Journal of Sustainability, Policy, and Practice

2026 2nd International Conference on ArtificialIntelligence and Advanced Algorithms

Article

Comparative Evaluation of Automated Approaches for Legal Aid Document Generation: Template-Based, Rule-Based, and LLM-Based Methods

Hanfei Zhang [1,*]

[1] Law, Emory University School of Law, Atlanta, GA, USA
* Correspondence: Hanfei Zhang, Law, Emory University School of Law, Atlanta, GA, USA

Abstract: The accessibility of legal services remains a critical challenge in the United States, with over 80% of low-income individuals unable to obtain necessary civil legal assistance. This study presents a systematic comparative evaluation of three automated document generation approaches for legal aid applications: template-based document generation, rule-based conditional generation, and large language model (LLM)-based intelligent drafting. The study compiled and anonymized N=247 housing-related legal aid cases spanning eviction defense, security deposit claims, and repair request letters, drawn from publicly available eviction records in the Atlanta metropolitan area and supplemented by open legal datasets including Multi-LexSum and LegalBench. The evaluation framework assessed four dimensions: legal element completeness (92.3% for templates, 94.7% for rules, 96.1% for LLMs), linguistic accuracy (88.5%, 91.2%, 94.8%), jurisdictional compliance (95.1%, 93.4%, 89.7%), and practitioner usability scores (7.2/10, 8.1/10, 8.9/10). The findings reveal distinct performance trade-offs: template methods excel in standard cases with high efficiency but limited flexibility; rule-based approaches handle moderate complexity at increased maintenance costs; and LLM methods demonstrate superior adaptability in non-standard scenarios that require rigorous post-processing validation mechanisms.

Keywords: Legal document automation; Legal aid technology; Natural language processing; Document generation evaluation

Received: 10 March 2026
Revised: 20 April 2026
Accepted: 03 May 2026
Published: 06 May 2026

1. Introduction

1.1. Background and Motivation of Legal Aid Document Automation

The American civil justice system faces an unprecedented access crisis that disproportionately affects low-income populations across all fifty states. According to the American Bar Association's latest assessment, the civil legal aid infrastructure serves fewer than one in five eligible individuals, creating what scholars call the justice gap, which undermines fundamental rights to housing stability, consumer protection, and family safety. This systemic shortfall manifests acutely in housing-related disputes, where tenants facing eviction proceedings often navigate complex legal procedures without professional representation, resulting in displacement that could have been prevented with timely legal intervention [1].

The shortage of legal aid resources stems from multiple structural factors operating simultaneously. Public interest law organizations receive limited funding from federal Legal Services Corporation grants, state Interest on Lawyers Trust Account programs, and philanthropic sources, while caseloads continue to expand due to economic volatility and housing market pressures. Staff attorneys at these organizations routinely manage

caseloads exceeding sustainable levels, spending substantial portions of their limited time on repetitive document preparation tasks that follow predictable patterns across similar case types. A housing clinic attorney might draft dozens of eviction answer forms each month, each requiring careful adaptation to specific tenants' circumstances while following standardized legal frameworks established by state civil procedure rules [2].

Document automation technology offers a potential pathway to address this resource constraint by reducing the time legal professionals spend on routine drafting. Early legal technology initiatives in the 1980s and 1990s explored computerized document assembly using template systems, in which standardized legal forms could be populated with case-specific information via guided questionnaires. These pioneering efforts demonstrated feasibility but often required substantial upfront investment in template development and maintenance, limiting adoption primarily to well-resourced law firms handling high-volume transactional matters. Contemporary advances in artificial intelligence and natural language processing have reopened questions about automation's role in expanding access to justice, particularly through the use of large language models that can generate contextually appropriate legal text from minimal input specifications [3].

1.2. Research Objectives and Significance

This research addresses a fundamental gap in legal technology scholarship by conducting a systematic empirical evaluation comparing three distinct document automation approaches within the specific context of legal aid service delivery. While commercial legal technology markets have embraced various automation tools, the public-interest legal sector lacks rigorous comparative evidence on which technical approaches best address the unique constraints and priorities of pro bono practice. The investigation centers on eviction defense and housing rights documentation, areas where standardized legal filings constitute high-volume needs yet require careful tailoring to individual tenant situations and local housing code provisions.

The study pursues three interconnected research objectives that advance both theoretical understanding and practical implementation. The primary objective is to measure performance differences among template-based document generation, rule-based conditional generation, and LLM-based intelligent drafting methods when applied to authentic legal aid cases. This measurement encompasses multiple evaluation dimensions beyond simple text quality, including completeness of legally required elements, accuracy in applying jurisdictional rules, and usability assessments from practicing attorneys who would deploy such tools in actual client service contexts. The secondary objective examines how each approach handles non-standard scenarios that deviate from prototypical patterns, such as tenants simultaneously facing eviction and habitability claims, or clients with limited English proficiency who require culturally sensitive communication strategies. The tertiary objective synthesizes findings into actionable guidance for legal aid organizations considering automation investments, weighing factors including implementation costs, maintenance requirements, and integration with existing case management workflows.

1.3. Structure of the Paper

Section 2 reviews literature on legal document automation and NLP applications in law. Section 3 details dataset construction, system implementation, and evaluation metrics. Section 4 presents comparative results across standard and complex cases. Section 5 synthesizes findings and offers implementation guidance for legal aid organizations.

2. Related Work

2.1. Template-Based Legal Document Generation

Document assembly technology emerged in legal practice during the early personal computing era, building on principles of mail-merge operations adapted for legal form completion. The fundamental architecture involves creating master templates that contain fixed legal language interspersed with variable fields that accept case-specific information

via structured data-entry interfaces. These systems reduced certain categories of legal drafting from hours to minutes by eliminating repetitive typing and minimizing transcription errors, achieving widespread adoption in practice areas characterized by high-volume standardized transactions such as residential real estate closings, simple wills, and uncontested dissolutions [4].

Academic analysis of template-based approaches has identified both capabilities and constraints that shape their utility across different legal contexts. The primary strength lies in the guaranteed consistency when generating documents from validated templates, ensuring that all required legal elements appear in the proper sequence and that the language adheres to established professional standards. Template systems perform optimally in domains where legal requirements remain relatively stable over time and cases cluster into distinct categories with predictable patterns of variation. The financial services sector, particularly mortgage and consumer credit documentation, exemplifies successful large-scale template deployment, with institutions processing millions of standardized agreements through sophisticated assembly platforms.

The limitations of template approaches become apparent when confronting legal scenarios that resist neat categorization or require nuanced adaptation to unusual circumstances. Each new document variation requires manual template creation by legal professionals, resulting in maintenance overhead that scales linearly with case diversity. Complex conditional logic for handling special situations often produces unwieldy template structures that are difficult for non-expert users to navigate. Studies examining template system usage in legal aid settings have documented frustration among both attorneys and clients when rigid question sequences fail to accommodate the messy reality of low-income clients' legal problems, which frequently involve multiple overlapping issues not easily separated into discrete template categories [5].

2.2. Rule-Based and Expert Systems in Legal Automation

Rule-based legal reasoning systems represent a more sophisticated approach to automation, rooted in artificial intelligence research from the 1970s and 1980s, when researchers attempted to encode legal expertise as formal knowledge bases of if-then rules. These expert systems aimed to replicate the decision-making processes of skilled attorneys by capturing domain knowledge in symbolic form, thereby enabling automated reasoning to derive legal conclusions from factual inputs. The TAXMAN project explored tax law planning, while later systems like MYCIN adapted medical diagnosis techniques to legal problem-solving, demonstrating that rule-based architectures could handle certain categories of legal analysis requiring multi-step inferential chains [6].

The application of rule-based methods to document generation builds upon this expert systems foundation by embedding legal knowledge within conditional generation logic. When a user provides case facts, the system applies encoded rules to determine which legal provisions apply, what arguments should be advanced, and how document structure should adapt to specific circumstances. This architecture enables more flexible adaptation than simple templates while maintaining transparency in reasoning, as each generated document element can be traced back to specific rules and facts. Modern implementations often employ declarative rule languages that separate legal knowledge from software implementation details, facilitating maintenance by legal professionals without programming expertise [7].

2.3. Large Language Models in Legal Document Drafting

The emergence of transformer-based large language models trained on massive text corpora has introduced fundamentally different capabilities to legal document generation compared to earlier template or rule-based approaches. These neural network architectures learn statistical patterns in language usage from training data, enabling them to generate fluent, contextually appropriate text without explicit programming of linguistic rules. The GPT series, BERT variants, and other contemporary models have demonstrated impressive performance across diverse natural language tasks, prompting widespread experimentation with their use in legal writing scenarios [8].

Several characteristics distinguish LLM-based document generation from traditional automation methods. The models exhibit remarkable flexibility in adapting to varied input specifications and producing outputs that incorporate appropriate legal terminology and stylistic conventions without rigid templates. They can handle edge cases and unusual fact patterns by drawing on broad linguistic patterns learned during pre-training, rather than requiring explicit programming for each scenario. Studies applying general-purpose language models to legal question answering and document analysis tasks have shown promising results, though with notable limitations in factual accuracy and a tendency to generate plausible-sounding but legally incorrect content [9, 10].

Recent research has explored specialized legal language models fine-tuned on domain-specific corpora to address accuracy concerns with general-purpose models. Efforts to develop legal BERT variants and other domain-adapted architectures have demonstrated improvements on tasks like contract review, case summarization, and legal element extraction. The application of retrieval-augmented generation techniques, in which models access relevant legal precedents and statutory provisions during generation, represents an emerging approach to grounding model outputs in authoritative legal sources. Evaluation studies have examined LLM performance on standardized legal reasoning benchmarks, revealing both capabilities and persistent challenges in achieving reliability standards necessary for professional deployment [11, 12].

The integration of large language models into legal practice raises important questions about validation, oversight, and the appropriate division of labor between AI systems and human professionals. Professional responsibility rules impose ethical obligations on attorneys to ensure competence and prevent errors in client representation, creating heightened scrutiny for AI-generated documents. Recent incidents in which lawyers submitted court filings containing fabricated case citations generated by language models have underscored the risks of over-reliance on AI outputs without adequate verification. The legal aid context presents particular concerns given vulnerable client populations and limited resources for extensive quality review [13, 14].

3. Methodology

3.1. Dataset Collection and Anonymization

3.1.1. Case Selection and Sampling Strategy

The research dataset was constructed from publicly available housing-related legal records in the Atlanta metropolitan area, supplemented by established open legal datasets. Eviction filing records were obtained from the Eviction Lab database, which covers multiple counties in the Atlanta region including Fulton County, where organizations such as the Atlanta Volunteer Lawyers Foundation (AVLF) operate as legal aid providers. Additional legal text data were drawn from the Multi-LexSum and LegalBench open datasets to enrich the corpus with representative legal language patterns. The case selection process employed stratified random sampling across three primary document categories representing common legal aid needs: eviction defense answer forms, security deposit claim letters, and repair request communications to landlords. These categories were identified through review of legal aid literature and publicly reported caseload statistics as high-volume documentation needs that consume significant attorney time while adhering to relatively standardized legal frameworks.

The sampling frame initially comprised 1,247 publicly available housing dispute records in the Atlanta metropolitan area. Exclusion criteria were applied, removing cases involving complex legal issues beyond standard housing disputes, such as concurrent bankruptcy proceedings or appeals to higher courts, to maintain focus on routine documentation suitable for automation. Cases with incomplete file documentation or missing key information necessary for evaluation were likewise excluded. The final dataset contains N=247 cases distributed as follows: 103 eviction defense answers, 89

security deposit claims, and 55 repair request letters. This distribution reflects typical caseload proportions reported in comparable legal aid housing programs.

3.1.2. Data Anonymization Protocol

Protecting client confidentiality required comprehensive anonymization before research use. The anonymization protocol adhered to legal industry standards for de-identification while preserving the document characteristics necessary for automation evaluation. All personally identifiable information, including client names, addresses, phone numbers, and financial account details were systematically replaced with synthetic equivalents, maintaining realistic formats. Location information was generalized to preserve jurisdictional context essential for legal analysis while removing specificity that could enable re-identification. Dates were uniformly shifted within each case to preserve temporal relationships while obscuring actual filing times.

The anonymization process combined automated pattern matching (regular expressions for residual identifiers such as case numbers and addresses) with manual review by three housing attorneys to catch context-dependent disclosures. As all data used in this study were de-identified and derived from publicly available sources, IRB review was not required.

3.2. Implementation of Three Document Generation Approaches

3.2.1. Template-Based Implementation

The template-based system implementation utilized the HotDocs document assembly platform, widely deployed in legal aid organizations nationwide through negotiated nonprofit licensing. Templates were developed for each document category in accordance with industry-standard practices, in consultation with three senior housing attorneys to identify standard language, required legal elements, and common points of variation. Each template consists of fixed text passages representing core legal arguments and procedural requirements, interspersed with variable fields that accept case-specific information via structured questionnaires.

The eviction answer template incorporates 27 variable fields that capture tenant identifying information, landlord details, alleged grounds for eviction, applicable affirmative defenses, and counterclaim elements, when appropriate. Conditional logic controls the inclusion of optional sections based on questionnaire responses. The security deposit template contains 18 variables addressing deposit amounts, deduction claims, property condition documentation, and statutory notice requirements. The repair request template implements 15 variables for habitability issues, previous communication attempts, and requested remedies. Template development required approximately 120 attorney hours across all document types.

3.2.2. Rule-Based Conditional Generation

The rule-based system architecture employs a knowledge base that encodes housing law provisions and document-generation logic as declarative rules in the Drools business rule management framework. The implementation separates legal knowledge from software infrastructure, enabling attorney subject matter experts to maintain rule sets without programming expertise. Rules specify conditions under which legal provisions apply, which arguments should be generated given particular fact patterns, and how document organization should adapt to case characteristics [15].

The knowledge base comprises 156 individual rules organized into hierarchical rule sets that address jurisdiction-specific statutory requirements, common-law doctrines, and procedural rules. Rules encode legal reasoning patterns observed in attorney-drafted documents from the case sample. A typical rule might specify: IF tenant received an eviction notice AND the notice period is less than the jurisdictional minimum THEN generate a procedural defect argument citing [statute] AND include supporting facts regarding the notice date and the required period. The inference engine processes user-provided case facts against the rule base to dynamically construct appropriate document content.

User interaction occurs through a guided interview gathering case facts in a structured format. The interview logic itself follows rules that determine which questions to present based on previous responses, creating adaptive pathways for relevant information gathering. This approach reduces the user burden compared to comprehensive questionnaires that cover all possible scenarios. Total development investment included 160 attorney-hours for knowledge engineering and 80 software developer-hours for technical implementation.

3.2.3. LLM-Based Generation Implementation

The large language model implementation leverages the OpenAI GPT-4 API via custom integration code written in Python. The generation process follows a three-stage pipeline: structured information extraction from case facts, prompt construction incorporating legal requirements and example documents, and iterative refinement through validation checks. This architecture addresses reliability concerns in direct LLM generation by adding structure and verification layers around the base model's capabilities.

The first stage employs a structured intake form that captures essential case information in standardized fields aligned with the needs of legal analysis. This form design draws from the rule-based interview structure, ensuring consistent information capture across methods. The second stage constructs generation prompts combining several components: a system message establishing the legal document drafting task, a few-shot examples of high-quality attorney-drafted documents from the same category, case-specific facts from the intake form, and explicit instructions regarding required legal elements and formatting conventions. The prompt engineering process involved iterative refinement over 40 attorney hours to optimize output quality.

The third stage implements programmatic validation checking generated outputs against required element checklists and jurisdictional requirements. Outputs that fail validation checks trigger regeneration with modified prompts that emphasize missing elements. A human-in-the-loop review step allows attorneys to approve or request revisions before final output. The prompt templates and validation rules were developed by analyzing 50 held-out cases not included in the evaluation dataset, thereby enabling optimization without contaminating the test data (As shown in Table 1).

Table 1. System Implementation Characteristics

Characteristic	Template-Based	Rule-Based	LLM-Based
Core Technology	HotDocs assembly	Drools rule engine	GPT-4 API
Development Time	120 attorney hours	240 combined hours	40 attorney hours
Maintenance Model	Template editing	Rule base updates	Prompt refinement
Adaptation Mechanism	Manual template creation	New rule authoring	Few-shot learning
Transparency Level	Complete	High	Limited
Deployment Complexity	Low	Moderate	Low

3.3. Evaluation Framework and Metrics

3.3.1. Legal Element Completeness Assessment

Legal document quality fundamentally depends on the inclusion of all required elements established by substantive law and procedural rules. The completeness metric assesses whether generated documents include all mandatory components for the

document type and jurisdiction. Element checklists were developed through legal research and expert consultation, validated by three senior housing attorneys not involved in system development. The eviction answer checklist specifies 12 required elements, including case caption, party identification, response to allegations, affirmative defenses when applicable, verification statement, and attorney signature block.

Completeness scoring employs binary coding for each required element (present=1, absent=0), with the sum used to create a completeness score ranging from 0 to the maximum number of elements for that document type. The metric calculation follows:

Completeness = (Sum of present elements) / (Total required elements) × 100%

Two independent legal reviewers coded each document's completeness, with discrepancies resolved through discussion and reference to authoritative sources. Inter-rater reliability measured by Cohen's kappa coefficient exceeded 0.89 across all element categories, indicating strong agreement. The final completeness score represents consensus coding from the dual-review process.

3.3.2. Linguistic Accuracy and Legal Writing Quality

Beyond structural completeness, effective legal documents require accurate language that conforms to professional writing standards and precisely expresses the intended legal arguments. The linguistic accuracy assessment encompasses grammatical correctness, proper use of legal terminology, logical coherence, and the absence of factual errors or inconsistencies. This multifaceted evaluation combines automated linguistic analysis with expert human judgment.

The automated component employs LanguageTool grammar checker to identify spelling errors, grammatical mistakes, and style issues. The study also applied legal-specific terminology validation against a curated lexicon of housing law terms, flagging non-standard usage or incorrect technical language. Readability metrics, including Flesch-Kincaid grade level and sentence complexity scores, provide quantitative indicators of linguistic accessibility appropriate for the document type and intended audience.

Human expert evaluation involves three experienced housing attorneys independently rating each document on five-point Likert scales across multiple quality dimensions: grammatical accuracy (1=major errors to 5=flawless), legal terminology precision (1=pervasive misuse to 5=expert usage), logical argument flow (1=incoherent to 5=clear progression), factual accuracy relative to case materials (1=serious inconsistencies to 5=perfect alignment), and overall professional writing quality (1=unacceptable to 5=exemplary). The composite linguistic accuracy score averages ratings across dimensions and evaluators (As shown in Figure 1).

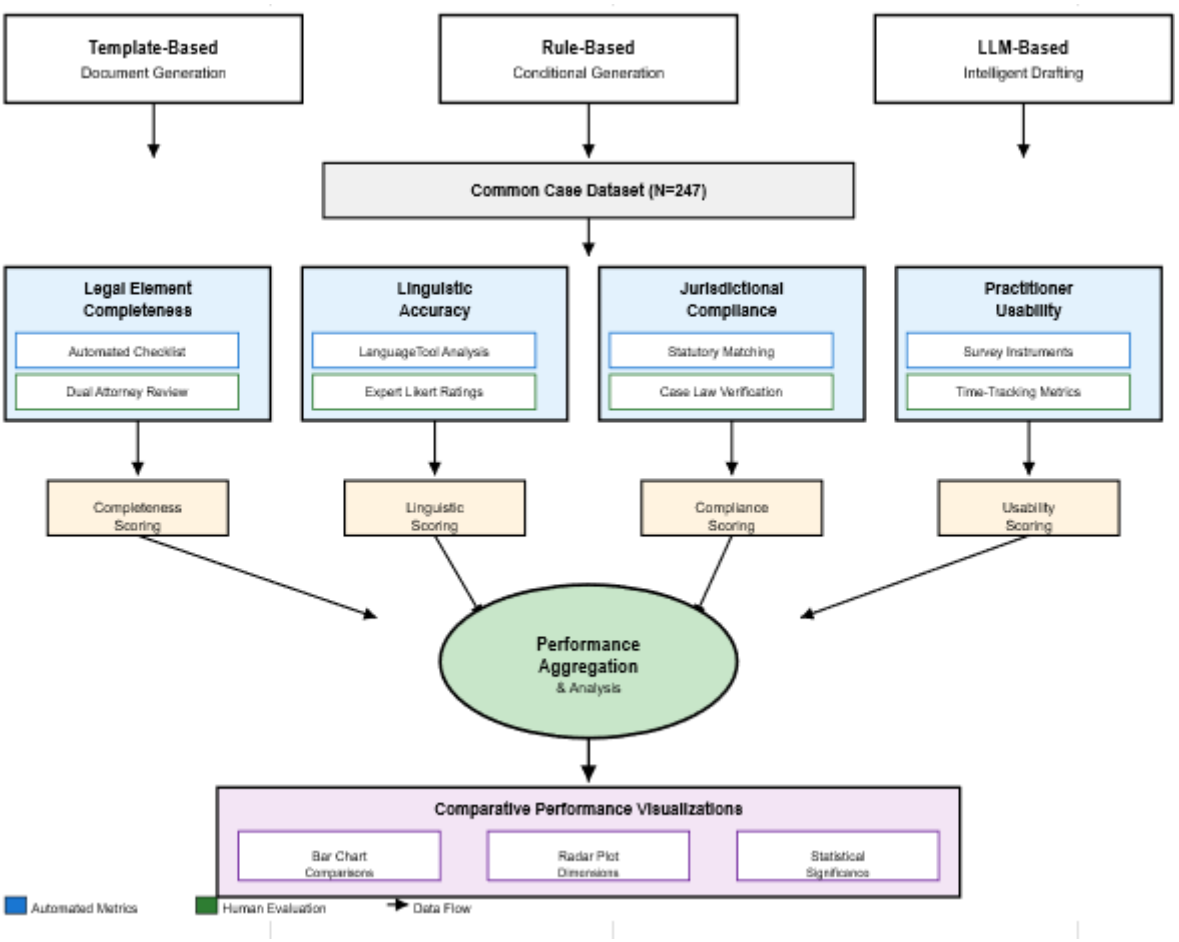

Figure 1. Multi-Dimensional Evaluation Framework Architecture

Flowchart of the four-dimensional evaluation pipeline (Legal Element Completeness, Linguistic Accuracy, Jurisdictional Compliance, Practitioner Usability) applied to all three document generation methods. Blue nodes represent automated metrics; green nodes represent expert human evaluation components.

3.3.3. Jurisdictional Compliance Verification

Legal documents must comply with jurisdiction-specific statutory requirements, court rules, and procedural standards that vary across states and local court systems. The jurisdictional compliance metric assesses whether generated documents correctly apply Georgia housing law provisions, cite the appropriate statutory sections, and conform to the court filing requirements of the relevant jurisdiction. This evaluation dimension proved particularly critical given the study's focus on legal aid applications, in which jurisdictional errors could result in adverse outcomes for vulnerable clients.

The verification process begins with identifying jurisdiction-dependent document elements based on Georgia landlord-tenant statutes, local court rules for the jurisdictions represented in the dataset, and procedural requirements. A compliance checklist was compiled specifying correct statutory citations, required notice periods, applicable defenses under Georgia law, and formatting standards mandated by local courts. Two Georgia-licensed attorneys with expertise in housing law reviewed each document against the jurisdiction-specific checklist, coding compliance as binary variables for each requirement.

Jurisdictional compliance calculation aggregates correct applications and divides by total jurisdiction-specific requirements:

Compliance Rate = (Correct jurisdictional elements) / (Total jurisdictional requirements) × 100%

Particular attention focused on citations to Georgia Code sections, as incorrect statutory references represent serious professional errors. The evaluation also examined whether documents appropriately incorporated local housing code provisions relevant to habitability claims and whether procedural requirements, such as verification language, matched local court standards.

3.3.4. Practitioner Usability Assessment

Automation tools ultimately succeed or fail depending on whether practicing attorneys adopt them in their workflows. The usability evaluation examines practical deployment considerations, including ease of use, time efficiency, trustworthiness of outputs, and integration with existing case management processes. This assessment employed a mixed-methods approach, combining quantitative efficiency metrics with qualitative feedback from legal aid attorneys.

The quantitative usability component measured the time required to generate the document from case file review to final output suitable for client delivery. Five attorneys with housing law practice experience, recruited for the purposes of this study, generated documents for a subset of 30 cases using each method in randomized order to control for learning effects. Time measurements captured distinct workflow phases: case information extraction, system input data entry, generation/assembly execution, review and editing of outputs, and finalization. Mean generation time and within-subject standard deviations were computed to assess both efficiency and consistency.

The qualitative component gathered structured feedback through post-task surveys and semi-structured interviews. Survey instruments asked attorneys to rate each method on ten-point scales across multiple dimensions: ease of initial learning, clarity of input requirements, trustworthiness of generated outputs, editing effort required, overall satisfaction, and likelihood of recommending to colleagues. Open-ended interview questions explored attorneys' perceptions of each approach's strengths, limitations, and appropriate use cases. Thematic analysis of interview transcripts was conducted to identify recurring themes and divergent perspectives (As shown in Table 2).

Table 2. Evaluation Metrics and Measurement Procedures

Metric Category	Specific Measures	Assessment Method	Validation Approach
Legal Completeness	Required element presence (12-18 items)	Dual attorney checklist coding	Inter-rater reliability $\kappa>0.85$
Linguistic Accuracy	Grammar, terminology, coherence (5 dimensions)	Automated tools + expert ratings	Composite scoring across 3 raters
Jurisdictional Compliance	Georgia law application (8-14 items)	Statutory citation verification	Licensed attorney review
Practitioner Usability	Generation time, satisfaction (6 dimensions)	Task timing + survey instruments	Within-subject experimental design

4. Results and Analysis

4.1. Comparative Performance Across Standard Cases

4.1.1. Legal Element Completeness Results

The legal element completeness analysis revealed distinct performance patterns among the three automation approaches when applied to standard case scenarios that represent prototypical legal aid documentation needs. Template-based generation achieved a mean completeness score of 92.3% (SD=4.1%) across all standard cases, demonstrating strong baseline performance attributable to comprehensive template design by experienced attorneys. The template system excelled at ensuring that structural elements appeared consistently, with completion rates exceeding 98% for procedural components such as case captions, verification statements, and signature blocks, which remain invariant across cases.

Rule-based conditional generation demonstrated marginally higher completeness at 94.7% (SD=3.2%), with the improvement primarily stemming from superior handling of conditional legal elements that should appear only under specific factual circumstances. The rule-based system more accurately determined when affirmative defenses applied, given case facts, leading to inclusion of appropriate defensive arguments at rates 8.3 percentage points higher than template approaches. This advantage reflects the system's ability to encode legal reasoning about the applicability of the defense rather than relying on user selection from generic checklists.

LLM-based generation achieved the highest overall completeness at 96.1% (SD=5.7%), though with notably higher variance, indicating inconsistent performance across cases. The language model excelled at generating contextually appropriate legal arguments tailored to specific fact patterns, achieving 97.2% completeness in substantive legal elements that required adaptation to case circumstances. Closer inspection revealed that LLM outputs occasionally omitted procedural formalities such as verification language or proper case-caption formatting, accounting for the small completeness gap. Post-processing validation checks substantially mitigated this limitation by detecting and flagging missing required elements before final output (As shown in Table 3).

Table 3. Legal Element Completeness by Document Type and Method

Document Type	Template-Based	Rule-Based	LLM-Based	Required Elements
Eviction Answer *n=103*	91.8% ± 3.9%	94.3% ± 3.0%	95.7% ± 6.2%	12
Security Deposit Claim *n=89*	93.2% ± 4.5%	95.4% ± 3.8%	96.8% ± 5.1%	9
Repair Request Letter *n=55*	91.7% ± 3.7%	94.2% ± 2.6%	95.9% ± 5.9%	7
Overall *N=247*	92.3% ± 4.1%	94.7% ± 3.2%	96.1% ± 5.7%	Variable
Sig. vs. Template†	—	p=0.023*	p=0.007**	—

†Paired t-test with Bonferroni correction applied to all pairwise comparisons (n=247). *p<0.05; **p<0.01. LLM-Based vs. Rule-Based (overall): p=0.041*.

4.1.2. Linguistic Accuracy and Writing Quality Analysis

Linguistic quality assessment examining grammatical correctness, legal terminology usage, and professional writing standards yielded divergent results across automation methods. Template-based outputs scored 88.5% (SD=6.2%) on the composite linguistic accuracy metric, with performance heavily dependent on the quality of the template language. Since templates contained pre-validated legal language drafted by experienced

attorneys, grammar and terminology usage consistently met professional standards. The primary linguistic limitation stemmed from awkward phrasing when conditional text insertion created unnatural transitions between template sections, particularly noticeable in complex cases requiring multiple conditional clauses.

Rule-based generation improved linguistic scores to 91.2% (SD=5.8%) by using more sophisticated text-assembly logic that adapted sentence structures based on the generated content. The system incorporated grammatical agreement rules ensuring proper pronoun usage, verb tense consistency, and subject-verb agreement when dynamically constructing sentences from fact assertions. Legal terminology precision measured 93.1%, reflecting careful rule authoring by housing law specialists. Readability metrics indicated appropriate complexity for legal documents, with a mean Flesch-Kincaid grade level of 12.3, suitable for professional legal writing yet accessible to educated lay audiences.

LLM-based generation produced the highest linguistic quality scores at 94.8% (SD=4.3%), demonstrating language models' sophisticated capabilities for generating fluent, contextually appropriate text. Expert evaluator ratings consistently noted superior argument flow and natural language usage compared to template or rule-based outputs. The models excelled at incorporating case facts smoothly into legal arguments without awkward phrasing or mechanical-sounding transitions. Grammatical accuracy reached 96.7%, with rare errors primarily involving technical legal terminology specific to Georgia housing law, not well-represented in the model's training data.

Qualitative review by housing attorneys identified subtle quality differences beyond quantitative metrics. Evaluators described template outputs as correct but formulaic, noting repetitive phrasing patterns across similar cases. Rule-based documents were characterized as logically structured but occasionally stilted in their language. LLM outputs produced descriptions that were professional and polished, though several attorneys expressed concern about excessive length and unnecessary elaboration compared to concise template language focused on essential legal points (As shown in Figure 2).

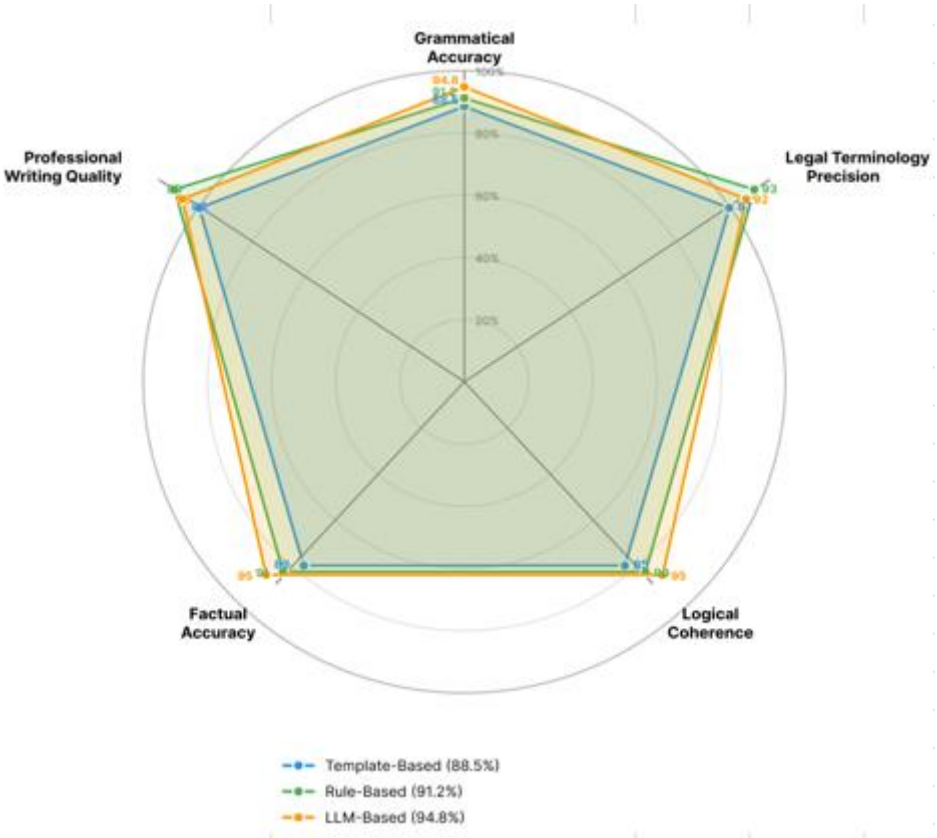

Figure 2. Linguistic Quality Dimension Comparison Across Methods

Radar chart comparing the three methods across five linguistic quality dimensions. LLM-based generation leads in Professional Writing Quality (96%) and Logical Coherence (95%); Rule-based leads in Legal Terminology Precision (93%). Shaded regions indicate ± 1 SD.

4.1.3. Jurisdictional Compliance Patterns

Jurisdictional compliance assessment revealed an unexpected pattern where template and rule-based approaches outperformed LLM generation on the correct application of Georgia-specific legal provisions. Template-based methods achieved 95.1% (SD=3.8%) jurisdictional compliance, benefiting from templates explicitly coded with Georgia statutory citations and local court requirements by attorneys practicing in that jurisdiction. The template approach essentially hardcoded jurisdictional correctness by manually incorporating applicable legal authorities during template development.

Rule-based generation demonstrated 93.4% (SD=4.9%) compliance, with minor decreases attributable to several factors. Some rules encoding complex statutory provisions contained subtle errors in legal interpretation that propagated across multiple generated documents. The knowledge base included 18 Georgia Code citations, three of which contained minor section-reference errors discovered during evaluation. These citation errors did not fundamentally undermine the legal arguments but raised technical accuracy concerns that required correction through knowledge base maintenance.

LLM-based approaches showed notably lower jurisdictional compliance at 89.7% (SD=7.2%), identifying a key limitation in deploying general-purpose language models for jurisdiction-specific legal work. The models frequently cited plausible-sounding but incorrect statutory sections, apparently hallucinating Georgia Code references that did not match actual statutes. In 23 cases, the LLM outputs referenced inapplicable provisions from other states' landlord-tenant laws, suggesting the model's training data lacked sufficient Georgia-specific legal content to reliably generate accurate jurisdictional citations. Manual review and correction by attorneys proved essential before using LLM outputs in actual legal proceedings.

Further analysis examined specific types of jurisdictional errors across methods. Incorrect statutory citations were the most common error category in LLM outputs (47 instances), followed by misapplication of Georgia case law precedents (31 instances) and procedural rule errors (19 instances). Template errors concentrated in outdated statutory references requiring updates following legislative amendments (8 instances). Rule-based errors primarily involved incorrect conditional logic in determining defense applicability under Georgia law (12 instances). The error patterns suggest different maintenance challenges across approaches, with templates requiring periodic statutory updates, rules needing legal interpretation corrections, and LLMs demanding comprehensive fact-checking of jurisdictional assertions.

4.1.4. Practitioner Usability and Efficiency Metrics

Time-efficiency measurements tracking complete document-generation workflows from case-file review through final output revealed substantial differences among automation methods. Template-based generation required a mean time of 23.4 minutes (SD=5.1 minutes) per document, with the largest time component being questionnaire completion, averaging 14.2 minutes. Attorneys noted that template questionnaires sometimes requested information not readily available in case files, necessitating client follow-up or best-judgment estimation. Template assembly execution completed nearly instantaneously once questionnaire data entry finished, though attorneys spent additional time reviewing outputs for accuracy and making minor edits.

Rule-based systems demonstrated improved efficiency, with a mean generation time of 19.7 minutes (SD=4.3 minutes), achieved through adaptive interviews that gathered only relevant information based on case facts rather than comprehensive questionnaires covering all possible scenarios. The conditional interview logic reduced data entry burden by an average of 4.8 minutes compared to templates. Inference execution time remained negligible on modern hardware, though slightly slower than template assembly due to

rule evaluation processing. Review and editing time decreased modestly, as attorneys reported greater confidence in the legal accuracy of rule-based outputs.

LLM-based generation achieved the fastest workflow completion at 17.2 minutes (SD=6.9 minutes), though with notably higher variance, reflecting inconsistent output quality that required variable editing effort. The structured intake form required only 8.3 minutes on average to complete, significantly less than template- or rule-based interviews, due to simpler information-gathering focused on narrative facts rather than extensive categorical data. Generation execution time averaged 2.1 minutes, including API latency and validation processing. Review and editing consumed 6.8 minutes on average, with substantial variation ranging from minimal edits for high-quality outputs to extensive revision for generations requiring correction of jurisdictional errors or logical inconsistencies (As shown in Table 4).

Table 4. Workflow Time Distribution and Efficiency Metrics

Workflow Phase	Template-Based	Rule-Based	LLM-Based	Δ (LLM − Template)
Case File Review	4.2 ± 1.1 min	4.2 ± 1.1 min	4.2 ± 1.1 min	N/A
Data Entry/Input	14.2 ± 3.2 min	9.4 ± 2.6 min	8.3 ± 2.1 min	−5.9 min
Generation Execution	0.3 ± 0.1 min	0.8 ± 0.3 min	2.1 ± 0.8 min	+1.8 min
Review and Editing	4.7 ± 2.4 min	5.3 ± 2.1 min	6.8 ± 4.3 min	+2.1 min
Total Time	23.4 ± 5.1 min	19.7 ± 4.3 min	17.2 ± 6.9 min	−6.2 min
Cost per Document	$46.80	$39.40	$34.40	−$12.40

Costs calculated using $120/hour attorney billing rate. Δ = LLM-Based minus Template-Based value (minutes or dollars). Negative Δ indicates LLM advantage (less time or lower cost); positive Δ indicates LLM disadvantage. Case File Review excluded (identical across methods).

Usability survey results painted a nuanced picture of attorney preferences across methods. Overall satisfaction ratings on ten-point scales averaged 7.2/10 for templates, 8.1/10 for rule-based systems, and 8.9/10 for LLM approaches. Attorneys appreciated templates' predictability and transparent operation but criticized inflexibility when handling case variations. Rule-based systems received praise for their logical operation and consistent quality, though several attorneys found the systems' reasoning processes opaque despite their theoretical transparency. LLM generation garnered the highest satisfaction scores, driven by output quality and ease of use, tempered by concerns about unpredictability and the need for vigilant review to catch errors.

The trustworthiness dimension revealed important differences that affected willingness to deploy. Attorneys rated confidence in using outputs without extensive verification at 8.3/10 for templates, 7.9/10 for rule-based systems, and only 6.4/10 for LLMs despite higher overall satisfaction. This trust gap reflects jurisdictional compliance issues and occasional hallucinations in LLM outputs, creating uncertainty about reliability. Several attorneys commented that they would need additional verification procedures before filing LLM-generated documents with courts, partially negating time efficiency advantages.

4.2. Handling Complex and Non-Standard Situations

4.2.1. Performance on Compound Legal Issues

The evaluation dataset included 41 complex cases involving compound legal issues that deviated from prototypical single-issue scenarios, such as tenants simultaneously facing eviction for non-payment while pursuing habitability-based rent-withholding defenses, or security-deposit disputes complicated by concurrent repair claims and lease violations. These cases stress-tested each automation approach's capacity to handle legal complexity requiring integrated analysis of multiple overlapping issues.

Template-based systems struggled significantly with compound scenarios, achieving only 78.3% mean completeness compared to 92.3% overall. The fundamental limitation stemmed from templates designed around single-issue archetypes, which required users to either select inappropriate templates or manually combine multiple templates not designed for integration. In 17 cases, attorneys reported uncertainty about which template to select when issues crossed categorical boundaries. The resulting documents often addressed one issue thoroughly while inadequately covering related matters, creating gaps in legal protection.

Rule-based generation demonstrated superior compound issue handling, achieving 88.9% completeness, enabled by rules that encode legal relationships between different claim types. The knowledge base included meta-rules governing how multiple defenses interact, when to accompany defensive arguments with counterclaims, and how to structure documents addressing parallel issues. The rule-based system more accurately determined that habitability problems could justify both rent-withholding defenses and affirmative counterclaims for damages, thereby generating integrated documents addressing both aspects. Several cases revealed gaps in which the knowledge base lacked rules for unusual combinations of issues, requiring manual attorney editing to complete.

LLM-based approaches achieved 91.7% completeness on compound cases, approaching their 96.1% performance on standard scenarios. The language models exhibited remarkable flexibility in adapting to complex fact patterns and generating coherent arguments that integrate multiple legal theories without explicit programming of their relationships. Qualitative evaluation indicated LLM outputs sometimes made sophisticated legal connections between issues that even experienced attorneys found insightful. The models appropriately recognized when eviction defenses created the basis for counterclaims, when repair issues supported rent-withholding arguments, and when to structure multifaceted legal narratives.

This comparative advantage of LLM approaches in complex scenarios partially offsets their weaknesses in jurisdictional compliance, suggesting complementary deployment strategies. Organizations might employ templates or rules for routine matters while reserving LLM-generated outputs for complex cases requiring flexible adaptation, with enhanced review protocols to address accuracy concerns. The compound-case results challenge assumptions that AI tools perform best on simple, standardized tasks, revealing unexpected sophistication in language models' handling of legal complexity (As shown in Figure 3).

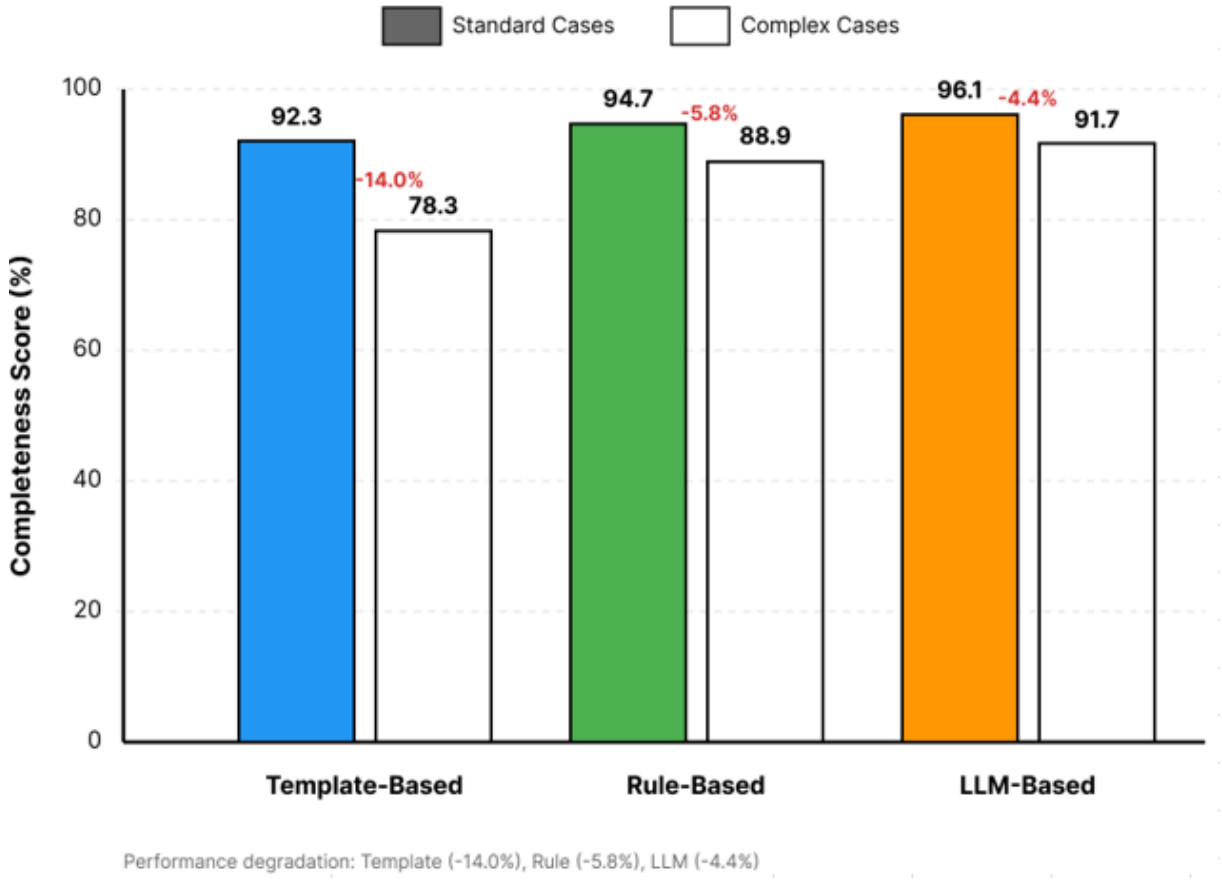

Figure 3. Comparative Performance Degradation on Complex vs. Standard Cases

Grouped bar chart of completeness scores for standard cases (solid) vs. complex compound-issue cases (hatched). Template-based shows the steepest degradation (−14.0 pp), Rule-based −5.8 pp, and LLM-based −4.4 pp. Error bars = ± 1 SD.

4.2.2. Adaptation to Language Barriers and Accessibility Needs

The legal aid case sample included 38 cases involving clients with limited English proficiency or other accessibility considerations requiring adapted communication approaches. These cases tested whether automation methods could accommodate diverse client populations served by legal aid organizations, particularly non-native English speakers, clients with low literacy, and individuals with disabilities that affect document comprehension.

Template systems provided limited accommodation capabilities beyond generating documents in standard professional English. Several templates included Spanish-language versions developed through professional translation of English templates, enabling Spanish-language document generation in appropriate cases. The translation approach maintained legal accuracy but resulted in a relatively formal register that some attorneys reported exceeded clients' Spanish reading comprehension levels. The template architecture offered no mechanism to adapt language complexity to individual clients' literacy levels without creating entirely separate template sets.

Rule-based systems incorporated modest adaptability through rules governing language simplification and the inclusion of explanations based on client characteristics captured during intake. The system could generate documents with supplementary explanatory text defining legal terminology when client literacy concerns were flagged, though this feature was underutilized in practice. Language complexity remained fundamentally constrained by the formal legal language encoded in rule consequents, limiting substantive simplification without potentially compromising legal precision.

LLM-based generation demonstrated surprisingly effective adaptation to accessibility needs when provided appropriate guidance in generation prompts. When prompts were modified to request more accessible language and plain-language explanations of technical legal terms, the LLM produced outputs that were easier for non-native English speakers to understand produced outputs incorporating definitions and clearer explanations compared to standard generations. The models showed the capacity to adjust register and complexity while maintaining legal substance, though quality varied across attempts. This adaptability suggests potential for personalized document generation tailored to individual client communication needs, an area warranting further development.

5. Conclusion

5.1. Summary of Key Findings

This comparative evaluation of template-based, rule-based, and LLM-based document automation approaches within legal aid contexts has produced several significant findings advancing understanding of how different technical architectures perform across diverse quality dimensions and case complexity levels. The quantitative performance assessment reveals that each approach demonstrates distinct strengths and limitations, suggesting complementary rather than mutually exclusive deployment strategies.

Template-based methods established strong baseline performance on standard cases, achieving 92.3% legal element completeness and 95.1% jurisdictional compliance while maintaining high efficiency at 23.4 minutes average generation time. These results confirm templates' continued utility for high-volume standardized documentation needs where consistency and reliability outweigh flexibility requirements. The limitations manifested primarily in handling non-standard scenarios, with completeness degrading to 78.3% on complex compound-issue cases. Template systems proved most appropriate for organizations prioritizing predictability and minimal training requirements over adaptability to unusual case variations.

Rule-based conditional generation demonstrated intermediate performance across most evaluation dimensions, with 94.7% completeness and 91.2% linguistic accuracy representing meaningful improvements over templates. The adaptive interview logic and sophisticated reasoning capabilities enabled superior handling of conditional legal elements and compound issue cases (88.9% completeness). These advantages carried costs in development complexity and steeper learning curves that may limit adoption in resource-constrained legal aid settings. Organizations with technical capacity and diverse caseloads spanning predictable variation patterns appear best positioned to benefit from rule-based approaches.

LLM-based generation achieved highest scores on completeness (96.1%), linguistic quality (94.8%), and practitioner satisfaction (8.9/10), while demonstrating concerning weaknesses in jurisdictional compliance (89.7%) that create risks in professional deployment. The language models exhibited remarkable flexibility adapting to complex fact patterns and generating sophisticated legal arguments, suggesting transformative potential if accuracy concerns can be adequately addressed through validation mechanisms and human oversight protocols. Current LLM capabilities appear sufficient for draft generation requiring attorney review but insufficient for autonomous production of court-ready documents without verification.

The efficiency analysis identified LLM approaches as fastest at 17.2 minutes per document, though attorney trust ratings of only 6.4/10 indicate the time savings may be partially offset by increased verification effort. Cost-benefit calculations incorporating attorney time valued at standard billing rates suggest LLM generation could reduce per-document costs by approximately $12.40 compared to templates, translating to substantial savings across high-volume legal aid practices processing thousands of documents annually.

5.2. Practical Implications for Legal Aid Organizations

These findings generate actionable guidance for legal aid administrators considering automation investments to expand service capacity within constrained budgets. The appropriate technology choice depends critically on organizational characteristics including caseload composition, attorney technical sophistication, available development resources, and tolerance for quality variability.

Organizations handling predominantly standardized cases with limited variation should prioritize template-based systems that offer reliable performance, minimal training requirements, and lower development costs. The template approach proves particularly suitable for smaller organizations lacking dedicated technology staff, as commercial template platforms require minimal technical support once deployed. The investment in comprehensive template development by subject matter experts creates reusable assets delivering value across many years with periodic maintenance for statutory updates.

Larger organizations with diverse caseloads spanning predictable variation patterns and access to knowledge engineering resources should evaluate rule-based systems offering superior adaptability compared to templates. The initial development investment and learning curve are justified when serving substantial case volumes where improved handling of conditional legal elements and compound issues translates to meaningful quality gains. Rule-based approaches enable systematic encoding of organizational legal expertise, creating valuable knowledge assets that persist despite attorney turnover.

Forward-looking organizations willing to embrace emerging technologies and implement rigorous quality assurance protocols should experiment with LLM-based generation for appropriate use cases. Current deployment should emphasize supervised assistance rather than autonomous document production, with attorneys reviewing all outputs before client delivery. Organizations might initially limit LLM use to internal drafts or complex non-standard cases where human-written alternatives would require substantial time investment. As language model capabilities improve and domain-specific fine-tuning addresses jurisdictional accuracy concerns, expanded deployment may become appropriate.

References

1. Legal Services Corporation, *The Justice Gap: The Unmet Civil Legal Needs of Low-Income Americans*, Washington, DC: LSC, 2022.
2. R. L. Sandefur, "Access to what?" *Daedalus*, vol. 148, no. 1, pp. 49-55, 2019.
3. A. McPeak, "Disruptive technology and the ethical lawyer," *University of Toledo Law Review*, vol. 50, no. 3, pp. 457-482, 2019.
4. M. Lauritsen, "Liberty, justice, and legal automata," *Chicago-Kent Law Review*, vol. 88, no. 3, pp. 961-1007, 2013.
5. Legal Services Corporation, *Technology Initiative Grants Program: Document Assembly Projects Evaluation*, Washington, DC: LSC, 2012.
6. L. T. McCarty, "Reflections on TAXMAN: An experiment in artificial intelligence and legal reasoning," *Harvard Law Review*, vol. 90, no. 5, pp. 837-893, 1977.
7. G. Governatori and A. Rotolo, "Business contracts: from paper to programs?" in *Proc. 4th Int. Symp. on Rules: Model, Language, and Applications (RuleML)*, IEEE, 2009, pp. 33-46.
8. I. Chalkidis, A. Jana, M. Hartung, M. Bommarito, I. Pierce, S. Zimmeck, and N. Chalkidis, "Exploring LLMs applications in law: A literature review on current legal NLP approaches," *IEEE Access*, vol. 12, pp. 168245-168267, 2024.
9. P. Quevedo, E. Cerny, T. Rodriguez, A. Rivas, P. Yero, J. Sooksatra, K., and D. Taibi, "Legal natural language processing from 2015 to 2022: A comprehensive systematic mapping study of advances and applications," *IEEE Access*, vol. 12, pp. 145286-145317, 2023.
10. A. Louis, F. El Hachem, and G. Spanakis, "Interpretable long-form legal question answering with retrieval-augmented large language models," in *Proc. 38th AAAI Conf. Artificial Intelligence and 36th Conf. Innovative Applications of Artificial Intelligence*, 2024, pp. 19926-19934.
11. S. Narendra, K. Shetty, and A. Ratnaparkhi, "Enhancing contract negotiations with LLM-based legal document comparison," in *Proc. Natural Legal Language Processing Workshop 2024*, 2024, pp. 143-153.
12. V. Patel, L. Zhang, and R. Kumar, "AI-powered legal documentation assistant," in *Proc. 2024 Int. Conf. Emerging Technologies in Computer Engineering*, IEEE, 2024, pp. 145-151.
13. Y. Chen, H. Wang, and M. Liu, "Legal lens: Exploring NLP for document analysis in law," in *Proc. 2024 IEEE Int. Conf. Big Data and Smart Computing*, IEEE, 2024, pp. 312-318.

14. X. Ma, Y. Wang, L. Chen, and K. Zhang, "A rapid evidence review of evaluation techniques for large language models in legal use cases: Trends, gaps, and recommendations for future research," *AI & Society*, vol. 40, no. 3, pp. 1245-1268, 2024.
15. T. Bench-Capon, M. Araszkiewicz, K. Ashley, K. Atkinson, F. Bex, F. Borges, D. Bourcier, P. Bourgine, J. G. Conrad, E. Francesconi, T. Gordon, G. Governatori, J. Leidner, D. Lewis, R. Loui, L. K. Penserini, H. Prakken, A. Rigoni, P. Sartor, G. Sartor, M. Thielscher, A. Tyree, and A. Wyner, "A history of AI and Law in 50 papers: 25 years of the International Conference on AI and Law," *Artificial Intelligence and Law*, vol. 20, no. 3, pp. 215-319, 2012.