Contract Clause Classification Using LegalBERT

Overview

This project addresses the problem of automated contract clause classification using modern legal-domain language models. It focuses on both **single-label** and **multi-label** clause classification using pretrained transformer-based models fine-tuned on the **CUAD (Contract Understanding Atticus Dataset)**. The system takes real-world legal contracts in .pdf or .docx format and outputs structured clause-level labels with high accuracy and interpretability.

System Architecture

a) Data Processing and Preparation

The dataset used was the **CUADv1** JSON, which includes thousands of question-answer pairs across legal documents. Two different representations of this dataset were constructed:

- **Single-label Format**: Clauses were filtered to retain only one high-confidence label per clause (top 10 most frequent labels) and downsampled to ensure balance in dataset. This supported classic classification using CrossEntropyLoss.
- **Multi-label Format**: Clauses were grouped by overlapping labels across questions. This enabled more accurate training of real-world contract clauses that often fall under multiple categories. Processed using binary label vectors and BCEWithLogitsLoss.

All inconsistencies and metadata lines were filtered. Label mappings were normalized and the dataset was split into train-validation sets with stratification based on label presence.

b) Model Design and Fine-Tuning

Both models used the **nlpaueb/legal-bert-base-uncased** pretrained transformer as the encoder.

Model	Architecture	Loss Function	Output Format
Single-Label	BERT + Linear (Softmax)	CrossEntropyLoss	One predicted label
Multi-Label	BERT + Linear (Sigmoid)	BCEWithLogitsLoss	Vector of label probabilities (0-1)

The **single-label model** was trained using the Hugging Face Trainer API with early stopping, LR scheduling, mixed-precision, and token-aware collators. The **multi-label model** was implemented via a custom PyTorch loop for finer control over logits, tokenization, and loss handling.

c) User Interface and Deployment

The application was deployed using **Gradio** with the following core functionalities:

- Upload .pdf or .docx contracts
- Clause segmentation (based on line boundaries)
- Classification per clause (real-time)
- Interactive explainability and Clause similarity

- CSV download of predictions for auditing or review

Evaluation Results

a) Single-Label Model

- **Accuracy:** 91%
- **F1-Score (macro):** 0.91
- **Misclassifications**: Mainly occurred where clauses contained overlapping semantics (e.g., exclusivity + license grant).
- **High-Performing Labels:**
    - Governing law (F1: 0.98), Cap on liability (F1: 0.96), Audit rights (F1: 0.93)
- **Low-Performing Label:**
    - Exclusivity (confusion with license grant due to semantic overlap)

b) Multi-Label Model

- **Micro F1-Score:** 0.96
- **Macro F1-Score:** 0.96
- **Sample-Averaged F1:** 0.96
- **Label Threshold:** 0.5 (tunable)
- **Observation**: When trained on multi-label annotated data, the model predicted overlapping labels accurately in >90% of synthetic and real-world cases.
- **Challenge**: Tended to miss secondary labels when one class dominated (e.g., a clause with 0.98 for one class but 0.48 for another below the threshold).

Conclusion

- The **single-label classifier** were cleaner and showed strong performance on well-separated legal clause categories, making it ideal for rapid deployment and strict classification pipelines.
- The **multi-label classifier**, although more sensitive to thresholding and class balance, excelled in capturing **real-world contract complexity and were more legally accurate** and informative for downstream tasks like contract review, auditing, and clause retrieval.

For precision-driven classification tasks, single-label models are sufficient. For nuanced legal clause understanding, multi-label classification offers far greater utility and realism.

.

The DemoClassifier Video shows working of both Multi and Single Classifiers respectively.