuses on clean numeric threshold amendments. It does not cover other jurisdictions, statutes, case law, contracts, agency guidance, or legal sources with different versioning practices. The benchmark also excludes transition clauses, retroactivity, tolling, cross-references, multi-condition rules, exceptions, and difficult legal interpretation.

Numeric thresholds, deadlines, caps, percentages, counts, penalties, ages, and frequencies are useful for controlled diagnosis because they support deterministic answer flips across versions. Many legal questions, however, require broader doctrinal reasoning, distributed definitions, factual uncertainty, agency discretion, and multi-step statutory interpretation. The results should therefore be interpreted as evidence about a controlled law-in-force retrieval failure mode, not as a complete evaluation of legal reasoning or legal advice.

AuthorityFirst-RAG is a transparent diagnostic retriever, not a deployable legal search engine. It assumes structured citation, section, paragraph, and validity-interval metadata, and metadata-assisted baselines should be interpreted diagnostically. Deployed systems may require more robust citation normalization, version alignment, and authority resolution; uncited authority inference remains weak.

Although the corpus preserves title, part, section, and paragraph_path metadata, AuthorityFirst-RAG does not explicitly navigate subsection/clause/subclause hierarchies (Blair-Stanek et al., 2023). Authority inference remains a major limitation: the uncited setting is largely unsolved, so the strongest results primarily reflect version selection when the relevant authority is known or partially specified.

Generation sample sizes differ across models and conditions, so we emphasize evidence-condition and failure-mode comparisons rather than model rankings. Future work should cover broader legal sources and more complex temporal legal phenomena.

Acknowledgments

Granted by the Tsinghua University Disruptive Innovation Talent Development Program.

References

Claire Barale, Leslie Barrett, Vikram Sunil Bajaj, and Michael Rovatsos. 2025. LexTime: A benchmark for temporal ordering of legal events. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5220–5236, Suzhou, China. Association for Computational Linguistics.

Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. 2023. Can GPT-3 perform statutory reasoning? In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 22–31. Association for Computing Machinery.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of

law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland. Association for Computational Linguistics.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021a. A dataset for answering time-sensitive questions. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021b. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, and others. 2023. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 36.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and others. 2025. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638.

Abe Bohan Hou, Orion Weller, Guanghui Qin, Eugene Yang, Dawn Lawrie, Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. 2025. CLERC: A dataset for U.S. legal case retrieval and retrieval-augmented analysis generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7913–7928.

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

Rishi Kalra, Zekun Wu, Ayesha Gulley, Airlie Hilliard, Xin Guan, Adriano Koshiyama, and Philip Colin Treleaven. 2024. HyPA-RAG: A hybrid parameter adaptive retrieval-augmented generation system for AI legal and policy applications. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pages 237–256, Miami, Florida, USA. Association for Computational Linguistics.

Nattiya Kanhabua and Avishek Anand. 2016. Temporal information retrieval. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1235–1238.

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

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.

Yixiao Ma, Yunqiu Shao, Yueyue Wu, Yiqun Liu, Ruizhe Zhang, Min Zhang, and Shaoping Ma. 2021. LeCaRD: A legal case retrieval dataset for chinese law system. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2342–2348.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523, Dublin, Ireland. Association for Computational Linguistics.

Mistral AI. 2024. Mistral-Nemo-Instruct-2407. Hugging Face model card.

Office of the Federal Register. n.d. eCFR API Documentation. Official documentation.

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

Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. Question answering over temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6663–6676, Online. Association for Computational Linguistics.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2024. Zephyr: Direct distillation of LM alignment. In *First Conference on Language Modeling*.

Santosh T.Y.S.S, Tuan-Quang Vuong, and Matthias Grabmair. 2024. ChronosLex: Time-aware incremental training for temporal generalization of legal classification tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3022–3039, Bangkok, Thailand. Association for Computational Linguistics.

U.S. Government Publishing Office. n.d. GovInfo Bulk Data Repository. Official bulk data repository.

Marc van Opijnen and Cristiana Santos. 2017. On the concept of relevance in legal information retrieval. *Artificial Intelligence and Law*, 25:65–87.

Changyue Wang, Weihang Su, Yiran Hu, Qingyao Ai, Yueyue Wu, Cheng Luo, Yiqun Liu, Min Zhang, and Shaoping Ma. 2024. LeKUBE: A knowledge update BEnchmark for legal domain. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 175–185. Association for Computing Machinery.

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

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and others. 2025b. Qwen2.5 technical report. ArXiv preprint arXiv:2412.15115.

Kepu Zhang, Guofu Xie, Weijie Yu, Mingyue Xu, Xu Tang, Yaxin Li, and Jun Xu. 2025. Legal mathematical reasoning with LLMs: Procedural alignment through two-stage reinforcement learning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1586–1598, Suzhou, China. Association for Computational Linguistics.

A Dataset Schema and Validation Checklist

This appendix summarizes the released QA records and a pair-level summary derived from them. Table 6 lists their principal fields. The 805-row `amendment_pairs.csv` preserves the old/new rule sentences, passage identifiers, and version dates available in the QA release; it does not reconstruct full validity intervals, numeric threshold fields, or per-pair validation records absent from that release.

Table 7 gives abridged QA examples following the dataset format. They illustrate the controlled answer-flip structure and omit raw source text and row-level metadata.

Table 8 summarizes the validation checklist used to retain deterministic amendment pairs and reject cases that require broader legal interpretation.

B Fine-Grained Error Taxonomy

Table 9 maps fine-grained error types to the three high-level groups used in Section 6.3. Full fine-grained row-level error files are provided as supplementary material.

Table 10 gives the per-condition counts underlying the error analysis.

Because generation runs cover multiple open-weight LLMs, we also report model-wise high-level error composition. The pattern is stable: GoldOracle and AuthorityFirst-RAG failures are dominated by rule application, while BM25-date-filter failures are dominated by semantic grounding. The citation-hybrid condition is more mixed, with one model showing a large output-stability component.

Record type	Principal fields
Amendment-pair summary	`pair_id`; split; source and amendment type; title, section, and citation; old/new gold passage identifiers, paragraph paths, and version dates; old/new rule sentences; fact value; unit; comparator type.
QA instance	`question_id`; `pair_id`; split; `question_type` (cited/uncited); question and facts; operative date; fact value and unit; comparator type; answer; gold passage identifier and authority metadata; old/new rule sentences.

Table 6: Compact field schema for the released pair-level summaries and QA records. The old/new QA variant is encoded in `question_id`.

Variant	Abridged question pattern	Governing rule	Fact	Answer
Cited old-date	Under the cited provision, is an application filed 75 days after the triggering event timely?	60-day limit	75 days	No
Uncited new-date	Without an explicit citation cue, is the same 75-day filing timely under the operative rule?	90-day limit	75 days	Yes

Table 7: Abridged QA examples showing how old and new versions can yield opposite answers for the same fact value.

Criterion	Accept when	Reject when
Rule equivalence	Old and new text express the same underlying rule apart from the numeric change.	The amendment changes scope, actor, condition, exception, or legal issue.
Single numeric threshold change	One local threshold drives the answer flip.	Multiple interacting values or drafting changes determine the answer.
Unit compatibility	Old and new thresholds use compatible units such as days, dollars, percent, counts, or years.	Units are ambiguous, incomparable, or require external conversion.
Comparator clarity	Direction is clear: upper bound, lower bound, deadline, range, cap, minimum, or maximum.	Comparator direction is implicit, conflicting, or interpretation-dependent.
Deterministic answer flip	A fact value between thresholds yields opposite yes/no answers across versions.	The answer depends on missing facts, discretion, exceptions, or unresolved interpretation.
Rejection conditions	None of the listed complications governs the answer.	Transition clauses, retroactivity, tolling, exceptions, cross-references, or non-deterministic drafting changes are present.

Table 8: Checklist for human validation of amendment pairs.

Fine-grained type	High-level group	Definition
`upper_bound_comparator_error`	Rule application	The model treats an upper bound, cap, maximum, or deadline as if exceeding it were allowed or required.
`lower_bound_comparator_error`	Rule application	The model reverses or misapplies a minimum, at-least, or lower-bound condition.
`temporal_deadline_comparison_error`	Rule application	The model compares dates, elapsed days, filing windows, or deadlines in the wrong direction.
`mixed_min_max_comparator_error`	Rule application	The model confuses interacting lower- and upper-bound constraints.
`answer_reason_contradiction`	Rule application	The final yes/no answer contradicts the model's stated rule application or intermediate reasoning.
`complex_or_unclear_legal_semantics_error`	Semantic grounding	The model grounds on non-applicable, mismatched, or semantically unresolved legal text.
`parse_failure_or_empty_answer`	Output stability	The output is empty, unparseable, or lacks a stable yes/no answer.

Table 9: Mapping from fine-grained generation error types to high-level taxonomy groups.

Error type	GoldOracle	AuthorityFirst-RAG	BM25-date-filter	citation-hybrid date filter
Total errors	1925	1995	2062	2141
Upper-bound comparator	1347	1214	306	516
Answer/reason contradiction	182	107	2	7
Temporal deadline comparison	147	182	209	329
Complex or unclear legal semantics	146	329	1244	876
Mixed min/max comparator	40	119	139	195
Lower-bound comparator	37	33	119	7
Parse failure or empty answer	26	11	43	211

Table 10: Fine-grained error counts by evidence condition.

Evidence	Model	Error N	Rule app. (%)	Semantic ground. (%)	Output stab. (%)	Dominant fine error
GoldOracle	Qwen2.5-32B	110	90.0	10.0	0.0	upper-bound
GoldOracle	Qwen2.5-14B	255	88.6	11.4	0.0	upper-bound
GoldOracle	Qwen2.5-7B	301	92.7	7.3	0.0	upper-bound
GoldOracle	Mistral-Nemo	315	88.6	7.9	3.5	upper-bound
GoldOracle	DS-R1-14B	314	92.0	7.0	1.0	upper-bound
GoldOracle	DS-R1-7B	327	89.9	7.0	3.1	upper-bound
GoldOracle	Zephyr-7B	303	94.7	4.6	0.7	upper-bound
AF-RAG	Qwen2.5-32B	132	84.1	15.9	0.0	upper-bound
AF-RAG	Qwen2.5-14B	304	79.3	20.7	0.0	upper-bound
AF-RAG	Qwen2.5-7B	305	90.2	9.8	0.0	upper-bound
AF-RAG	Mistral-Nemo	311	75.9	22.2	1.9	upper-bound
AF-RAG	DS-R1-14B	311	86.5	13.5	0.0	upper-bound
AF-RAG	DS-R1-7B	323	78.6	19.8	1.5	upper-bound
AF-RAG	Zephyr-7B	309	87.1	12.9	0.0	upper-bound
BM25-date	Qwen2.5-32B	150	36.7	63.3	0.0	semantic/complex
BM25-date	Qwen2.5-14B	323	33.4	66.6	0.0	semantic/complex
BM25-date	Qwen2.5-7B	321	37.7	62.3	0.0	semantic/complex
BM25-date	Mistral-Nemo	319	40.8	58.0	1.3	semantic/complex
BM25-date	DS-R1-14B	322	38.2	55.0	6.8	semantic/complex
BM25-date	DS-R1-7B	322	34.8	63.0	2.2	semantic/complex
BM25-date	Zephyr-7B	305	41.3	55.4	3.3	semantic/complex
cite-hybrid	Qwen2.5-32B	149	49.0	51.0	0.0	semantic/complex
cite-hybrid	Qwen2.5-14B	326	50.0	50.0	0.0	semantic/complex
cite-hybrid	Qwen2.5-7B	317	51.7	48.3	0.0	semantic/complex
cite-hybrid	Mistral-Nemo	308	59.7	39.6	0.6	semantic/complex
cite-hybrid	DS-R1-14B	398	36.2	17.6	46.2	parse failure
cite-hybrid	DS-R1-7B	331	47.7	44.7	7.6	semantic/complex
cite-hybrid	Zephyr-7B	312	53.8	46.2	0.0	semantic/complex

Table 11: Model-wise high-level error composition by evidence condition. Percentages are computed over erroneous generations for each model-condition pair. Short labels: AF-RAG = AuthorityFirst-RAG; BM25-date = BM25-date-filter; cite-hybrid = citation-hybrid date filter.

C Supplementary Retrieval Results

Table 12 reports the same retrieval metrics over the full 3,220-instance benchmark. The ranking pattern is consistent with the main split: AuthorityFirst-RAG remains strongest overall, while absolute Recall@1 is lower on the full benchmark.

Table 13 reports retrieval ablations on the full benchmark. Authority identification, temporal scoping, and rule-aware ranking are important. Section-first retrieval is not uniformly beneficial in this reproduction: removing it improves Recall@1 but reduces deeper recall. The cited/uncited split also shows that uncited authority inference remains a major weakness.

D Tool-Using LLM Agent Baseline

The agent receives the same corpus schema and metadata as the retrieval baselines and can use lexical search, authority-field search, validity filtering, and passage inspection. It searches the corpus directly rather than reranking AuthorityFirst-RAG or BM25 outputs, and receives no oracle evidence or gold authority metadata from the QA records. Table 14 reports its results on the balanced subset.

For the tool-using LLM agent, the gold passage was retrieved in 25.0% of instances and entered the final candidate set in 23.0% of instances. Conditional accuracy when the gold passage was present in the final candidate set was 65.2%.

System	N	Recall@1	Recall@3	Recall@5	Recall@10	MRR
BM25-only	3220	0.0016	0.0090	0.0124	0.0283	0.0074
BM25-date-filter	3220	0.0339	0.0547	0.0848	0.1189	0.0549
citation-hybrid date filter	3220	0.0755	0.1786	0.2590	0.3950	0.1561
AuthorityFirst-RAG	3220	0.1997	0.4053	0.4481	0.4888	0.3080
GoldOracle	3220	1.0000	1.0000	1.0000	1.0000	1.0000

Table 12: Full-benchmark retrieval results over all 3,220 QA instances.

Ablation	N	Recall@1	Recall@3	Recall@5	Recall@10	MRR
AuthorityFirst-RAG full	3220	0.1997	0.4053	0.4481	0.4888	0.3080
w/o authority identification	3220	0.0913	0.1845	0.2649	0.2988	0.1545
w/o temporal scoping	3220	0.0742	0.1615	0.2031	0.2553	0.1283
w/o section-first retrieval	3220	0.2146	0.3665	0.3845	0.3929	0.2901
w/o rule-aware ranking	3220	0.0755	0.1792	0.2835	0.3901	0.1574
w/o comparator feature	3220	0.1888	0.3851	0.4416	0.4860	0.2957
w/o threshold/unit feature	3220	0.0755	0.1792	0.2835	0.3901	0.1574
cited-only subset	1610	0.3981	0.8093	0.8944	0.9739	0.6142
uncited-only subset	1610	0.0012	0.0012	0.0019	0.0037	0.0017
old-date subset	1610	0.2031	0.4087	0.4497	0.4919	0.3107
new-date subset	1610	0.1963	0.4019	0.4466	0.4857	0.3052

Table 13: Full-benchmark retrieval ablations.

System	N	R@1	Cited R@1	Uncited R@1
BM25-only	100	0.000	0.000	0.000
BM25-date-filter	100	0.010	0.020	0.000
citation-hybrid date filter	100	0.070	0.020	0.120
AuthorityFirst-RAG	100	0.230	0.460	0.000
Tool-using LLM agent	100	0.150	0.300	0.000

Table 14: Retrieval results on a balanced 100-instance subset with 50 cited and 50 uncited queries.