

AI for Statutory Simplification: A Comprehensive State Legal Corpus and Labor Benchmark

Emaan Hariri
ehariri@stanford.edu
Stanford University
Stanford, CA, USA

Daniel E. Ho
dho@law.stanford.edu
Stanford University
Stanford, CA, USA

Abstract

One of the emerging use cases of AI in law is for code simplification: streamlining, distilling, and simplifying complex statutory or regulatory language. One U.S. state has claimed to eliminate one third of its state code using AI. Yet we lack systematic evaluations of the accuracy, reliability, and risks of such approaches. We introduce LaborBench, a question-and-answer benchmark dataset designed to evaluate AI capabilities in this domain. We leverage a unique data source to create LaborBench: a dataset updated annually by teams of lawyers at the U.S. Department of Labor, who compile differences in unemployment insurance laws across 50 states for over 101 dimensions in a six-month process, culminating in a 200-page publication of tables. Inspired by our collaboration with one U.S. state to explore using large language models (LLMs) to simplify codes in this domain, where complexity is particularly acute, we transform the DOL publication into LaborBench. This provides a unique benchmark for AI capacity to conduct, distill, and extract realistic statutory and regulatory information. To assess the performance of retrieval augmented generation (RAG) approaches, we also compile StateCodes, a novel and comprehensive state statute and regulatory corpus of 8.7 GB, enabling much more systematic research into state codes. We then benchmark the performance of information retrieval and state-of-the-art large LLMs on this data and show that while these models are helpful as preliminary research for code simplification, the overall accuracy is far below the touted promises for LLMs as end-to-end pipelines for regulatory simplification.¹

CCS Concepts

• **Applied computing** → **Law**; • **Information systems** → **Evaluation of retrieval results**; **Question answering**; *Specialized information retrieval*.

Keywords

benchmark, statute, language models, retrieval, reasoning

¹StateCodes and LaborBench are available <https://huggingface.co/collections/reglab/a-reasoning-focused-legal-retrieval-benchmark-67a00c363f7e0d14619e95c5>.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICAIL '25, June 16–20, 2025, Chicago, IL

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM

<https://doi.org/10.1145/nnnnnnnn.nnnnnnn>

ACM Reference Format:

Emaan Hariri and Daniel E. Ho. 2025. AI for Statutory Simplification: A Comprehensive State Legal Corpus and Labor Benchmark. In *Proceedings of Twentieth International Conference on Artificial Intelligence and Law (ICAIL '25)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnn>

1 Introduction

While AI is increasingly integrated into legal practice—using AI to research positive law—one of the emerging, ambitious, and potentially most consequential use cases lies in legal reform itself—using AI to normatively change the law. An area of increasing political focus has been statutory and regulatory reform. Given the vast terrain of statutory and regulatory provisions, can advances in AI and large language models (LLMs), help with code and regulatory simplification, revision, and reform? Many politicians, AI vendors, and researchers have espoused this use case. After cutting some 22k royal decrees, Italy’s Minister for Institutional Reform touted “AI for regulatory simplification” to identify duplicative, conflicting, or outdated requirements that can trap citizens and “clog up the courts,” [1, 37]. The U.S. Department of Health and Human Services used AI in its “regulatory cleanup initiative” to identify outdated or erroneous provisions [45]. Relying on an AI tool marketed by Deloitte, the state of Ohio has purported to eliminate 5M words (or one third) of its state code, with Ohio Lieutenant Governor Husted noting “an unprecedented opportunity to use AI tools to eliminate regulations, to make them more easily understandable, and thus make them easier to comply with,” [26]. The benefits for governments, citizens, and civil servants could be immense. Former Deputy Chief Technology Officer Jen Pahlka, who helped found the US Digital Service, said, “[i]f you use AI for that regulatory simplification . . . you’ll be burdening your folks less, so that they can do more of the right things that you care about on behalf of the people in your state,” [16]. Most ambitiously, the Trump Administration’s regulatory reform agenda has unveiled the Elon Musk-led “Department of Government Efficiency” (DOGE) as a reorganization of the US Digital Service to streamline regulatory requirements using technology across all regulatory agencies (with reported utilization of AI to recommend revisions to regulations [10]). Within the private sector, vendors have touted tools for regulatory simplification (e.g., Deloitte’s RegExplorer, Aptus.AI) and a wide range of “RegTech” firms have integrated AI to automatically scan for diverse and changing regulatory requirements across jurisdictions and for simplifying compliance initiatives (see, e.g., [19]).

What is sorely lacking, however, are systematic evaluations of these capacities. To date, legal benchmarks have focused on discrete questions of rule retrieval, application, and reasoning. The ambition

of AI for regulatory reform, however, requires systematic benchmarking of *legal code interpretation* of statutes and regulations far beyond existing benchmarks.

We hence introduce LaborBench, a new benchmark dataset to assess the capacity of state-of-the-art models for code understanding, and StateCodes, a comprehensive dataset of nearly all underlying state statutory codes and regulations, totaling 5.1 and 3.6 GB, respectively. Our contributions are sixfold.

First, we leverage a unique source of data. Every few years, a team of a half dozen lawyers at the U.S. Department of Labor spends roughly six months combing through complex state statutes and regulations—described as “byzantine” [14] and requiring “a master’s degree in confusion”—to formally characterize differences in state unemployment insurance (UI) schemes. Their efforts result in a 200-page compilation, entitled “Comparison of State Unemployment Insurance Laws,” of tabular comparisons of key code differences along hundreds of dimensions across the states. This benchmark is a unique and highly realistic one—requiring years of effort by teams of lawyers and subject matter experts that continue annually as provisions are revised. We turn this into a high-fidelity dataset of 3,700 questions and answers in LaborBench.

Second, this setting is a highly realistic and consequential one for code simplification. It was inspired by a direct collaboration with a state to explore the use of LLMs to simplify unemployment insurance (UI). UI is one of the largest benefits programs to provide cash payments to workers who lose their job through no fault of their own. During the pandemic, roughly 46M Americans relied on UI, but the complexity of benefits determinations has led to strong calls for simplifying the UI thicket. In the book “Recoding America,” Pahlka devotes an entire chapter to the “new guy,” who despite 17 years of experience was still learning the ropes of benefits determinations [34]. The annual DOL compilation forms an important component of federal oversight into a federal-state cooperative scheme. More generally, many legal AI providers have marketed their AI capacity to conduct 50-state surveys to capture differences in laws across jurisdictions.² Thomson Reuters, for instance, markets Westlaw with the ability to “creat[e] comprehensive and up-to-date surveys of the law using generative AI.” [44] Burris [3] estimates that a thoroughly researched manual survey might take 3 months, with more complex areas of law requiring even more time and substantial teams.

Third, one of the emerging techniques in legal AI has been the use of retrieval augmented generation (RAG). Unfortunately, comprehensive, structured sources of all U.S. state statutes and regulations are not easily available to researchers, impeding systematic research in legal AI. We hence introduce StateCodes, which includes all statutory code from 50 U.S. states³, containing 488M words across 1.8M sections, and state regulations from 45 U.S. states, containing 268M words across 2.2M sections. We use this dataset to test RAG type approaches to solving the LaborBench tasks, but this dataset provides an ideal setting to develop further 50-state RAG benchmarks in other areas of law.

Fourth, we benchmark five state-of-the-art models (GPT-4o mini, Gemini 1.5 Flash, Claude 3.5 Haiku, Llama 3.1 70B Instruct, and Deepseek V3) with retrieval augmentation using a range of dense and sparse retrieval methods (E5, Gemini, Okapi BM25, OpenAI) on LaborBench. We find that while LLMs may help in preliminary research for code simplification, the overall accuracy is far below the touted promises for LLMs as end-to-end pipelines for regulatory simplification (F1 of 0.67). To provide one simple example, UI laws define employers as persons who employed individuals on 20 days across different weeks in a calendar year. But complexities in statutory language led Gemini identify the threshold for Colorado’s statute as *20 days*, while erroneously identifying the threshold of the federal model statute as *20 weeks*⁴. These mistakes are not isolated. Overall accuracy of the best performing model (Gemini) is 0.68, demonstrating that there are significant advances required for applications in code simplification.

Fifth, we show, however, that AI systems may nonetheless aid in the human process. We design a set of experiments to test for the impact of information retrieval (IR) in a RAG pipeline. This allows us to separately investigate (a) the performance on the IR component (citation verification), and (b) the performance of models without an IR component, which replicate the substantive setting of a search without and with knowing the relevant code provisions, respectively. We find that RAG approaches boost performance substantially, increasing F1 by 0.18, demonstrating the need for the StateCodes dataset and improved IR methods. Moreover, we show that citation elicitation via chain-of-thought prompting, even if imperfect (top-5 recall of 0.93), can aid in legal research process for code simplification.

Last, one of the unique aspects of LaborBench is that state statutes and regulations follow from a cooperative federal-state scheme, with a model federal statute. Longstanding questions exist around state legislative and regulatory capacity [25]. We illustrate how we can examine how AI assistance interacts with such capacity differences with the Correlates of State Policy dataset [12].

This paper proceeds as follows. Section 2 discusses related works that motivate the design of LaborBench and StateCodes. Section 3 articulates the construction of our new benchmark. Section 4 discusses our evaluation approach and experimental setup. Section 5 presents results and Section 6 concludes with brief implications.

2 Related Works

Efforts at improving legal retrieval and reasoning have spanned decades [27, 31]. We explore recent work focuses on applying newer retrieval methods and domain-specific legal QA.

2.1 Statute Retrieval. Our work contributes to important work on statutory retrieval. The Competition on Legal Information Extraction/Entailment (COLIEE) [11] has been important for focusing community efforts on legal AI and contains an information retrieval task (Task 3) and an entailment/QA task (Task 4). These tasks use the Official English Translation of the Japanese Civil Code, with

²The surveys can be either *cross-sectional*, focusing one point in time, or *longitudinal*, looking at multiple points in time [3].

³“State” in this paper will be used to refer to state-level jurisdictions, including the territories and the federal district (D.C.).

⁴When we exam the statutes, it is apparent they are similarly worded. *Compare* COLO. REV. STAT. § 8-70-113(1)(a)(II)(B) (“Employed at least one individual in employment for some portion of the day on each of twenty days during the calendar year ... each day being in a different calendar week.”) with 26 U.S.C. § 3306(a)(1)(B) (“[O]n each of some 20 days during the calendar year ... each day being in a different calendar week, employed at least one individual in employment for some portion of the day.”)

768 Articles, as their corpus. Task 3 involves retrieving the appropriate subset of articles for a query Q —appropriate meaning those articles that are dispositive of Q —and contains 996 test and 101 test queries. Task 4 involves the same queries datasets, but evaluates only the entailment task, i.e., “Yes/No” question answering, rather than retrieval itself. While insights from COLIEE have advanced the understanding of retrieval in legal reasoning, it remains a relatively small corpus sourced from one jurisdiction. LaborBench builds on these efforts by providing a benchmark for large-scale multi-jurisdictional retrieval and analysis.

Su et al. [42] developed STAtute Retrieval Dataset (STARD), a Chinese-language dataset consisting of 1,543 queries lacking precise legal terminology made by *non-professionals*. The corpus used is comprised of 55,348 statutory articles selected from the statutory and regulatory law of China. The stated objective is to retrieve the smallest set of relevant statutes necessary to answer the query, where relevancy was determined by law student annotators. The best method tested by Su et al. [42] achieves a recall@100 of 0.907. STARD provides valuable insights on non-professional queries in one jurisdiction; our work expands on this by including technical queries that cover 50 U.S. states on the same legal topic.

Other datasets provide large sets of queries but have relatively small corpora and limited target statutes. The Chinese AI Law (CAIL) dataset created by Xiao et al. [50] focuses on legal judgment prediction from 2.6 million criminal cases, filtered to include only those arising from 183 articles of the Chinese Criminal Code. Paul et al. [35] introduced the Indian Legal Statute Identification (ILSI) and the European Court of Human Rights (ECtHR-B) datasets, the latter building on Chalkidis et al. [6]’s ECHR dataset. ILSI contains 65,950 queries derived from Indian Supreme Court cases, with 100 target statutes from the Indian Penal Code, while ECtHR-B has 10,825 queries, with just 10 target statutes from the European Convention on Human Rights, all in English. While ECtHR-B focuses on distinct statutes, ILSI highlights challenges in retrieving and reasoning over similarly worded laws⁵. Our work extends this focus by providing an order of magnitude more target statutes and addressing similar legal provisions across diverse jurisdictions⁶, enabling richer evaluation of LLMs’ handling of semantic overlap and nuanced statutory language.

2.2 State Law Corpora. One of the major impediments to research on U.S. state jurisdictions has been the lack of a structured corpus on state statutes and regulations. Henderson et al. [15], for instance, include U.S. state codes in the Pile of Law, but codes are concatenated and then chunked into 150 sections evenly and Pile of Law does not contain state regulations, which often play an outsized role in legal implementation. StateCodes is the first dataset we are aware that offers U.S. state codes in a structured manner and contains the vast majority of U.S. state regulations.

2.3 Legal RAG Benchmarks. Existing benchmarks address a wide array of legal reasoning tasks, including legal judgment prediction, topic classification, interpretation, knowledge memorization,

and issue spotting [7, 9, 13]. These benchmarks, however, place less emphasis on the retrieval and processing of very long texts, which is increasingly important given the rise of legal RAG systems [29]. Guha et al. [13] introduced LegalBench, which encompasses 162 tasks comprehensively covering six types of legal reasoning but excludes tasks that focus on processing long documents. Fei et al. [9] explicitly exclude from their benchmark, LawBench, any tasks like legal case retrieval that involve reasoning over long documents. LegalBench contains only 9 sections of text (sourced from the Internal Revenue Code) exceeding 5K tokens. StateCodes provides a highly realistic benchmark, with extensive context length, with, for instance, over 100 sections exceeding 5K tokens.

Pipitone and Alami [36] developed LegalBench-RAG to focus specifically on the evaluation of RAG systems in the legal domain. LegalBench-RAG contains 6,858 queries which are divided so as to relate to 4 different corpora—NDA related documents, private contracts, public M&A documents, and privacy policies—containing 79M characters. This benchmark covers exclusively private law matters, not including case law, statutory, or regulatory retrieval tasks. With 4.8B characters, StateCodes expressly focuses on the important setting of statutory and regulatory law.

3 Datasets

We now describe the creation of StateCodes—state statutes and regulations across all domains—and LaborBench, which enables assessment of legal RAG in the domain of unemployment insurance (UI). Focusing on UI has several benefits. First, UI is notoriously complex [34], providing an ideal testing ground. Second, all state laws nonetheless must address the same minimum set of issues, due to the cooperative federal scheme. Many provisions are modeled on the Federal Unemployment Tax Act (FUTA) (as illustrated in Footnotes 4 and 6). This structure enables the tabular comparisons compiled by DOL that we transform into LaborBench. Third, UI resembles many other areas of law where states rely heavily on model statutes to promote legal consistency across jurisdictions, while retaining flexibility to tailor provisions.⁷ The value of StateCodes beyond testing just LaborBench is clear, as it enables researchers to systematically study state statutory and regulatory law across all domains.

For a simplified illustration of the steps involved in creating and benchmarking LaborBench and StateCodes, see Figure 1.

3.1 LaborBench.

3.1.1 Source Data. The data underlying LaborBench was sourced from the 2023 edition of the *Comparison of State Unemployment Insurance Laws (CSUIL)*, an annually published report which describes itself as providing “state-by-state information on workers covered, benefit eligibility, methods of financing, and other areas of interest in the Unemployment Insurance” [40].

As noted, many legal experts spend almost six months updating CSUIL annually. In developing state surveys, lawyers define legal questions; identify divergence from federal legislation; and update table accuracy based on recent legislation, regulation, and case law.

⁵Paul et al. [35] compare Indian Penal Code (India Act XLV), 1860, §§ 323, 325, 326 with Convention for the Protection of Human Rights and Fundamental Freedoms arts. 2–14, Nov. 4, 1950, 213 U.N.T.S. 222 as an example.

⁶For instance, LaborBench includes QAs about UI definitions, which are often similar. Compare, e.g., MASS. GEN. LAWS ch. 151A, § 1(j) with HAWAII REV. STAT. § 381-1c.

⁷For instance, the Uniform Commercial Code is at least partially adopted in all 50 states and D.C.U.C.C. (AM. LAW INST. & UNIF. LAW COMM’M); Within estate law, the Uniform Probate Code see wide usage. UNIF. PROB. CODE, 8 pt. 1 U.L.A. 1 (2013).

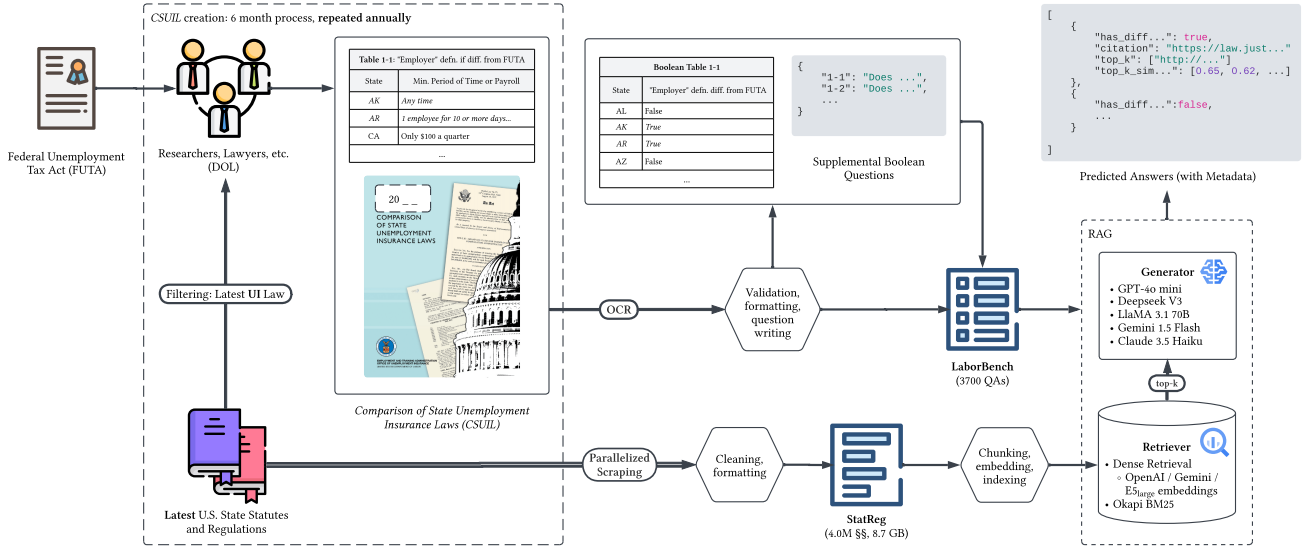


Figure 1: Summary of compilation and benchmarking processes of LaborBench and StateCodes.

The report is the compilation of 86 extensive state comparison tables spread across 8 different chapters, each representing a sub-field of UI. Each state survey is a topic-focused table where the rows represent states and the columns each represent a subtopic related to the topic. The majority of the tables have 53 rows, representing all 50 states, the District of Columbia, and the territories of Puerto Rico and the U.S. Virgin Islands (although some tables have fewer rows, with one having only 2 rows). The average table has just over 2 columns. Each column will have entries of a common data type, which may be either string, integer, or boolean; many string columns in the dataset may be cast into categorical columns (e.g., some tables in *CSUIL* chapter 5 have columns akin to [‘L’, ‘R’, ‘L’, ‘I’, ‘I’, ...]).

Some table entries contain footnotes. When storing these entries in the dataset, the footnote is removed and the footnote text is saved under a separate ‘metadata’ key for that entry. Additionally, there is a master table of abbreviations covering the table.

3.1.2 Question/Answer (QA) Construction. We employ the tables to construct a QA dataset with (1) each table’s respective context, usually the text immediately preceding each table, and (2) an auxiliary set of questions derived from the headers of the original table.

For example, *CSUIL* Table 3-16, entitled “Withholding State and Local Income Tax at Claimant Option” and having the column “State Taxes,” yields the following question:

Given the description above, can state income taxes be withheld from unemployment insurance benefits at the claimant’s option in {jurisdiction}?

For each row of the table, we replace the placeholder text ‘jurisdiction’ with the full jurisdiction (typically state) name.

Each entry of LaborBench is composed of: (1) an index (integer), (2) the column number (integer), (3) the column name (string written in snake case (e.g. ‘has_different_employer_definition’)), (4) the column data type (string, one of ‘bool’, ‘str’, or ‘int’), (5)

the jurisdiction (string), (6) the abbreviation for that jurisdiction (string), (7) question context (string), (8) a question derived from the column header and jurisdiction (string), and (9) the answer to that question (type corresponding to the column data type).

3.2 Boolean Supplement to LaborBench. We also construct a boolean supplement (with binary answers). We start with the initial set of boolean questions from LaborBench and add 1,272 questions, derived from the state survey tables. For each of 20 tables, we typically formulate a question that asks whether the state has express statutory language on the subject. For simplicity, we use one-hot encodings for categorical answers. This query can then be formulated to *all* jurisdictions. For example, *CSUIL* Table 1-1 (in Table 1(a)) lists provisions defining “employer” in states where the definition deviates from the FUTA. From this, we generate Table 1(b), a boolean table indicating for all states whether their definitions deviate from FUTA.

Table 1: Converting (a) *CSUIL* Table 1-1, Definition of Employer (If Different from FUTA 20 Weeks/\$1,500 Rule), to (b) a Boolean Table for LaborBench

| (a) | | (b) | |
|-------|--------------------|-------|-------------------------|
| State | Min. Time/Payroll* | State | FUTA Diff. [†] |
| AK | Any time | AL | |
| AR | 1 employee ... | AK | X |
| CA | Over \$100 ... | AR | X |
| ⋮ | ⋮ | AZ | |
| ⋮ | ⋮ | ⋮ | ⋮ |

* Minimum Period of Time or Payroll

[†] Defines “Employer” Differently From FUTA

3.3 StateCodes. In addition to the QAs, we provide 50 U.S. state statutory codes and 45 U.S. state regulatory codes (or administrative codes) in a structured corpus. This corpus dates to 2023, corresponding to LaborBench’s source data (*CSUILL* 2023). Sections of statute and regulation exhibit great variation in length: statutory code sections have average of 265 words ($SD = 817$), while regulatory code sections have an average of 122 words ($SD = 780$). The most verbose statutory and regulatory sections have 332K and 609K words, respectively. Different state statutory codes vary in size from having 4,700 code sections (Illinois) to nearly 160K sections (California).

We source our state statutes and regulations from Justia, a legal information provider.⁸ The U.S. regulations on Justia are themselves sourced from Fastcase [18].⁹ While state codes and regulations can be obtained digitally from each state, there is little standardization in the way the states offer their codes digitally [4, 43, 48]. Even within the same state, codes and regulations may not be standardized [4, 5].

While much legal AI has focused on case law research, state statutes and regulations are of particular interest, as they determine how the largest government programs are administered, exhibit distinct language, and have extensive scope and complexity. Table 2 provides a sense of the size of StateCodes relative to existing corpora and benchmarks.

Table 2: Relative Size of Statutory Corpora Used for IR or LLM Benchmarking

| Legal Corpus | Characters | Sections |
|-----------------------|------------|----------|
| COLIEE (Task 3) [11] | ~315K | 768 |
| CAIL [50] | 113K | 183 |
| STARD [42] | 6.6M | 55K |
| ILSI [35] | ~40K | 100 |
| U.S. Code | 49M | 54K+ |
| StateCodes (Statutes) | 3.0B | 1.8M |
| StateCodes (Regs.) | 1.8B | 2.2 M |

4 Experimental Setup

We focus our experiments on the boolean subset of LaborBench, using a range of generative and retrieval models with a filtered subset of StateCodes of UI laws.¹⁰ This allows us to retain their complexity while making our analysis more tractable, but researchers can of course make the task more difficult by omitting this filter. (As we show below, this task proves sufficiently difficult for the range of systems we consider.) Table 3 presents an example of prompts and system interaction.

4.1 RAG. Legal RAG has gained in popularity, given its effectiveness on knowledge-intensive NLP tasks like open-domain QA

[23] and ability to reduce hallucinations [41]. LaborBench and its boolean supplement provide an ideal setting for testing RAG using StateCodes. We test a number of retrieval methods, including dense passage retrieval (DPR) [20, 23] and sparse retrieval, against a baseline.

We test DPR using three popular embedding models, two proprietary one open source. The proprietary models we use are OpenAI’s ‘text-embedding-3-small’ [33] and Google’s ‘text-embedding-004’ [22]. We also use the E5_{large} model (‘e5_large_v2’) from the open source family of E5 models[46], itself derived from BERT-based MiniLM[47]. For sparse retrieval, we use the efficient and popular Okapi BM25 retrieval model [17, 30, 38, 39].

4.2 Ingestion. To ingest StateCodes, we implement a hybrid of semantic and fixed-size chunking strategies, since StateCodes (both its statutes and regulations) are organized into individual sections. Here we choose to ingest both corpora, but we may choose to ingest just one (or neither, in the case of our baseline) of them. In particular, chunking is done only *within* each statute section, i.e., a chunk will never span two sections. Within each section, fixed-size chunking is used with a chunk size of 1,000 tokens and an overlap of 200 tokens.¹¹ We use the GPT-4 tokenizer (tiktoken with encoding ‘cl100k_base’) at this stage.

Top- k retrieval for relevant chunks is done using Maximum Inner Product Search (MIPS) for DPR. Upon querying, MIPS is used to obtain the top- k relevant chunks. For context selection, we then obtain the full *section* (from StateCodes) associated with each chunk to provide more context to the model during inference.¹² For the purposes of our experiments, we use $k = 5$.

Our query is augmented in a straightforward manner. For each inference model text (see below) the sections derived from the top-5 chunks are placed in the system prompt in this fashion:

```
Given the context, generate a response to the user query:
Document 1: {title[0]}
Citation URL: {url[0]}
Context: {content[0]}
Document 2: {title[1]}
...
```

Where `title[i]` is the title of the i^{th} section of statute or regulation fetched (of the modified top- k , i.e., $0 \leq i < k$), which contains the path,¹³ `url[i]` is the Justia URL linking to that statute or regulation, and `content[i]` is the text of that section.

4.3 Inference. For generation, we tested five popular LLMs: GPT-4o mini (gpt-4o-mini-2024-07-18), Gemini 1.5 Flash, Claude 3.5 Haiku (claude-3-5-haiku-20241022), Llama 3.1 70B Instruct, and the recently released Deepseek V3 [8]. Claude 3.5 Haiku and Llama 3.1 70B Instruct were only used with the OpenAI embedding model. To optimize the extraction of boolean responses when testing LaborBench’s boolean supplement, we implemented a two-step inference strategy.

¹¹While chunking is not strictly necessary for Okapi BM25, long documents have been shown to degrade Okapi BM25’s retrieval performance [28].

¹²It is possible many of the top- k chunks retrieved are from the same section. In this case, we drop the chunks associated with lower similarities (since the resulting set of retrieved *sections* will be the same).

¹³For instance, a full title looks like “2023 Code of Alabama > Title 1 - General Provisions. > Chapter 1 - Construction of Code and Statutes. > Section 1-1-1 - Meaning of Certain Words and Terms.”

⁸To handle the structure and sheer size of the state laws, we write a script which collects them in parallel. This script can be used to collect the laws for any given year. This script can be found here: <https://github.com/reglab/lawscraper>.

⁹We note that such governmental materials, including state statutes and regulations, are not subject to copyright protection. See *generally* Georgia v. Public.Resource.Org Inc., 590 U.S. 255 (2020); Wheaton v. Peters, 33 U.S. 591 (1834).

¹⁰We determine which sections of each jurisdiction’s statutory and regulatory codes correspond to UI laws. Hand labeling is required because states publish their UI laws under different names. See, e.g., KAN. STAT. ANN. §§ 44-701 to 44-777 (“Employment Security Act”); FLA. STAT. XXXI.443 (“Reemployment Assistance”). This is done using a regular expression filter over the citation URLs.

Table 3: Annotated Sample Evaluation of LaborBench using LLM

| Role | Content | Step |
|------------------|--|--|
| System | Given the context, generate a response to the user query: <i>Document 1:</i> 2023 Alaska Statutes > ... > Art. 9. General Provisions. > Sec. 23.20.520. Definitions. <i>Cite URL 1:</i> https://.../alaska/title-23/chapter-20/article-9/section-23-20-520/ <i>Context 1:</i> In this chapter, unless the context otherwise requires: (1) "American vessel" means (2) "average annual wage" means ... (10) "employer" means Explain your reasoning, and think step by step ... | 0. Context is retrieved from StateCodes using Q . |
| User | Q: "Does Alaska define employer differently than FUTA?" | 1. Query $Q \in$ LaborBench using above system prompt. |
| Assistant | Yes, the state of Alaska defines "employer" differently from the FUTA (Federal Unemployment Tax Act) standards, particularly regarding the 20 weeks/\$1,500 rule. Under the Alaska Employment ... | 2. Request the answer conform to the JSON Schema derived from Q . |
| User | Please respond using the following JSON schema: <pre>{ "properties": { "has_different_employer_definition": { "type": "boolean", ... }, "citation_url": ... }, ... }</pre> | |
| Assistant | <pre>{ "has_different_employer_definition": true, "citation_url": ".../alaska/title-23/chapter-20/article-9/section-23-20-520/" }</pre> | (3. Result may be validated using JSON Schema.) |

First, the query from LaborBench was presented to the models, generally prompting them to provide their reasoning alongside the response. Optionally, we enhanced this step by explicitly requesting chain-of-thought (CoT) reasoning [21, 49] and, where relevant, an explicit citation to support the response.

Second, providing all information from the previous step, we then prompt the model to produce a JSON fitting a schema. To dynamically create JSON Schemas for each query, we use a Pydantic model that contains at most fields: the first field has a name matching the column name for that query (e.g., 'has_different_employer_definition', see Section 3.1.2) and a type matching the column data type (in the case of LaborBench's supplement, this will always be 'bool'); the second and optional field is named 'citation_url' and is of type string. This allows us to obtain a final output that is boolean in the vast majority of cases.¹⁴

5 Results

We now present results, which suggest that while LLMs may be useful aids, much effort is still needed to advance the performance of LLMs and retrieval models in this space.

5.1 Retriever and Generator Model Performances. The evaluation reveals notable differences in performance across various retriever and generator models, as highlighted in Table 4.

The Gemini retriever demonstrates strong performance across all metrics. It outperforms all other retrieval methods on all metrics except recall, see Table 4 (although the Gemini-GPT configuration has the highest recall among all RAG configurations). We observe that the BM25 generally lags behind the DPR methods in most metrics, albeit not substantially (and it even achieves higher precision than $E5_{large}$). The BM25-Gemini configuration performs poorly across metrics.

Overall, the highest F1 revealed among all RAG configurations was 0.691, indicating that the problem remains challenging, forming a good benchmark task for measuring advances in this space. Even

in the simplified boolean QA task with RAG, models misinterpret statutory and regulatory requirements over one third of the time.

Table 5 also provides a comparison of the average results by generator LLM (conditional on retrieval). Gemini exhibited relatively strong accuracy and recall results, but lower F1 scores. Claude and Llama had the highest and lowest F1 scores among generators, but were only tested with the OpenAI retriever; Deepseek had the highest F1 scores of those LLMs tested with all retrievers. Nevertheless, all F1 scores were within close range and leave substantial room for improvement.

Table 5: Comparison of Mean Performance Metrics Across Generator LLMs, Using Retrieval

| Generator | Accuracy | Precision | Recall | F1 |
|-----------|--------------|--------------|--------------|--------------|
| Claude | 0.692 | 0.604 | 0.798 | 0.688 |
| Deepseek | 0.649 | 0.556 | 0.855 | 0.674 |
| GPT | 0.631 | 0.540 | 0.884 | 0.670 |
| Gemini | 0.698 | 0.644 | 0.644 | 0.644 |
| Llama | 0.600 | 0.517 | 0.866 | 0.647 |

5.2 Measuring the Effects of RAG and CoT.

5.2.1 Effects of RAG. Table 6 shows that, as expected [20, 23], RAG provides a substantial performance increase across all relevant metrics. RAG is particularly effective at increasing recall, reducing the high number of false negatives with baseline LLMs. In Table 4, the baseline Gemini model appears to be an outlier, and RAG improves recall substantially.

The asymmetric impact of retrieval on precision and recall suggests that while RAG helps models identify when statutory differences exist, it does not correspondingly improve their ability to either determine correctness or significantly reduce false positives. LLMs without retrieval struggle with false negatives, and RAG effectively and unsurprisingly addresses this issue by providing a needed knowledge source. However, the relatively small improvement in precision indicates that models remain limited in assessing the validity of retrieved statutes.

¹⁴Fewer than 20 out of over 20,000 results did not produce a boolean.

Table 4: Comparison of Mean Performance Metrics Across Retriever and Generator Models (\pm SE, calculated using a nonparametric bootstrap with $n = 1000$)

| Retriever | Generator | Accuracy | Precision | Recall | F1 |
|--------------------------|-----------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Baseline (\emptyset) | Deepseek | 0.534 ± 0.012 | 0.461 ± 0.014 | 0.673 ± 0.018 | 0.547 ± 0.013 |
| | Gemini | 0.564 ± 0.012 | 0.419 ± 0.037 | 0.114 ± 0.012 | 0.179 ± 0.018 |
| | GPT | 0.511 ± 0.012 | 0.450 ± 0.014 | 0.758 ± 0.014 | 0.564 ± 0.013 |
| | Average | 0.536 ± 0.007 | 0.452 ± 0.010 | 0.515 ± 0.011 | 0.481 ± 0.009 |
| E5 _{large} | Deepseek | 0.648 ± 0.012 | 0.555 ± 0.017 | 0.860 ± 0.014 | 0.675 ± 0.014 |
| | Gemini | 0.486 ± 0.041 | 0.426 ± 0.048 | 0.767 ± 0.055 | 0.548 ± 0.048 |
| | GPT | 0.625 ± 0.014 | 0.536 ± 0.017 | 0.866 ± 0.013 | 0.662 ± 0.014 |
| | Average | 0.628 ± 0.010 | 0.538 ± 0.012 | 0.858 ± 0.010 | 0.661 ± 0.010 |
| Gemini | Deepseek | 0.674 ± 0.012 | 0.579 ± 0.017 | 0.857 ± 0.015 | 0.691 ± 0.014 |
| | Gemini | 0.716 ± 0.011 | 0.665 ± 0.021 | 0.670 ± 0.017 | 0.667 ± 0.015 |
| | GPT | 0.645 ± 0.014 | 0.550 ± 0.017 | 0.901 ± 0.013 | 0.683 ± 0.015 |
| | Average | 0.679 ± 0.008 | 0.588 ± 0.011 | 0.809 ± 0.010 | 0.681 ± 0.009 |
| Okapi BM25 | Deepseek | 0.635 ± 0.016 | 0.545 ± 0.019 | 0.847 ± 0.017 | 0.663 ± 0.017 |
| | Gemini | 0.705 ± 0.015 | 0.659 ± 0.023 | 0.632 ± 0.021 | 0.645 ± 0.019 |
| | GPT | 0.603 ± 0.015 | 0.519 ± 0.017 | 0.880 ± 0.015 | 0.653 ± 0.015 |
| | Average | 0.648 ± 0.008 | 0.561 ± 0.011 | 0.786 ± 0.010 | 0.654 ± 0.009 |
| OpenAI | Claude | 0.692 ± 0.014 | 0.604 ± 0.019 | 0.798 ± 0.018 | 0.688 ± 0.016 |
| | Deepseek | 0.638 ± 0.013 | 0.547 ± 0.017 | 0.854 ± 0.013 | 0.667 ± 0.014 |
| | Gemini | 0.699 ± 0.014 | 0.654 ± 0.022 | 0.616 ± 0.021 | 0.635 ± 0.018 |
| | GPT | 0.650 ± 0.014 | 0.555 ± 0.018 | 0.889 ± 0.014 | 0.683 ± 0.015 |
| | Llama | 0.600 ± 0.014 | 0.517 ± 0.015 | 0.866 ± 0.015 | 0.647 ± 0.014 |
| | Average | 0.656 ± 0.006 | 0.566 ± 0.008 | 0.805 ± 0.008 | 0.665 ± 0.007 |

Table 6: Performance Comparison of RAG and Baseline LLMs

| Method | Accuracy | Precision | Recall | F1 |
|----------|--------------|--------------|--------------|--------------|
| Baseline | 0.536 | 0.452 | 0.515 | 0.481 |
| RAG | 0.656 | 0.567 | 0.806 | 0.665 |

5.2.2 Effects of Citation Elicitation Using Chain of Thought. We use chain-of-thought (CoT) to invoke reasoning in the model [23] while simultaneously producing a citation, selected from the retrieved (top-5) sources. Specifically, we apply CoT while requesting the generator explicitly provide a “relevant” citation for its answer (while allowing it to provide no answer if all citations appear irrelevant). Relative to RAG, citation elicitation with CoT appears to have little effect on performance. While performance improves to a degree as shown in Table 7, most increases are minimal.

Table 7: Performance Impacts of Citation Elicitation (CE) with Chain of Thought (CoT) over Baseline

| Method | Accuracy | Precision | Recall | F1 |
|----------|--------------|--------------|--------------|--------------|
| Baseline | 0.656 | 0.567 | 0.806 | 0.665 |
| CE & CoT | 0.671 | 0.582 | 0.794 | 0.671 |

The modest performance gains from CoT and citation extraction suggest that explicitly structuring the reasoning process through prompting does not meaningfully enhance model performance in statutory retrieval and QA tasks. Although the CoT prompting

includes specific guidance for citation processing and negative inference from missing citations (e.g., inferring “False” if no citation appeared relevant), this appears to primarily affect response format rather than retrieval and interpretive accuracy. This indicates that statutory retrieval and interpretation require more advanced techniques to handle than CoT and citation extraction.

Despite having minimal effect on performance, citation elicitation with CoT provides us with valuable insight because of the citation analysis done by the inference model on top of the retriever. Here, we prompt for CoT reasoning while also requesting that the inference model return the citation it found most “relevant” its answer or None if none are relevant. This citation will be picked out of the top- k retrieved citations (which are sorted by similarity using MIPS as per Section 4). Relevancy, in this case, is not rigidly defined and is determined by the LLM.

Figure 2 looks at the index i of the citation deemed most relevant by the LLM out of the top-5 fetched during retrieval, where $i = -1$ indicates the model found no results relevant. We observe that the LLM finds the first result ($i = 0$) most relevant almost half of the time, and finds no citation relevant almost a fifth of the time. Furthermore, when the LLM finds no citation law, recall is a remarkably low 0.12. A high false negative rate might be expected here as this includes situations where the retriever is unable to fetch the correct statute or regulation.

5.2.3 Citation Elicitation Performance. We now assess the accuracy of citations elicited. Unfortunately, because DOL has not released citations, this requires us to manually validate answers, which can

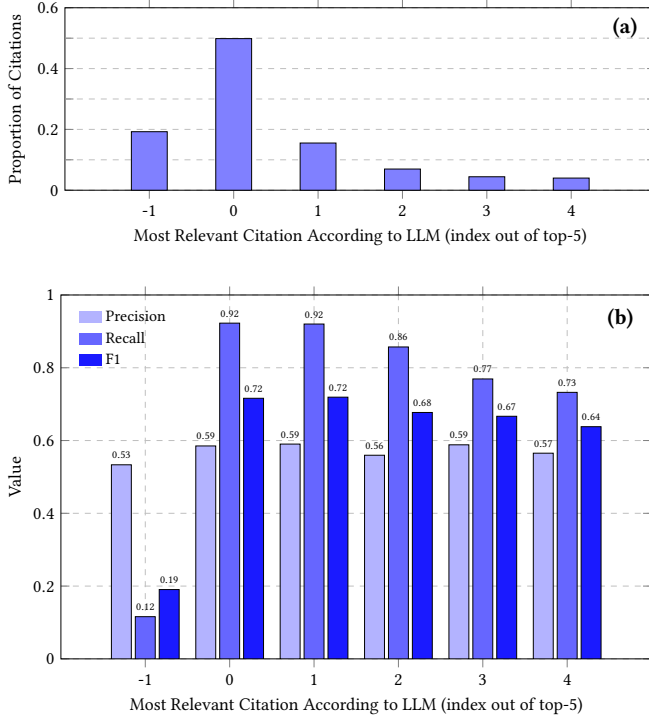


Figure 2: (a) distribution of the citation index produced during inference, representing the position of the returned citation among the top- k results (with 0 as the top result and -1 indicating no relevant citations) (b) performance metrics for each index.

be a time-consuming process given the complexity of the UI program and its provisions. We hence do this by manually reviewing a simple random sample of 30 outputs from experiments, performed with the GPT, Gemini, and Deepseek generators all using the OpenAI retriever. Each output includes the initial QA from LaborBench, the top-5 sections retrieved, the elicited citation—produced through CoT and citation elicitation—and the inferred answer. To evaluate an output, we locate the statute or regulation relevant to the query and review whether the elicited citation or any of the top-5 citations match.

We find that the elicited citations have an accuracy rate of 0.73 while OpenAI’s retriever achieves a recall@5 of 0.93. The accuracy, conditional on the correct citation appearing in the top-5, is only 0.77. Furthermore, conditional on the elicited citation being correct, the accuracy is 0.79.¹⁵ These results indicate that OpenAI’s retriever performs reasonably as an aide for statutory research, but citation accuracy as judged by the (generator) LLM remains limited, even when the appropriate citation was retrieved.

5.3 State-by-State Results. One important feature of LaborBench and StateCodes’s structure is that we are able to compare how the performance of LLMs differs between the state statutory codes. We observe that the states exhibit stark variations in both their

baseline knowledge, i.e., without retrieval, and in their knowledge with RAG.

Analyzing performance on queries made on each state’s respective statutory and regulatory code with RAG (averaged across all embeddings and LLMs), F1 scores vary from 0.52 (Virginia) to 0.81 (Arkansas). In a baseline setting, LLM performance varies from 0.316 (Louisiana) to 0.59 (Mississippi).

We illustrate our state-by-state RAG and baseline results in Figure 3. There, we sort the states by the difference between their F1 scores with RAG and at baseline. We consider this difference the *knowledge gained from RAG*. We see that the introduction of retrieval itself affects states differently. California receives relatively minimal gains in F1 score (0.05) from retrieval, while Idaho benefits considerably (0.30 gain in F1 score). Evidently, more work is necessary to understand how LLMs and retrievers handle state codes differently.

5.3.1 Correlates with Knowledge Gain From RAG. So far, our results suggest that code simplification efforts will benefit in divergent ways across states from AI. This raises the puzzle of why there is such diversity in state-level performance. Do differences in legislative processes, administrative organization, or economic demand drive such differential performance of AI systems?

Prior to inspecting other variables, we examine how population correlates with performance gain with RAG, given population’s strong influence on other economic and political factors. We observe a statistically significant negative correlation ($r = -0.40$, $p = 0.01$).¹⁶

We study how performance gain with RAG relates to variation in state policy and institutional metrics, obtained from Grossmann et al. [12]. We find no covariates (of 2,970) that have statistically significant predictive power in bivariate tests, adjusting for multiple testing [2]. We find similar results in regression analyses. Nonetheless, this combination of institutional and jurisdictional performance may be a fruitful inquiry enabled by LaborBench and StateCodes for other comparative work.

6 Conclusion

Through LaborBench and StateCodes, we provide an open framework for evaluating AI system performance in statutory retrieval, comparison, and interpretation. LaborBench and StateCodes offer a realistic setting for assessing LLMs ability to handle the complexity of statutory language. As calls for regulatory simplification increase pressure for employment of LLMs, robust and open means of evaluating statutory understanding such as LaborBench become increasingly important.

Notwithstanding widely touted efforts to employ AI for ambitious reform efforts, our results reveal that there is considerable room for improvement in the state-of-the-art RAG system’s performance in jurisdiction-specific QA. Although using retrieval provides significant gains in F1 and other techniques like CoT provide marginal gains, systems still are unable to ascertain salient dimensions of state UI law. While they may be used as aides for code understanding, current approaches appear to fall short of touted

¹⁵There are situations where the generator will produce the correct True/False answer, but will cite to the wrong section as proof.

¹⁶Performing a linear regression, we observe that Texas is an Tukey outlier ($k = 1.5$); without Texas, the correlation becomes considerably stronger ($r = -0.58$).

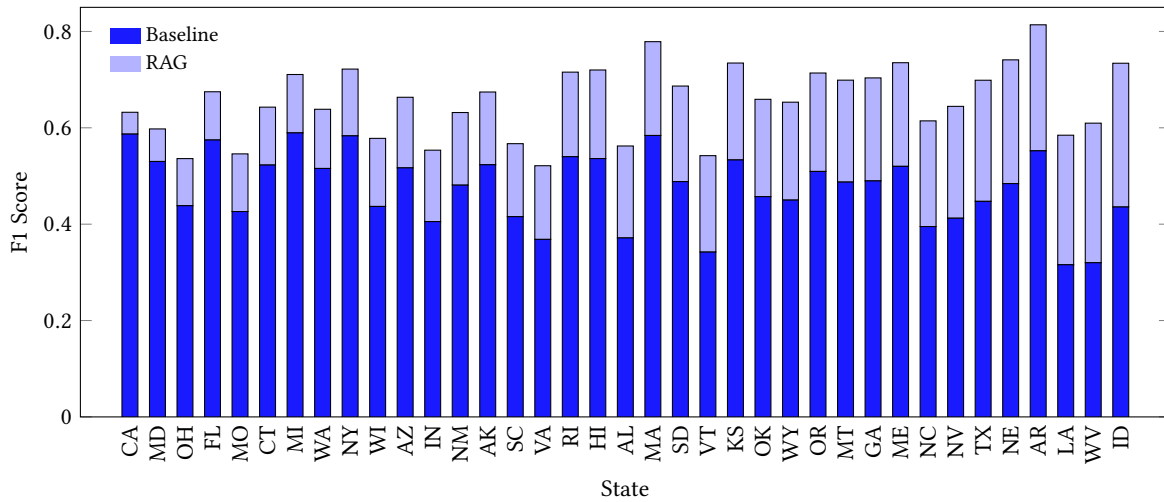


Figure 3: Mean F1 Scores With and Without RAG Across Jurisdictions (Sorted by RAG-Baseline Difference)

claims that AI can serve as end-to-end solutions for code simplification.

6.1 Future Work. First, we anticipate there are many retrieval strategies that may be applied to StateCodes to still increase LLM performance on LaborBench. We expect that many of those strategies that are specific to statutory retrieval and entailment, like those employed in COLIEE tasks 3 and 4 (e.g., Nguyen et al. [32]’s CAPTAIN) might improve performance on our benchmark. Modifications to prompting and methods such as context generation might also be employed to improve performance without retrieval. Adjusting retrieval granularity is another promising avenue: Lima [24] shows using hierarchical, multi-layer embeddings mirroring statutory structure—from sections down to individual clauses—could similarly boost LaborBench performance.

Second, we hope the completeness of StateCodes as well as its accessible structure enables much richer analysis of state statutory and regulatory law. For reasons articulated above, we focused here on the consequential and legally challenging area of UI law, but researchers can build on StateCodes to develop methods across many more domains.¹⁷

Third, the benchmark can be built out to represent the many years in which the state scan was completed by DOL. Fourth, while we have limited our inquiry to the (already challenging) boolean setting, our benchmark enables evaluation with the wider range of data types (categorical, integer, long and short form text) in LaborBench.

Last, another fruitful direction would be to test AI-assistance for processes like constructing DOL’s annual survey of state UI laws.

In sum, we hope that LaborBench and StateCodes will (1) enable more systematic assessment and tracking of statutory understanding, (2) help advance methods for statutory RAG, and (3) offer a useful resource for research in regulatory domains, including not

only the highly consequential domain of unemployment insurance, but beyond.

Acknowledgments

We are grateful to Neel Guha, Chris Manning, Derek Ouyang, Jen Pahlka, Amy Perez, Faiz Surani, Mirac Suzgun, Lucia Zheng, and participants of the RegLab meeting for comments and feedback. We also thank Faiz Surani for his help and recommendations in applying OCR for optimizing extraction from CSUILL.

References

- [1] Aptus.AI. 2023. Too many laws in Italy! The Italian regulatory maze... and how to get out of it. <https://www.aptus.ai/en/blog/too-many-laws-in-italy-the-italian-regulatory-maze-and-how-to-get-out-of-it/>
- [2] Yoav Benjamini and Yoel Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* 57, 1 (1995), 289–300.
- [3] Scott Burris. 2014. A Technical Guide for Policy Surveillance. <https://doi.org/10.2139/ssrn.2469895>
- [4] California Legislature. 2025. California Codes: Search. <https://leginfo.ca.gov/faces/codes.xhtml>
- [5] California Office of Administrative Law. 2025. California Code of Regulations - California Code of Regulations. <https://govt.westlaw.com/calregs/Index?transitionType=Default&contextData=%28sc.Default%29>
- [6] Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural Legal Judgment Prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Anna Korhonen, David Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 4317–4323. <https://doi.org/10.18653/v1/P19-1424>
- [7] 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)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 4310–4330. <https://doi.org/10.18653/v1/2022.acl-long.297>
- [8] DeepSeek-AI. 2024. DeepSeek-V3 Technical Report. <https://doi.org/10.48550/arXiv.2412.19437>
- [9] Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Songyang Zhang, Kai Chen, Zongwen Shen, and Jidong Ge. 2023. LawBench: Benchmarking Legal Knowledge of Large Language Models. <https://doi.org/10.48550/arXiv.2309.16289> [cs].
- [10] David Gilbert. 2025. DOGE Put a College Student in Charge of Using AI to Rewrite Regulations. <https://www.wired.com/story/doge-college-student-ai-rewrite-regulations-deregulation/> Section: tags.

¹⁷One limitation is that we were unable to access and structure a small number of states. We will continue to expand LaborBench to include reference citations for each answer when applicable. DOL has historically not included these, but their team appears willing to share this information prospectively. This would allow for testing of both retrieval and entailment separately, and also for testing metrics like recall@k.

- [11] Randy Goebel, Yoshinobu Kano, Mi-Young Kim, Juliano Rabelo, Ken Satoh, and Masaharu Yoshioka. 2024. Overview and Discussion of the Competition on Legal Information, Extraction/Entailment (COLIEE) 2023. *The Review of Socionetwork Strategies* 18, 1 (2024), 27–47. <https://doi.org/10.1007/s12626-023-00152-0>
- [12] Matt Grossmann, Marty P Jordan, and Joshua McCrain. 2021. The correlates of state policy and the structure of state panel data. *State Politics & Policy Quarterly* 21, 4 (2021), 430–450.
- [13] 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, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models. <http://arxiv.org/abs/2308.11462> arXiv:2308.11462 [cs].
- [14] Lee Harris. 2021. “A byzantine and high-bar system”: Governor pushes to saddle.... <https://nysfocus.com/2021/04/02/excluded-workers-fund-saddled-requirements>
- [15] Peter Henderson, Mark S. Krass, Lucia Zheng, Neel Guha, Christopher D. Manning, Dan Jurafsky, and Daniel E. Ho. 2022. Pile of Law: Learning Responsible Data Filtering from the Law and a 256GB Open-Source Legal Dataset. <https://doi.org/10.48550/arXiv.2207.00220> arXiv:2207.00220 [cs].
- [16] Austin Jenkins. 2024. Lawmakers Explore AI Benefits, Risks at National Summit. <https://pluribusnews.com/news-and-events/lawmakers-explore-ai-benefits-risks-at-national-summit/>
- [17] K Sparck Jones, Steve Walker, and Stephen E. Robertson. 2000. A probabilistic model of information retrieval: development and comparative experiments: Part 2. *Information processing & management* 36, 6 (2000), 809–840. Publisher: Elsevier.
- [18] Justia. 2024. U.S. Regulations & Administrative Codes. <https://regulations.justia.com/>
- [19] SK Karanam. 2020. A Radically New Approach to Simplifying Regulatory Compliance. *ABA Bank Compliance Magazine* 2020, January/February (Jan. 2020), 26. <https://www.wolterskluwer.com/en/expert-insights/a-radically-new-approach-to-simplifying-regulatory-compliance>
- [20] Vladimir Karpukhin, Barlas Ögüz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. <https://doi.org/10.48550/arXiv.2004.04906> arXiv:2004.04906 [cs].
- [21] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large Language Models are Zero-Shot Reasoners. <https://doi.org/10.48550/arXiv.2205.11916> arXiv:2205.11916 [cs].
- [22] Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftikhar Naim. 2024. Gecko: Versatile Text Embeddings Distilled from Large Language Models. <https://doi.org/10.48550/arXiv.2403.20327> arXiv:2403.20327 [cs].
- [23] 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. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. <http://arxiv.org/abs/2005.11401> arXiv:2005.11401 [cs].
- [24] João Alberto de Oliveira Lima. 2024. Unlocking Legal Knowledge with Multi-Layered Embedding-Based Retrieval. <https://doi.org/10.48550/arXiv.2411.07739> arXiv:2411.07739 [cs].
- [25] Brink Lindsey. 2021. *State Capacity: What Is It, How We Lost It, and How to Get It Back*. Study. Niskanen Center, Washington, DC.
- [26] Dan Lips. 2024. Ohio Is Using AI to Cut Red Tape. DOGE Should, Too. <https://www.city-journal.org/article/ohio-is-using-ai-to-cut-red-tape-doge-should-too/>
- [27] Daniel Locke and Guido Zuccon. 2022. Case law retrieval: problems, methods, challenges and evaluations in the last 20 years. <https://doi.org/10.48550/arXiv.2202.07209> arXiv:2202.07209 [cs].
- [28] Yuanhua Lv and Chengxiang Zhai. 2011. When documents are very long, BM25 fails!. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR '11)*. Association for Computing Machinery, New York, NY, USA, 1103–1104. <https://doi.org/10.1145/2009916.2010070>
- [29] Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D. Manning, and Daniel E. Ho. 2024. Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools. <https://doi.org/10.48550/arXiv.2405.20362> arXiv:2405.20362 [cs].
- [30] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, England.
- [31] Jeffrey A. (Jeffrey Alan) Meldman. 1975. *A preliminary study in computer-aided legal analysis*. Thesis. Massachusetts Institute of Technology. <https://dspace.mit.edu/handle/1721.1/27423> Accepted: 2005-09-16T21:34:00Z.
- [32] Chau Nguyen, Phuong Nguyen, Thanh Tran, Dat Nguyen, An Trieu, Tin Pham, Anh Dang, and Le-Minh Nguyen. 2024. CAPTAIN at COLIEE 2023: Efficient Methods for Legal Information Retrieval and Entailment Tasks. <https://doi.org/10.48550/arXiv.2401.03551> arXiv:2401.03551 [cs] version: 1.
- [33] OpenAI. 2024. New embedding models and API updates. <https://openai.com/index/new-embedding-models-and-api-updates/>
- [34] Jennifer Pahlka. 2023. *Recoding America*. St Martin's Press, New York, NY.
- [35] Shounak Paul, Rajas Bhatt, Pawan Goyal, and Saptarshi Ghosh. 2024. Legal Statute Identification: A Case Study using State-of-the-Art Datasets and Methods. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 2231–2240. <https://doi.org/10.1145/3626772.3657879>
- [36] Nicholas Pitone and Ghita Hourir Alami. 2024. LegalBench-RAG: A Benchmark for Retrieval-Augmented Generation in the Legal Domain. <https://doi.org/10.48550/arXiv.2408.10343> arXiv:2408.10343 [cs].
- [37] Redazione di Rainews. 2023. Casellati: “Il premierato è la riforma delle riforme. Il Colle resta garante dell’unità”. <https://www.rainews.it/articoli/2023/09/le-buone-leggi-semplificare-per-far-ripartire-litalia-segui-la-diretta-40e92fb6-c3b-491e-86cc-a0ed18a21392.html> Section: Politica.
- [38] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389. <https://doi.org/10.1561/15000000019>
- [39] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline M. Hancock-Beaulieu, and Mike Gatford. 1995. Okapi at TREC-3. In *Proceedings of the third text retrieval conference (TREC-3) (NIST special publication, Vol. 500-225)*, Donna K. Harman (Ed.). National Institute of Standards and Technology (NIST), Gaithersburg, MD, 109–126. <http://trec.nist.gov/pubs/trec3/papers/city.ps.gz>
- [40] John Schuettlinger, Tiffany Agnew, Nazriet Gebremeskel, and Will Potterveld. 2023. *Comparison of State Unemployment Insurance Laws*. Technical Report. Department of Labor Employment and Training Administration, Washington DC USA. <https://oui.doleta.gov/unemploy/pdf/uilawcompar/2023/complete.pdf>
- [41] 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*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, Punta Cana, Dominican Republic, 3784–3803. <https://doi.org/10.18653/v1/2021.findings-emnlp.320>
- [42] Weihang Su, Yiran Hu, Anzhe Xie, Qingyao Ai, Quezi Bing, Ning Zheng, Yun Liu, Weixing Shen, and Yiqun Liu. 2024. STARD: A Chinese Statute Retrieval Dataset Derived from Real-life Queries by Non-professionals. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 10658–10671. <https://doi.org/10.18653/v1/2024.findings-emnlp.625>
- [43] Texas Legislature. 2024. Texas Constitution and Statutes - Home. <https://statutes.capitol.texas.gov/>
- [44] Thomson Reuters. 2024. Generative AI Legal Research | Westlaw Precision. <https://legal.thomsonreuters.com/en/products/westlaw-precision>
- [45] U.S. Department of Health and Human Services. 2020. Regulatory Cleanup Initiative. *Federal Register* 85 (2020), 72899.
- [46] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text Embeddings by Weakly-Supervised Contrastive Pre-training. <https://doi.org/10.48550/arXiv.2212.03533> arXiv:2212.03533 [cs].
- [47] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., Virtual, 5776–5788. <https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>
- [48] Washington State Legislature. 2024. Revised Code of Washington (RCW). <https://app.leg.wa.gov/rcw/>
- [49] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. <https://doi.org/10.48550/arXiv.2201.11903> arXiv:2201.11903 [cs].
- [50] Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, and Jianfeng Xu. 2018. CAIL2018: A Large-Scale Legal Dataset for Judgment Prediction. <https://doi.org/10.48550/arXiv.1807.02478> arXiv:1807.02478 [cs].