Report: Contract Document Classification Project

Introduction

This project explored the application of machine learning and large language models (LLMs) in automatically classifying legal contract documents into their respective categories. The focus was on five types of contracts that are common in professional and business settings: **Non-Disclosure Agreements (NDAs), Service-Level Agreements (SLAs), Employment Contracts, Vendor Agreements, and Partnership Agreements.**

The primary objective was to design a system that could take in either raw text or a PDF document (including scanned files) and accurately classify it into one of the five categories. This required careful attention to **data preparation, model selection, evaluation, and iterative experimentation**.

Data Preparation

The dataset was constructed from a combination of real and synthetic sources:

- **Manual Collection:** I manually searched online to find two authentic samples for each contract type.

- **Synthetic Augmentation:** To extend this very small dataset, I leveraged ChatGPT to generate synthetic documents, each 2–3 paragraphs long, based on the real examples. This helped diversify the training set while retaining structural and linguistic features of genuine contracts.

- **Formatting & Conversion:** Each contract type was saved as a PDF collection. I then wrote scripts to parse these PDFs into structured JSON files. Finally, I merged the JSONs into a single dataset that contained all the examples in a machine-friendly format.

This multi-step preparation ensured I had a usable dataset that, while limited in size, captured enough variety to support initial experimentation with text classification models.

Models and Rationale

1. TF-IDF + Logistic Regression

I began with a traditional machine learning baseline: a **TF-IDF (Term Frequency–Inverse Document Frequency) vectorizer** combined with a **Logistic Regression (LR) classifier**. The rationale was twofold:

- TF-IDF provides a lightweight yet effective way to represent textual data, especially when working with short documents.
- Logistic Regression is interpretable, efficient to train, and surprisingly powerful for small-scale text classification problems.

This combination gave me a strong starting point with minimal computational overhead.

2. Few-Shot Classification

To explore more modern approaches, I implemented **few-shot classification** using prompt-based techniques. This method allowed me to provide example contracts in the prompt and ask the model to generalize to new inputs. The benefit here was that it did not require extensive training, and it leveraged the ability of LLMs to adapt quickly with minimal supervision.

3. OpenAI Chat Models (LLM-based Classification)

Finally, I incorporated **OpenAI's GPT-based models** for classification. The motivation here was to test the state-of-the-art in natural language understanding. LLMs are particularly powerful when dealing with nuanced and context-rich documents like contracts, as they can pick up subtle cues in language beyond simple keyword frequency.

Results and Evaluation

The **TF-IDF + Logistic Regression model** performed exceptionally well on my small test set. Below is the classification report:

	precision	recall	f1-score	support
Employment	1.00	1.00	1.00	1
NDA	1.00	1.00	1.00	1
Partnership	1.00	1.00	1.00	1
Service	1.00	1.00	1.00	1

```
      Vendor       1.00      1.00      1.00         1

    accuracy                           1.00         5
   macro avg       1.00      1.00      1.00         5
weighted avg       1.00      1.00      1.00         5
```

While the results appear **perfect (100% across all metrics)**, it is important to note the dataset size was small. This means the model may not yet be robust in real-world scenarios with noisier, more varied contracts.

The few-shot and LLM-based approaches also produced **highly accurate classifications**, particularly when contracts were more verbose. However, they were more resource-intensive compared to the TF-IDF baseline.

Challenges and Mitigations

- **Model Selection:** I initially planned to fine-tune a BERT-based classifier. However, the computational requirements were too heavy for my resources. To address this, I pivoted to TF-IDF + LR as a simple and effective alternative, and supplemented it with few-shot and LLM-based approaches to still capture more advanced modeling techniques.

- **Zero-Shot Limitations:** I experimented with zero-shot classification, but results were inconsistent. This highlighted the importance of providing context (few-shot) or using a fine-tuned approach rather than relying on zero-shot.

- **Time Constraints:** The project coincided with the conclusion of my **National Service (internship)**, during which I was also tasked with other work responsibilities. As a result, I had to manage my time carefully. I dedicated about **7 days (~8 hours total)** to complete the project. By prioritizing lightweight models and automating parts of the data preparation, I was able to meet my goals within this timeline.

Future Improvements

Looking ahead, I see several opportunities to extend and strengthen this work:

- **LangSmith Evaluation Dashboard:** Implementing a LangSmith-powered evaluation dashboard would allow me to systematically monitor, compare, and analyze LLM performance over time. This would provide a more reliable way to catch drift, highlight

weak areas, and fine-tune prompts or models.

- **Few-Shot Evaluation Script:** I plan to write a dedicated evaluation script for the few-shot approach, enabling automated benchmarking against the TF-IDF baseline and LLM methods. This will bring more rigor to the comparison of classical and modern models.

- **Interactive Frontend:** To make the system more user-friendly, I could build a lightweight UI in **Streamlit** (for rapid prototyping) or even **ReactPy** (for a production-ready web interface). This would allow non-technical users to upload contracts and instantly view classification results in an accessible interface.

By adding these improvements, the project can evolve from a proof-of-concept into a **practical, production-ready tool** for real-world contract management.

Conclusion and Reflections

This project gave me the opportunity to combine **classical machine learning techniques** with **modern LLM-based methods** to tackle the problem of contract classification. The contrast between the efficiency of TF-IDF + LR and the adaptability of few-shot and GPT-based models was particularly illuminating.

Beyond the technical achievements, I genuinely enjoyed working on this project. It strengthened my interest in building **practical NLP applications** where data is limited but the need for automation is high. I see great potential in expanding this work by:

- Collecting a larger and more diverse dataset of contracts.

- Fine-tuning transformer-based models (BERT, RoBERTa) when computational resources allow.

Overall, this project reinforced my enthusiasm for developing intelligent systems that bring real-world value and made me eager to pursue similar challenges in the future.