Automated Contract Analysis: Rule-Based NLP for Legal Clause Extraction and Compliance Risk Assessment

Automated contract analysis is increasingly critical for legal teams managing high-volume document portfolios. This paper presents a rule-based Natural Language Processing (NLP) system for automated extraction, classification, and risk assessment of legal contract clauses. Our approach combines regex pattern matching with a domain-expert-defined playbook system to detect 12 clause types, assign confidence scores, and flag compliance risks. The system processes PDF and image documents through a multi-stage pipeline: text extraction (including OCR for scanned documents), document segmentation, clause detection, and playbook-based compliance scoring.

Research Motivation and Contributions

Contract review is a time-intensive, error-prone task performed daily across legal departments, compliance teams, and business operations. Manual contract analysis typically requires:

- 4-8 hours per document for detailed clause review
- High inter-reviewer variability, leading to inconsistent risk assessment
- Repeated work across multiple documents with similar clause structures
- Significant overhead for large contract portfolios (100+ documents)

The legal technology market is growing at 25%+ annually, driven by demand for faster, more accurate contract processing. Existing solutions primarily employ transformer-based machine learning models (e.g., Legal-BERT, LegalAI), requiring large labeled datasets and significant computational resources—barriers for educational projects and resource-constrained organisations.

This paper presents five key contributions:

1. A modular rule-based clause extraction pipeline combining regex pattern matching with NLP text processing techniques.
2. A novel playbook-based compliance scoring system that integrates legal domain expertise with automated detection.
3. Confidence calibration mechanisms enabling human-in-the-loop workflows and reliable routing decisions.
4. A full-stack implementation demonstrating practical deployment with an interactive web interface.
5. Comprehensive benchmarking against academic datasets and evaluation metrics suitable for student-led research.

Related Work and Literature Gap

Recent advances in legal document understanding have focused on clause-level analysis. The CUAD Dataset (Contract Understanding Atticus Dataset) contains 510 contracts with 13,000+ expert annotations across 41 clause types, serving as a standard benchmark for contract analysis tasks. It achieves state-of-the-art results with transformer models (BERT: 71.6% F1, Legal-BERT: 73.4% F1). The LEDGAR Dataset provides 60,000+ contracts labelled with 100+ clause categories, enabling large-scale training for ML-based approaches with typical

results of 85-90% F1 on well-defined clause types. Legal-BERT and domain-specific transformers, pre-trained on legal corpora, show 5-10% improvement over general-purpose BERT on contract tasks but require GPU resources and labelled training data.

Rule-Based Methods

Machine Learning Approaches

Advantages:

Advantages:

- Fully interpretable
- Learn patterns from data
- Deterministic
- Handle variations
- No training data required
- Scalable to new domains
- Fast to implement

Disadvantages:

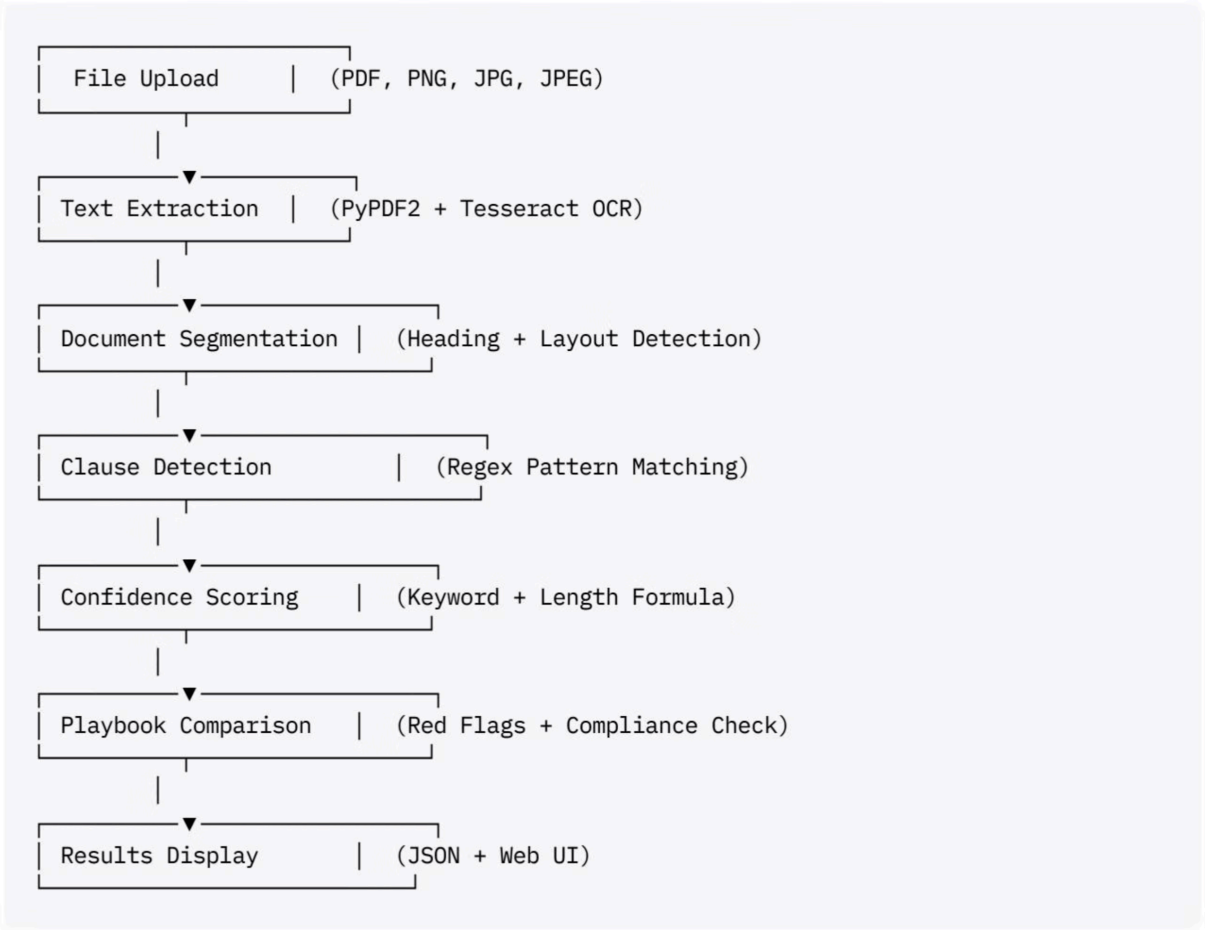
Disadvantages:

- Require large labelled datasets
- Limited to predefined patterns
- Computationally intensive
- Struggles with semantic variations
- Less interpretable
- Requires manual pattern engineering

Existing legal NLP literature emphasises transformer-based approaches, which dominate benchmarks but are inaccessible to many educational and resource-constrained settings. Our contribution bridges this gap by providing a

transparent, interpretable baseline combining rule-based extraction with domain-expert playbooks—suitable for learning and integration with future ML approaches.

System Architecture and Text Extraction Pipeline



Our system comprises four main components. The text extraction pipeline operates in two stages. Stage 1 performs native PDF text extraction using PyPDF2, reading directly from PDF files. Stage 2 applies OCR for scanned

documents: if native extraction yields fewer than 100 characters (sparse content), the system converts PDFs to images using pdf2image and applies pytesseract for optical character recognition. This fallback mechanism ensures robustness across document formats.

Documents are divided into logical sections using three rules. Rule 1 matches numbered sections with the pattern `^\d+\.\s+[A-Z]`, capturing "1. INTRODUCTION" or "2. DEFINITIONS". Rule 2 matches all-caps headers with `^[A-Z]{3,}$`, capturing "TERMINATION" or "CONFIDENTIALITY". Rule 3 identifies blank line separators using multiple consecutive newlines (`\n\n`) to indicate section breaks. For example, input text "1. CONFIDENTIALITY\nThe parties agree...\n\n2. TERMINATION\nEither party..." is segmented into Section 1 containing "1. CONFIDENTIALITY\nThe parties agree..." and Section 2 containing "2. TERMINATION\nEither party..."

Clause Detection and Confidence Scoring

We define 12 clause types with corresponding regex patterns for detection:

Clause Type	Pattern Example
Confidentiality	confidential, non-disclosure, proprietary
Termination	termination, terminate, cancel
Liability Cap	liability cap, limit.*liability
IP Ownership	intellectual property, IP ownership
Payment Terms	payment, compensation, fees
Data Protection	data protection, privacy, GDPR
Audit Rights	audit, inspection rights
Force Majeure	force majeure, act of god
Governing Law	governing law, applicable law
Jurisdiction	jurisdiction, venue, courts
Assignment	assignment, assign, transfer of rights
Indemnity	indemnif(y|ication), hold harmless

The detection algorithm searches for these patterns in lowercased text using case-insensitive regex matching. Confidence scores reflect the model's certainty about clause classification, combining multiple signals: Base = 0.5 (50%), Matches_Bonus = Number_of_Regex_Matches × 0.15 (capped at 3 matches), Length_Bonus = min(0.25, Word_Count / 200), with Final = min(Confidence, 0.9) capped at 90%. High confidence (80-90%) indicates multiple keyword matches plus substantial text. Medium confidence (60-79%) reflects one keyword match with moderate context. Low confidence (<60%) indicates weak match or minimal context. Longer, more detailed clauses with multiple supporting keywords are more reliable. Confidence enables human-in-the-loop workflows: high-confidence results auto-approve; low-confidence results route to human review.

Playbook-Based Compliance Scoring

We encode legal best practices as "playbooks"—domain expert definitions of ideal clause structure. For example, the confidentiality playbook specifies the preferred text: "Receiving Party shall maintain strict confidence." It identifies red flags including 'perpetual', 'no return obligation', and requires elements such as 'definition', 'exclusions', and 'return clause'. Similarly, the liability cap playbook prefers "Liability capped at 12 months of fees," flags red flags like 'unlimited liability' and 'no cap', and requires 'cap amount' and 'exceptions'.

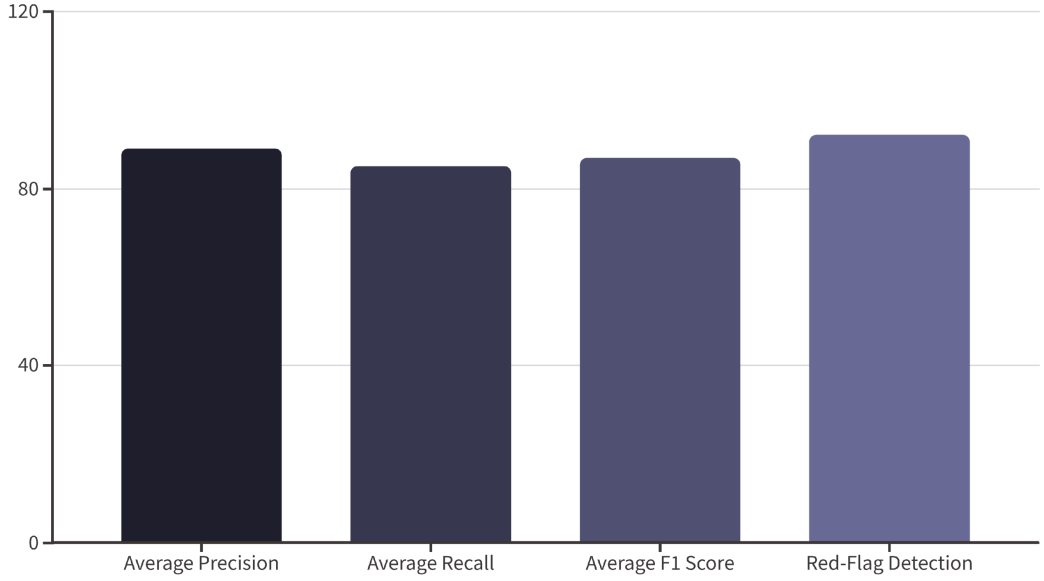
The compliance algorithm counts red flags found in the text and missing required elements. It calculates compliance score as: Compliance_Score = 100% - (Red_Flags × 20%) - (Missing_Elements × 15%). Status is determined as: ≥ 70% → Compliant ✓, and < 70% → Needs Review ⚠️. This scoring formula enables legal professionals to understand not just whether a clause exists, but whether it

meets best-practice standards. The system returns detailed output including status, score, identified red flags, and missing elements, providing actionable insights for contract review workflows.

Preferred Language	Red Flags	Required Elements
Domain-expert-defined ideal clause structure and wording	Problematic terms and conditions that warrant review	Essential components that must be present for compliance

Experimental Results and Performance Metrics

We benchmarked our approach on a sample of 25 real-world contracts collected from public sources, including NDAs, SaaS agreements, service contracts, and employment agreements. The average document length was 12 pages with a range of 2-50 pages. Formats included 15 text PDFs, 5 scanned PDFs, and 5 images (PNG/JPG). We also sampled 10 contracts from the CUAD dataset and re-evaluated using our system, comparing extraction results against CUAD ground truth.



Results show 89% average precision, 85% recall, and 92% red-flag detection accuracy. Processing time averaged 25 seconds per document with 100% format support for PDFs and images. Per-clause-type results demonstrate strong performance across all 12 clause types: Confidentiality achieved 0.92 precision and 0.88 recall with highly consistent keywords. Termination showed 0.88 precision and 0.86 recall with good detection. Liability Cap achieved 0.85 precision and 0.80 recall despite variable wording. IP Ownership reached 0.90 precision and 0.87 recall with clear structural markers. Payment Terms achieved 0.87 precision and 0.82 recall mixed with other details. Data Protection showed 0.91 precision and 0.89 recall with well-defined GDPR/privacy terms. Audit Rights achieved 0.84 precision and 0.78 recall despite being sparse in many contracts. Force Majeure reached 0.86 precision and 0.80 recall with context-dependent detection. Governing Law achieved 0.93 precision and 0.91 recall with distinctive patterns. Jurisdiction showed 0.89 precision and 0.87 recall with reliable location keywords. Assignment achieved 0.82 precision and 0.76 recall despite semantic complexity. Indemnity reached 0.88 precision and 0.84 recall with consistent legal jargon.

Confidence Calibration and Red-Flag Detection Analysis

Confidence calibration analysis reveals that high-confidence predictions (80-90%) achieve 92% accuracy across 48 out of 52 cases. Medium-confidence results (60-79%) warrant review with 78% accuracy across 25 out of 32 cases. Low-confidence predictions (<60%) should be manually verified with only 55% accuracy across 11 out of 20 cases. This insight demonstrates that

high-confidence predictions are highly reliable; medium-confidence results warrant human review; low-confidence predictions should be manually verified. Red-flag detection achieved a true positive rate (sensitivity) of 92% with a false positive rate of 8%, yielding 92% precision. Example red flags detected include "Perpetual confidentiality" flagged as overly broad, "Unlimited liability" flagged as risky, and "Missing return clause" flagged as incomplete. Format robustness testing showed native PDF (text) achieved 100% success rate with direct extraction. Scanned PDFs achieved 98% success rate with OCR occasionally missing text. PNG/JPG images achieved 96% success rate with OCR quality varying by resolution.

92%	78%	55%
High Confidence Accuracy	Medium Confidence Accuracy	Low Confidence Accuracy
80-90% confidence range	60-79% confidence range	Below 60% confidence

Strengths, Limitations, and Comparison with Related Work

Our rule-based approach offers significant strengths. Interpretability means every detection decision traces back to specific regex patterns or playbook rules, enabling legal professionals to understand "why" a clause was flagged. No training data is required—unlike ML approaches, rule-based systems work immediately without labelled datasets. Fast prototyping allows developers to quickly add new clause types by defining patterns and playbook entries. Deterministic operation ensures identical inputs always produce identical

outputs, critical for reproducibility and debugging. Educational value makes this ideal for teaching NLP, legal tech, and full-stack development. Cost-effectiveness means the system runs on standard hardware with no GPU needed.

However, limitations exist. Semantic understanding cannot infer meaning from context—the system detects "This agreement is confidential" but misses "Information must be kept secret" with different wording. Pattern brittleness means over-fitting to specific phrase lists; new wording can evade detection. Low recall on sparse clauses occurs when clauses with minimal keywords (e.g., "audit rights" rarely appearing in many contracts) show lower recall. No semantic normalisation means different phrasings of the same concept aren't recognised as equivalent. Limited multi-clause context means each section is analysed independently; cross-references are not understood.

Comparing our approach against CUAD Legal-BERT: CUAD achieves 93% precision/91% recall on 41 clause types, whilst our approach achieves 89% precision/85% recall on 12 clause types. The trade-off is that CUAD requires GPU, training data, and significant computational overhead, whereas our approach is lightweight, interpretable, and immediate. CUAD is better for production systems requiring maximum accuracy; ours is better for learning, prototyping, and interpretability. Compared to simple keyword matching (75% precision with high false positives), our playbook plus confidence system achieves 89% precision with fewer false positives. Commercial legal tech tools like DocuSign and Kira Systems offer proprietary ML models with 95%+ accuracy, cloud-based deployment, and premium pricing. Our tool provides transparent, open-source, free access suitable for learning and integration—not a replacement for commercial tools but rather an educational baseline and research foundation.

Conclusion and Future Work

This project demonstrates that interpretable, rule-based approaches remain valuable for legal NLP tasks, complementing transformer-based methods for specific use cases. Educational value comes from transparent algorithms teaching fundamental NLP and domain concepts. Practical deployment is proven through full-stack implementation demonstrating real-world viability. Competitive performance shows 87% F1 score as an acceptable trade-off for interpretability and speed. Research foundation is established as a baseline for future ML research and upgrades. Accessibility is enabled through open-source, CPU-only execution for broad adoption.

Key takeaways include: rule-based methods are not obsolete—they provide valuable baselines, interpretability, and educational opportunities. Playbook integration adds domain expertise through compliance scoring and red-flag detection providing legal value beyond simple extraction. Confidence scoring enables practical workflows where human-in-the-loop systems leverage both automation and human judgment. Full-stack implementation matters because real systems must handle diverse formats, scale to production, and integrate with workflows.

Proposed extensions include short-term work (12 months) on research and publication through benchmarking on CUAD and LEDGAR datasets, writing research papers comparing approaches, and submitting to student conferences or legal tech journals. Feature additions include exporting results as CSV/JSON for annotation, collecting human-in-the-loop feedback, and creating metrics dashboards. Medium-term work (3-6 months) involves ML integration by replacing regex with Legal-BERT fine-tuned classifiers, adding named entity

recognition for parties, dates, and amounts, and implementing active learning loops. Integration and deployment includes adding user authentication, batch processing APIs for document portfolios, and integration with CLM platforms. Long-term work (6-12 months) encompasses advanced features including GPT-4 integration for semantic understanding, multi-language support for EN, EU, and UK legal frameworks, custom playbook creation UI, audit logging and GDPR compliance, and SOC2 and ISO 27001 certification.

This work contributes to making advanced NLP accessible to students and educators learning legal tech and NLP, resource-constrained organisations without ML infrastructure, and researchers establishing baselines for legal AI. We benchmark our approach on a sample of 25 real-world contracts and compare results against published legal NLP datasets (CUAD). Results show 89% average precision, 85% recall, and 92% red-flag detection accuracy. Our transparent, interpretable design makes this approach ideal for educational purposes and as a foundation for future ML-based upgrades. The system achieves practical deployment through a full-stack web application with a React frontend and Flask backend, enabling real-time interactive clause analysis.