LEGAL CLAUSE EXTRACTION USING TEXT GENERATION MODELS

Vanellsa Acha

University of California, Berkeley

Vacha20@berkeley.edu

Abstract

Automating clause extraction from legal contracts poses unique challenges due to the density, variability, and ambiguity of legal language. This study benchmarks three generative transformer models—T5, FLAN-T5, and a custom BART Fusion architecture—on the Contract Understanding Atticus Dataset (CUAD) to evaluate their effectiveness in producing clause-level answers for predefined legal questions. By reframing clause extraction as a generative task, we test each model's ability to generate complete, contextually grounded clauses, rather than simply identifying token spans. FLAN-T5 delivered the most consistent results for exact extraction, benefiting from instruction tuning and stronger clause fluency. BART Fusion, enhanced with cross-chunk fusion and classification layers, outperformed in detecting clause presence and reconstructing partial clauses from scattered text, especially for legally nuanced categories like Post-Termination Services and Revenue Sharing. Notably, clause frequency was a poor predictor of accuracy; structurally distinct clauses were more learnable than frequent but diffuse types like "Parties." These findings highlight the value of hybrid strategies that combine classification-driven clause detection with generation precision—offering practical implications for scalable, contract-aware legal ML systems.

1. Introduction

Extractive question answering (QA) has emerged as a powerful paradigm in natural language processing (NLP), where a model identifies and extracts a span of text from a passage that directly answers a given question. Large pretrained transformers like BERT and its successors have demonstrated strong performance in understanding and generating human-like text, achieving or surpassing human benchmarks on datasets such as SQuAD 2.0 (Rajpurkar et al., 2018) and SuperGLUE (Wang et al., 2019). This progress raises an important question: can such models transfer effectively to specialized, high-stakes domains?

Contract review is a resource-intensive process requiring legal professionals to identify clauses tied to specific provisions or entities and interpret their implications within the broader contractual context—an effort that, if automated, could significantly streamline analysis while preserving legal rigor. The legal domain review presents distinct challenges for NLP applications. Contract review documents specifically are characterized by complex syntax, domain-specific terminology, and intricate clause structures. The language is often formal, verbose, and laden with legal jargon, making it fundamentally different from the general-domain texts most models are trained on. Legal texts also exhibit high variability in how obligations and provisions are expressed, often requiring domain-specific terminology, multi-sentence reasoning and interpretation of implicit references. These traits complicate clause-level extraction and demand a level of precision and reliability not always achievable by general-purpose models.

This study explores the application of text generation models—specifically T5, BART, and FLAN-T5—for clause extraction in legal contracts. By framing clause extraction as a generative task, we aim to assess the efficacy of these models in identifying and extracting relevant clauses corresponding to predefined legal categories. Our approach seeks to bridge the gap between the capabilities of general-domain NLP models and the specialized requirements of legal document analysis. to a highly specialized and closed-domain context: legal contract analysis. We evaluate their ability to output precise, clause-level answers to predefined questions about legal provisions, using the CUAD dataset as a benchmark. In doing so, we aim to understand the suitability and limitations use generative models for clause extraction in a domain where language and stakes are both high.

2. Literature Review

The application of NLP in the legal domain has seen gradual growth, particularly in extracting structured information from unstructured legal texts, such as legislation, court decisions, and contracts (Almeida et al., 2020; Hendrycks et al., 2021). However, tasks like party extraction have received comparatively limited attention. Only a few studies have directly addressed the identification of contractual entities.

Chalkidis et al. (2017) proposed a rule-based system for extracting parties from contracts, but its scalability was constrained by the rigidness of hand-crafted rules, which struggled to generalize across the diversity of contract formats encountered in practice. Leivaditi et al. (2020) targeted a specific contract type—leases—thus limiting the broader applicability of their system. More recently, Hendrycks et al. (2021) approached the task with a question-answering system, framing party extraction as an extractive QA problem. While this strategy enabled domain-specific adaptation, the method led to a high rate of false positives and inconsistent extraction quality.

2.1 Related work

Recent efforts have explored transformer-based models for clause-level classification and extraction tasks in contracts. The Contract Understanding Atticus Dataset (CUAD), introduced by Hendrycks et al. (2021), represents a cornerstone in this area. CUAD includes over 13,000 annotated clauses across 41 legal categories and has served as a benchmark for clause extraction using transformer models, including fine-tuned BERT and RoBERTa. The primary task is formulated as span prediction: for each category, the model identifies the relevant clause(s) in a contract by predicting the start and end token positions. This enables the model to "highlight" relevant content for legal professionals.

Prior legal NLP research has largely approached clause and entity extraction through span prediction or classification tasks. The **CUAD dataset** (Hendrycks et al., 2021) has become a standard benchmark in this space, supporting fine-tuned BERT and RoBERTa models to predict start-end token spans for 41 legal categories. While effective in some domains, these models often struggle with clause fragmentation, implied references, and non-contiguous spans. For instance, **Sivapiran et al. (2023)** applied contextual span representations to extract contracting parties, yet their method remains limited to narrow entity types.

Our work extends this by evaluating generative models capable of producing full clause outputs—offering more flexibility in handling diverse clause structures. This shift addresses limitations in token-bound span prediction and aligns better with legal contracts’ structural complexity.

3. Data

This study utilizes the Contract Understanding Atticus Dataset (CUAD), which consists of 510 commercial legal contracts annotated by legal experts with over 13,000 labeled clauses spanning 41 legal categories such as "Governing Law," "Termination for Convenience," and "Confidentiality." Contracts vary significantly in length—from 7 to over 150 pages—with word counts ranging from 8,000 to over 36,000 words per document. On average, labeled clauses make up only 10% of each contract; given 41 categories, this means roughly 0.25% of the text is relevant per clause type, creating a highly sparse signal for models to learn from. CUAD provides answer_start positions but not answer_end, and answers for a single question may appear in multiple, non-contiguous spans—highlighting a structural limitation of extractive approaches. We address this by framing clause extraction as a text generation task, training models to produce the full clause text directly based on the contract and the legal category prompt. For this study, we focus mainly on the labeled clauses heavily under sampling questions with no answers within CUAD, framing the clause extraction task as a text generation problem.

4. Methodology

4.1 Dataset Restructuring and Preprocessing

Given the substantial length and variability of CUAD contracts—some exceeding 36,000 words—we adopt a chunking-based strategy to manage input size while preserving legal clause integrity. Our initial approach divided contracts into fixed 512-token chunks, assigning each question only to the first chunk. This setup led to limited question exposure across the document and ultimately poor performance, as much of the text was processed without any associated prompt or clause relevance signal.

To address this, we implemented a revised preprocessing pipeline inspired by the method proposed in *Party Extraction from Legal Contracts Using Contextualized Span Representations* (Sivapiran et al., 2023). In this setup, contracts are divided into overlapping 512-token chunks using a 256-token sliding window. Each chunk is paired with a clause-type-specific question, and a binary flag is used to indicate whether the answer is present in that chunk. When no relevant clause is found, the target is set to a consistent placeholder ("No answer"), which proved more stable than alternative fillers during experimentation. This restructuring led to a highly imbalanced dataset, with a large majority of chunks lacking relevant content (248,260 "no answer" vs. 22,070 "answer present"). To mitigate this, we under sampled "no answer" examples to constitute 40% of the training data, a ratio that offered the best downstream performance among the configurations we tried.

The resulting dataset consists of question-context-answer triplets, each representing a localized view of the contract. Each example includes the document ID, contract title, clause type, a natural language question, the input chunk of contract text, the expected output (i.e., clause text or a placeholder), and a binary flag for answer presence. An illustrative example is shown below:

Modeling

4.2 T5 Baseline: Initial Performance and Limitations

As a first baseline, we fine-tuned a T5-small model to perform clause extraction using question-context-answer triplets from the CUAD dataset. While this offered a straightforward framing of clause extraction as a sequence generation task, the model frequently underperformed—especially in accurately identifying clause spans within legal language.

One clear pattern was the model's over-reliance on predicting "No answer," even when relevant clauses were explicitly present. This issue was particularly acute for clause types like *Parties*, *Effective Date*, and *Anti-Assignment*, which often appear in subtle or varied phrasings. For instance, in response to questions about "Irrevocable or Perpetual License," the model would incorrectly output "No answer" despite the presence of explicit language such as "a perpetual, irrevocable, worldwide license…".

Further clause-type breakdowns revealed that the model struggled most with categories involving abstract legal language or long entity spans. The *Parties* clause exhibited the highest false negative rate, with outputs like "October 30, 2019" instead of legally significant named entities such as "Nissin Kogyo Co., Ltd." This suggests challenges in grounding named entities and legal roles within the document context. Conversely, clause types like *Governing Law* and *No-Solicit of Employees* showed higher accuracy (70–78%) due to their formulaic and consistent language.

These findings echo limitations reported in prior work (Chalkidis et al., 2021; Sivapiran et al., 2023), where models struggle with ambiguous supervision signals and structurally diverse clauses. They also revealed a key insight: even when the model could correctly classify that a relevant answer existed, its generative output often failed to recover the correct clause. This motivated our exploration of hybrid approaches combining classification and generation and emphasized the need for robust document-level aggregation and clause-type-specific evaluation in subsequent iterations.

4.3. FLAN-T5 (Instruction-tuned generation)

After establishing the limitations of our T5 baseline—chiefly its tendency to default to "No answer" and its inability to model long-range dependencies—we moved to flan-t5-base, motivated by its instruction tuning and improved handling of generation-based tasks. We preserved the same chunk-question-answer format but added stronger prompting to guide clause-specific behavior, such as: *"Only extract the clause if it exists, otherwise respond with 'No answer.'"* Among several placeholder experiments for empty chunks, the string "No answer" produced the most stable performance; alternatives like blanks or varied strings led to poor generalization, erratic loss, and ultimately degraded evaluation consistency at the document level.

This model showed better clause fidelity and learned when to abstain more responsibly. Compared to the baseline, FLAN-T5 extracted longer and more structurally correct answers across clauses like *Renewal Term* and *Irrevocable License*. However, several persistent failure modes emerged. First, wrong anchoring: the model often returned semantically plausible but unrelated clauses—for instance, predicting IP assignment when asked about usage rights. Second, distractor content: when faced with entity-dense or verbose sections, it latched onto surface-level cues, generating outputs like "Deutsche Telekom" for unrelated prompts. Third, fragmentation: when references spanned multiple chunks, the model returned partial or noisy spans. In one case, it predicted a line on joint IP prosecution while the actual reference pertained to audit procedures in a separate chunk. Finally, hallucination under ambiguity remained a problem—e.g., for *Parties*, it sometimes fabricated names or defaulted to header text when the clause was absent.

Despite these challenges, FLAN-T5 demonstrated clear gains in clause-level awareness, abstention behavior, and structural fluency. These improvements reflect their alignment with instruction-guided tasks and underscore the potential of generation-based models for legal clause extraction—particularly when supported by effective chunking and prompting strategies. However, it continued to struggle with clause type disambiguation, especially when different clauses share overlapping legal phrasing (e.g., *Termination*, *Renewal*, *Expiration*), and with extracting answers that are implied through context rather than explicitly stated. Additionally, it lacked sufficient local reasoning across clause chunks, often missing connections that spanned multiple segments. These limitations are consistent with findings in prior legal NLP research (e.g., Zhong et al., 2020), which highlight the challenges of entity normalization and clause scope detection in complex contractual language.

4.3 BART Fusion (Contextual fusion + classification layer)

To address the persistent clause fragmentation observed in earlier models, we developed a custom fusion-based architecture built on top of BART. Unlike standard encoder-decoder models which process each chunk in isolation, this architecture fuses representations across chunks using a lightweight bidirectional LSTM layer. Each chunk was encoded independently with a prompt-question pair and passed through BART's encoder. While BART is not typically paired with prompts in the way FLAN-T5 is, prior work such as *BARTScore* (Yuan et al., 2021) and legal QA applications like *LexGLUE* suggest that instruction tuning and context fusion can be beneficial in structured legal tasks. We extracted the first token ([CLS]-like) representation from each chunk, concatenated them across the document, and used the LSTM to integrate contextual signals. This fused document-level representation was passed to both a generation head (to decode the clause) and a binary classification layer predicting whether a chunk contained an answer. We selected an LSTM over Transformer-based fusion due to its efficiency on small sequences (typically <10 chunks per document) and its ability to retain useful sequential signal without the added overhead. Transformer fusion was tested but yielded negligible gains.

Adding the classification layer proved useful in balancing precision and recall on "No answer" cases—helping the model abstain in low-relevance contexts. However, several new failure modes emerged. BART, not natively instruction-tuned like FLAN-T5, occasionally hallucinated plausible-sounding legal boilerplate even when "No answer" was correct. Despite using the same prompt format and data structure as FLAN, BART was less responsive to question specificity, sometimes defaulting to contract headers or repeating prior answers. It also struggled with long, multi-sentence clauses (e.g., *Parties*, *License Grant*), returning only partial spans. The fusion layer helped recover some fragmented answers (e.g., *Cap on Liability*, *Renewal Term*), but did not eliminate truncation entirely. Ultimately, the model was strongest on syntactically compact, semantically distinct clauses (e.g., *Agreement Date*, *Irrevocable License*) and weakest where answers were diffused, repetitive, or contextually entangled.

5. Results & Evaluation

We evaluate model performance at two levels: chunk-level, which measures how accurately the model identifies relevant content in individual text segments, and document-level, which assesses whether the model can correctly assemble a complete clause from multiple chunks. This is important because legal clauses often span multiple non-contiguous chunks. We report Exact Match, F1, Jaccard, BLEU, and ROUGE scores at both levels, and track correct/incorrect counts and clause-type accuracy, with a focus on document-level performance.

5.1 Chunk Level Evaluation

Chunk-level evaluation analyzes model predictions at the unit level (each segmented chunk of a document). Each chunk corresponds to a potential clause candidate—so high chunk accuracy means the model identifies and generates clauses precisely where they occur.

Chunk-Level Evaluation (All Models)

Model	EM	F1	Jaccard	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	Accuracy %
T5 Baseline	0.40	0.41	0.40	0.06	0.41	0.38	0.41	39.55%
Flan-T5 Base	0.78	0.82	0.81	0.47	0.82	0.68	0.82	78.46%
BART Fusion	0.58	0.86	0.84	0.46	0.86	0.84	0.85	57.73%

Chunk-Level Error Analysis

At the chunk level, the T5 baseline consistently underperformed across all evaluation metrics, particularly BLEU and exact match (EM), highlighting its difficulty in extracting precise clause spans with accurate phrasing. FLAN-T5 showed measurable improvement, especially in EM, reflecting stronger alignment with instruction-style inputs and greater caution—often abstaining from generating when uncertain rather than hallucinating content.

BART Fusion, while slightly lower in EM than FLAN-T5, achieved higher F1, Jaccard, and ROUGE scores. These gains point not to improved paraphrasing but to more accurate extraction of clause segments with stronger token-level overlap and fewer false positives. The added classification layer likely contributed by anchoring the generative process in more reliable chunk-level evidence. T5 and FLAN-T5 commonly misidentified clauses with similar surface forms (e.g., *Termination* vs. *Expiration*) or struggled with cases where clauses were implied but not explicitly stated. In contrast, BART Fusion successfully reduced both omission and misclassification, particularly in clauses like *Audit Rights*, *Revenue Sharing*, and *Ownership Assignment*, which had previously posed challenges.

5.2 Document-Level Evaluation

Document-level evaluation reflects how accurately the model reconstructs complete clauses from multiple chunks within a document. Document Accuracy measures the proportion of documents where the reconstructed answer exactly matches the reference. This is stricter than chunk accuracy and better reflects model reliability for contract-level extraction.

We evaluated document-level performance using three aggregation approaches:

1. **All Answers Aggregated**: Natively concatenates all chunks. No single answer, used as a baseline comparison.
2. **Best Answer via ROUGE-L:** Selects the highest quality chunk by highest ROUGE-L score against the reference.
3. **Exact Match → Fallback:** Uses exact match to select chunk otherwise defaults to ROUGE-L.

Document-Level Aggregation Summary (All Models)

Model	Aggregation Method	EM	F1	Jaccard	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	Doc Accuracy (%)
T5 Baseline	All Spans Aggregated	0.06	0.28	0.22	0.04	0.28	0.2	0.26	5.74%
	Best Span (ROUGE-L)	0.26	0.28	0.27	0.03	0.28	0.26	0.28	25.61%
	EM priority or ROUGE-L	0.34	0.35	0.35	0.05	0.35	0.32	0.35	34.22%
Flan-T5	All Spans Aggregated	0.26	0.68	0.60	0.51	0.69	0.58	0.63	25.86%
	Best Span (ROUGE-L)	0.46	0.55	0.52	0.34	0.56	0.46	0.54	46.12%
	EM priority or ROUGE-L	0.78	0.79	0.79	0.63	0.79	0.66	0.79	78.02%
BART Fusion	All Spans Aggregated	0.17	0.71	0.64	0.38	0.72	0.65	0.62	17.41%
	Best Span (ROUGE-L)	0.36	0.57	0.54	0.22	0.58	0.52	0.56	35.88%
	EM priority or ROUGE-L	0.58	0.63	0.63	0.11	0.64	0.62	0.63	57.96%

The T5 baseline exhibited weak document-level performance, with consistently low exact match (EM) and document accuracy. Even when aggregating across chunks, its extracted spans lacked the precision and completeness necessary to reconstruct accurate clause sets. FLAN-T5, by contrast, achieved notable gains when paired with a fallback mechanism, underscoring the benefit of retaining high-precision chunks while leveraging a secondary scoring method for coverage. BART Fusion, though slightly trailing FLAN-T5 in EM, demonstrated higher clause recall and reduced overgeneration. Its fusion strategy helped maintain strong F1 and Jaccard scores, even in documents with mixed-quality extractions, by filtering redundant or noisy spans while preserving relevant clause content.

Analysis of clause-level performance revealed that structural clarity, not frequency, was a stronger predictor of extraction accuracy. For instance, despite having only eight examples, “Most Favored Nation” clauses achieved 75% accuracy, while the much more frequent “Parties” clause scored only 8.3%—likely due to its inconsistent span length and noise.

6. Conclusion

While Flan-T5 was the most reliable model for exact clause extraction, consistently outperforming others in chunk-level exact match and generation fluency, BART Fusion brought key strengths that made it a valuable addition. Flan-T5 handled long-form generation with greater consistency, producing structurally complete and well-aligned clauses—especially for types like *Renewal Term*, *Change of Control*, and

Effective Date. In contrast, BART Fusion occasionally truncated outputs or repeated phrases, limiting its reliability for open-text clause spans.

However, BART Fusion excelled at detecting clause presence, particularly for edge cases where no answer was appropriate (e.g., *Liquidated Damages*, *Termination for Convenience*). It also handled scattered or partial clauses better—for example, recovering fragments of *Post-Termination Services*, *Revenue Sharing*, and *Ownership Assignment* across chunks. These cases show BART's classification layer enhanced clause localization even when generation struggled. Together, the findings support using Flan-T5 for precise clause generation, with BART Fusion as a strong candidate for hybrid setups where classification confidence guides when and what to generate.

References

- Almeida, M., Martins, A. F. T., & Coheur, L. (2020). Legal named entity recognition with distant supervision. *Proceedings of the 17th International Conference on Artificial Intelligence and Law (ICAIL 2020)*, 31–40. https://doi.org/10.1145/3390454.3390646

- Chalkidis, I., Androutsopoulos, I., & Michos, A. (2017). Extracting contract elements. *Proceedings of the 16th International Conference on Artificial Intelligence and Law*, 19–28. https://doi.org/10.1145/3086512.3086517

- Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2021). Legal-BERT: The muppets straight out of law school. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2898–2904. https://doi.org/10.18653/v1/2020.findings-emnlp.261

- Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2021). CUAD: An expert-annotated NLP dataset for legal contract review. *arXiv preprint arXiv:2103.06268*. https://arxiv.org/abs/2103.06268

- Leivaditi, E., Koutraki, M., & Koutrika, G. (2020). ContractNER: Contract clause and named entity recognition. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '20)*, 2969–2976. https://doi.org/10.1145/3340531.3411982

- Rajpurkar, P., Jia, R., & Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 784–789. https://doi.org/10.18653/v1/P18-2124

- Sivapiran, S., Lawrie, D., & Moradshahi, M. (2023). Party extraction from legal contracts using contextualized span representations. *arXiv preprint arXiv:2302.03817*. https://arxiv.org/abs/2302.03817

- Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *Advances in Neural Information Processing Systems*, 32, 3266–3280. https://papers.nips.cc/paper_files/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf

- Yuan, W., Neubig, G., & Liu, P. (2021). BARTScore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34, 27263–27277. https://arxiv.org/abs/2106.11520

- Zhong, Z., Xiao, C., Chaudhury, S., & Wang, D. (2020). A closer look at few-shot classification for legal clause detection. *Proceedings of the Natural Legal Language Processing Workshop at EMNLP 2020*, 13–22. https://arxiv.org/abs/2010.12784