**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. **Few-Shot + TF-IDF + Cosine Similarity + Keyword Matching (Hybrid Method)**

The first approach combined multiple lightweight techniques:

* + Few-shot examples with TF-IDF and cosine similarity: Compare the input text against pre-written sample texts for each contract type.
  + Keyword matching: Identify domain-specific keywords that frequently appear in each contract type.
  + Final prediction: A weighted combination of similarity (70%) and keyword scores (30%) was used to classify the document.

This hybrid method provided a balance between semantic similarity and keyword presence, making it robust for smaller datasets.

1. **TF-IDF + Logistic Regression**

As a strong traditional machine learning baseline, I implemented a TF-IDF (Term Frequency–Inverse Document Frequency) vectorizer combined with a Logistic Regression classifier:

* + TF-IDF provides an effective way to represent textual data.
  + Logistic Regression is interpretable, efficient, and works well even with limited data.

This gave me a solid benchmark for evaluating more advanced approaches.

1. **OpenAI Chat Models (LLM-based Classification)**

Finally, I tested OpenAI’s GPT-based models for classification. These LLMs excel in understanding context-rich and nuanced language, making them well-suited for contract classification tasks where subtle linguistic cues matter.

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 that the dataset size was very small, so real-world performance may differ.

The **hybrid few-shot + TF-IDF + keyword matching method** performed reliably on contracts with recognizable patterns, while the **LLM-based classification** excelled on longer, more complex contracts but was more resource-intensive.

Challenges and Mitigations

* **Model Selection**: I initially planned to fine-tune a BERT-based classifier but lacked the resources. I pivoted to lightweight models (TF-IDF + LR, hybrid method) and added LLMs for advanced capability.
* **Zero-Shot Limitations**: Pure zero-shot classification gave inconsistent results, reinforcing the importance of few-shot context.
* **Time Constraints**: With only ~5 days (~8 hours total) to complete this project, I optimized by automating data prep and focusing on efficient models.

Future Improvements

* **LangSmith Evaluation Dashboard**: For continuous monitoring and benchmarking of model performance.
* **Few-Shot Evaluation Script**: To systematically compare hybrid, TF-IDF + LR, and LLM approaches.
* **Interactive Frontend**: A UI (Streamlit or ReactPy) for user-friendly document uploads and instant classification.

Conclusion and Reflections

This project combined classical ML (TF-IDF + LR), a hybrid lightweight approach (few-shot + TF-IDF + keyword matching), and modern LLMs for contract classification. Each method had strengths and trade-offs, providing valuable insights into balancing efficiency, interpretability, and accuracy.

Beyond the technical results, I truly enjoyed working on this project. It reinforced my interest in NLP solutions for real-world tasks and made me eager to expand this work by collecting larger datasets, fine-tuning transformer models, and building production-ready tools.