Smart Legal Assistant for Contract Analysis and Risk Assessment (SLA-CARA)

Abstract

In today's legal operations, efficient clause-level analysis is crucial for contract review and risk assessment. This project presents a comprehensive system that allows users to upload full legal contracts (PDF/DOCX) or input individual clauses directly. Uploaded documents are first converted into text using document parsers, and subsequently segmented into logical paragraphs through natural language processing (NLP) chunking techniques. For each clause, our FAISS + Cosine Similarity hybrid system retrieves semantically similar clauses from the training set to serve as contextual references, while avoiding direct content injection into the prompt. Each extracted clause is then processed for multi-task analysis: (1) legal clause classification, (2) clause summarization (3) risk assessment and (4) Key legal Entities Extraction (And additional task we performed not originally mentioned in the proposal) using an advanced LLM, LLAMA 3.2 3B. The Performance evaluation of each of the tasks are done according to the task specific metrics and present in detail. Our system showed to outperform the traditional models, while being more cost effective, scalable and up to date context.

Introduction

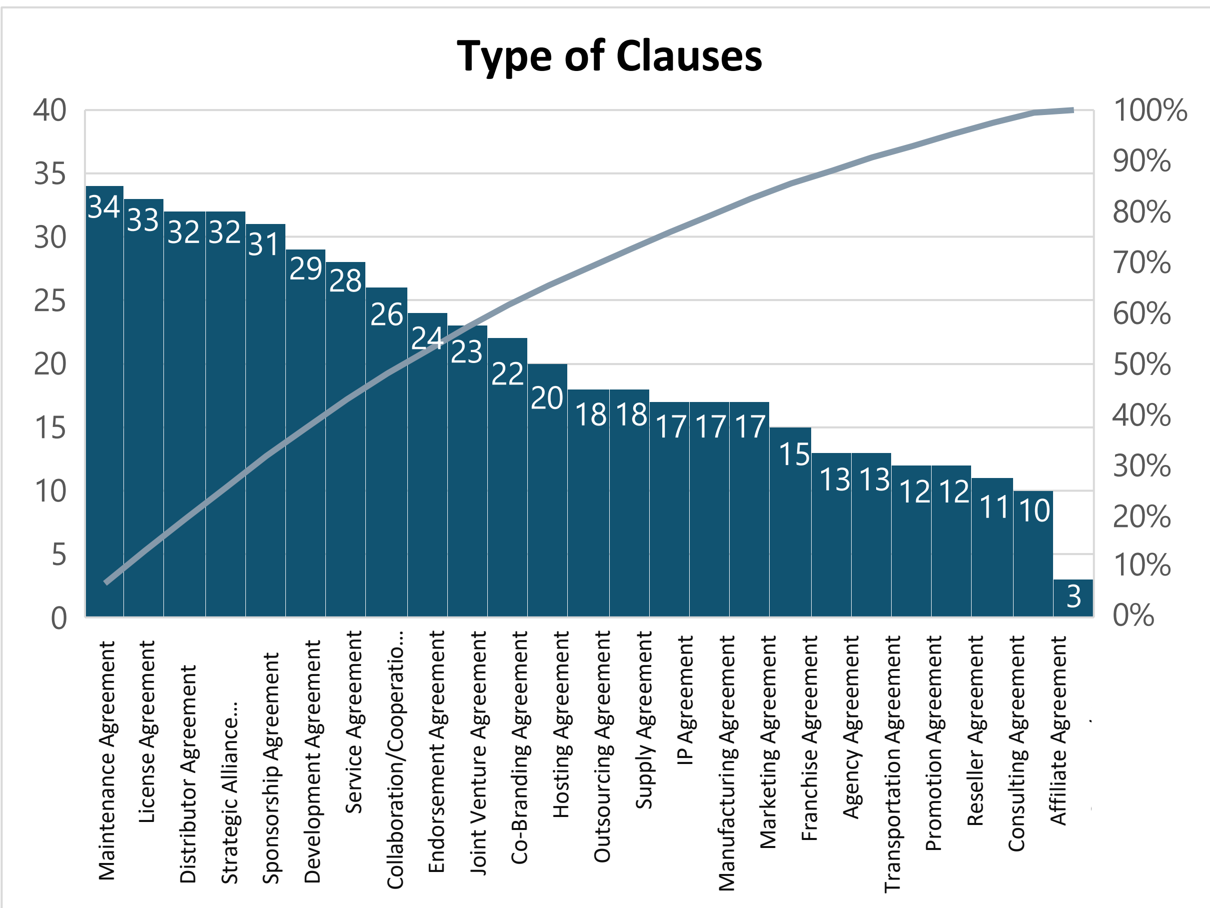
In the Legal world, the accurate analysis of legal contracts is crucial for efficient document review, risk assessment and summarization. Traditionally, people hire some legal experts to manually process the contracts which can be time-consuming, error-prone and costly. This is where the LLMs comes to play, these can help to automate and enhance the analysis while maintaining high standards of precision and reliability [1]. But the problem with the traditional method of fine-tuning a LLM model is that it can tend to be time consuming, require computation power, and labeling. Also, Traditional methods are not scalable, secure or up to Date and not suitable for legal enterprise use cases. This is exactly where a new AI technology like RAG can help us [2].

Retrieval Augmented Generation (RAG) is an innovative approach proposed by Meta that enhances LLMs by integrating a powerful information retrieval system. In this mechanism, it pulls out relevant external data, ensuring that AI-generated responses are contextually rich and accurate. RAG **i**s generally better for most enterprise use cases because it is more secure, scalable, and cost-efficient. It allows for enhanced security and data privacy, reduces compute resource costs, and provides trustworthy results by pulling from the latest curated datasets. RAG can optimize legal processes and provide unparalleled support to legal professionals [3].

The retrieval component serves as the backbone of RAG systems, responsible for sourcing relevant information in response to user queries. In our case we used the FAISS + Cosine Similarity hybrid which was powerful in retrieval task. We use this hybrid to utilize the faster retrieval capability of FAISS system and precision of the Cosine Similarity. And Employing the Legal-BERT for tokenization and embedding generation can offer legal specific insights, increased accuracy and compatibility with our project. LLAMA 3.2 3B has been used with our project as it can offer ideal trade-off between efficiency, computational cost and inference speed, it is light weighted enough to deploy within our pipeline and customizable. By leveraging these advanced techniques and prompting strategies, we aim to generate concise and accurate contract analysis report.

Dataset Overview

In this project, we used the Contract Understanding Atticus Dataset (CUAD) v1 [4], prepared by The Atticus Project, Inc. CUAD v1 consists of over 13,000 manually labeled clauses across 510 commercial legal contracts, identifying 41 important legal clause categories (In CSV) and reviewed by attorneys during corporate transactions such as mergers and acquisitions, investments, and IPOs. It has 26 different types of clause level categories such as Termination, Governing Law, Indemnification, Liability, and Confidentiality etc. to perform legal clause classification. The files in CUAD v1 include 1 CSV file, 1 SQuAD-style JSON file, 28 Excel files, 510 PDF files, and 510 TXT files. The Dataset contains legal experts annotated labels and summaries crafted for finetuning LLMs but was repurposed by us to be used in the RAG system.



This combined Json file will be the Master Dataset from which the Context will retrieved for the LLM response by our Hybrid system.

Methodology

1. Data Preparation and Preprocessing:

I. Although the dataset was preprocessed and cleaned by the Atticus Project team, it was not intended for the RAG project but prepared for finetuning LLM for classification task. So, needs some processing to be done on it.
II. For our RAG feature, first the CSV files with QAs are converted to Json format as it is really require for our Legal Entity extraction task and merged it with the provided master Json file.
III. Master Json is then checked for the ground truths and context clauses. Only QA pairs containing a text field were used as true summary references.
IV. Then each clause is pattern-matched with patterns given in table below to tag contract clauses with respective potential risk categories, severity and weight. And it is saved as Risk Summarization.
V. Now that Dataset is ready, we can move onto to the next Step of the project.

Risk Category	**Patterns**	**Severity**	**Weight**
Termination Risk	terminate, termination for convenience, without cause, termination date	High	3
Confidentiality Risk	confidential, non-disclosure, keep information private	Medium	2
Payment Risk	payment, invoice, fee, compensation	Medium	2
Liability Risk	liability, liable, indemnify, damages	High	3
IP Risk	intellectual property, ownership, copyright, assigns to the recipient	Medium	2
Change of Control Risk	change of control, merger, assignment by operation of law	Low	1
No Risk	-	Low	1

2. Tokenization and Embedding Creation:

I. Now that the dataset Json file is ready, next would-be tokenization of each clause with its QA content from the dataset.
II. We are using "nlpaueb/legal-bert-base-uncased" using Hugging Face's 'AutoTokenizer' to tokenize each clause content. This breaks complex legal text into model-ready tokens while preserving contextual meaning.
III. Once tokenized, the text is passed through an embedding model to create dense vector representations. We are using the 'AutoModel' from Hugging Face to load the Legal-BERT embedding model.
IV. we apply mean pooling over the hidden states to obtain a fixed-size representation for each input text. This vector captures the semantic meaning of that clause.
V. These embeddings were further normalized using L2 normalization to ensure uniformity of vector magnitude, which is especially important for similarity-based retrieval tasks.

3. Vector Database Creation and Retrieval System:

I. With the Embeddings created, we need to store the clause embeddings in a vector embedding and design a retrieval system [3] to find the clauses which are semantically similar clauses.
II. We are using the FAISS (Facebook AI Similarity Search) to build the vector database for fast and scalable nearest-neighbor search. All clauses' embeddings are inserted into FAISS index
III. Metadata such as clause ID, title, and associated information are separately stored in a mapping file (JSON), maintaining a reference to the raw clause text.
IV. Now that Vector database is ready, we need to create a retrieval system. We are first using the FAISS search to retrieve top-k^2 relevant clauses.
V. Then these top-k^2 relevant clauses are further filtered or reranked into top most relevant context by the Cosine similarity. The retrieved clause is ready to be passed with prompt to the LLM model.

4. Prompting the LLM for Specific Legal Tasks:

I. Now it is time to build a prompt tailored to each of the task we discussed before. This step is one of the most important steps as we need to guide the Large Language Model (LLM) toward precise outputs without any deviation or hallucinations and clear task boundaries.
II. In this step, our pipeline sends these custom-crafted prompts to LLaMA 3.2 3B API Client offered by Lambda Labs API to perform specific legal NLP tasks on each contract clause.
III. The prompts contain both the input clause and the semantically similar retrieved context to control the response quality.
IV. Each legal task uses a dedicated prompt template with a clear structure, reference context and expected response format in Json as shown below:

Task	Prompt Content	Prompt Context
Clause Classification	Classifies a clause into one of 25 predefined contract categories (e.g., License Agreement, NDA, Collaboration Agreement).	Clause text + relevant **reference titles**
Risk Assessment	Assigns a legal **risk category** (e.g., Liability Risk, IP Risk) and a **Risk Score** (scale of 1–3).	Risk summaries from **similar clauses**
Clause Summarization	Summarizes the clause in 1–2 sentences with emphasis on legal clarity and contractual intent.	Tailored instructions based on **clause type** (e.g., Termination, Confidentiality, IP)
Key Legal Insight Extraction	Extracts key legal details from the clause, including: – Entities – Dates – Obligations – Parties	Clause text + relevant **reference Q&A pairs**

SLA-CARA Pipeline Overview:

The SLA-CARA pipeline is designed to automate the processing, understanding, and analysis of legal documents. It consists of the following sequential stages:

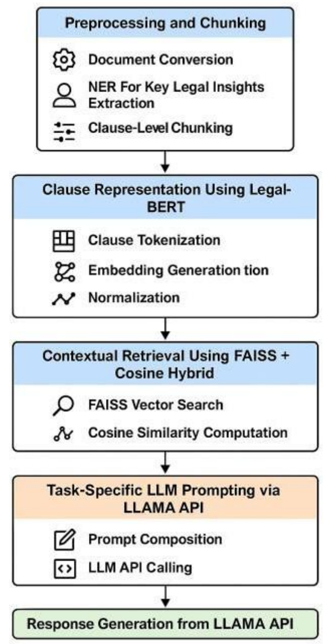
1. Preprocessing and Chunking:

Document Conversion: If the pipeline is served with Contract files (PDF, DOCX formats) then first converted into raw text. Or else if it is served with single clause then it is sent for further processing.

NER For Key Legal Insights Extraction:

Then we perform Named Entity Recognition (NER) using the Spacy "en_core_web_md" on the raw data and use it for the Key Legal Insights Extraction Task.

Clause-Level Chunking: If, it is a contract, Then the extracted text is segmented into paragraph-based clauses, ensuring that each unit captures a coherent legal concept for downstream analysis. Here we are using Regex and Spacy "en_core_web_md"[7] to perform chunking by looking for numbered clause patterns or newline breaks intelligently.



Clause Cleaning: The text is then processed for cleaning the special characters or symbols which are not relevant.

2. **Clause Representation Using Legal-BERT:**

   Clause Tokenization: Each chunked clause is tokenized using "nlpaueb/legal-bert-base-uncased"[6] using Hugging Face's 'AutoTokenizer' for the vector embedding creation.

   Embedding Generation: Then the tokenized clause is then converted into vector embeddings using Pre-trained Legal-BERT.

   Normalization: Embeddings are normalized to maintain vector uniformity and enhance retrieval performance.

3. **Contextual Retrieval Using FAISS + Cosine Hybrid:**

   FAISS Vector Search: A FAISS-based vector database is utilized to perform top-k^2 nearest neighbor searches for each clause.

   Cosine Similarity Computation: Then these top-k^2 relevant clauses are further filtered or reranked into top most relevant context by the Cosine similarity. The retrieved clause is ready to be passed with prompt to the LLM model.

4. **Task-Specific LLM Prompting via LLAMA API:**

   Prompt Composition: A custom prompt is composed with task requirements, response instructions, input clause, relevant context and response format.

   LLM API Calling: The prompt is then sent to LLAMA 3.2 3B Instruct hosted via the Lambda API [5] to generate task-specific responses.

5. **Response Generation from LLAMA API:**

   Finally, we get response tailored to individual tasks which gets processed and shown in the UI part of the pipeline. The LLaMA model handles various downstream tasks, including:

   - Clause Classification
   - Plain Language Summarization
   - Risk Labeling and Analysis
   - Key Legal Insights Extraction

Results with Interpretation Discussions:

In our project, we have four different legal specific tasks, and the evaluation has been on each task. And the response of the model pipeline will also be shown in this section:

1. **Evaluation of the Clause Retrieval Hybrid System:**

   We have performed Recall@k metric evaluation on our Hybrid model which shows that how often the ground truth clause appears in the top k results retrieved. It **focus on Relevance, Not Ranking;** In legal NLP, we care **whether the right clause is found**, not where it ranks. We were able to achieve a strong retrieval score of .77 on the top 5 recalls especially in legal text retrieval context.

   Interpretation of results:

   A 77.12% recall@5 is quite strong, especially in legal text retrieval, where clauses can be phrased very differently but mean the same thing. It implies that our Clause Retrieval Hybrid System is doing a solid job of retrieving contextually relevant clauses.

2. **Evaluation of Clause Classification Task:**

   For the clause classification, we have calculated the F1-Score, Precision, recall metrics on each of the class category. **Accuracy** can be **misleading** if your dataset is imbalanced which is the case with our dataset. The Precision measures how accurately your model assigns the right clause category. While Recall measures how *completely* your model catches all instances of a category. Finally, F1-Score is the **harmonic mean** of Precision and Recall. We were able to achieve an overall accuracy of 66% and rest of the results are shown below.

📊 Classification Report:

	precision	recall	f1-score	support
AFFILIATE AGREEMENT	1.00	0.55	0.71	33
AGENCY AGREEMENT	0.49	0.52	0.51	42
CO-BRANDING AGREEMENT	0.85	0.83	0.84	121
COLLABORATION/COOPERATION AGREEMENT	0.00	0.00	0.00	0
CONSTRUCTION AND MAINTENANCE AGREEMENT	0.00	0.00	0.00	0
CONSULTING AGREEMENT	0.78	0.72	0.75	25
DEVELOPMENT AGREEMENT	0.82	0.58	0.68	139
DISTRIBUTOR AGREEMENT	0.72	0.84	0.78	120
ENDORSEMENT AGREEMENT	0.55	0.77	0.64	65
FRANCHISE AGREEMENT	0.87	0.93	0.90	112
HOSTING AGREEMENT	0.76	0.33	0.46	49
INTELLECTUAL PROPERTY AGREEMENT	0.00	0.00	0.00	0
IP AGREEMENT	0.00	0.00	0.00	61
JOINT VENTURE AGREEMENT	0.79	1.00	0.88	23
LICENSE AGREEMENT	0.49	0.81	0.61	114
MAINTENANCE AGREEMENT	0.92	0.64	0.75	119
MANUFACTURING AGREEMENT	0.81	0.72	0.76	58
MARKETING AGREEMENT	0.69	0.27	0.39	67
NON-COMPETE / NO-SOLICIT / NON-DISPARAGEMENT AGREEMENT	0.00	0.00	0.00	0
OUTSOURCING AGREEMENT	0.62	0.82	0.71	44
PROMOTION AGREEMENT	0.72	0.69	0.70	48
RESELLER AGREEMENT	0.61	0.39	0.47	36
SERVICE AGREEMENT	0.62	0.42	0.50	31
SPONSORSHIP AGREEMENT	0.00	0.00	0.00	0
STRATEGIC ALLIANCE AGREEMENT	0.73	0.80	0.76	107
SUPPLY AGREEMENT	0.74	0.55	0.63	98
TRANSPORTATION AGREEMENT	0.86	0.75	0.80	8
accuracy			0.66	1520
macro avg	0.57	0.52	0.53	1520
weighted avg	0.71	0.66	0.67	1520

Interpretation of results:

This classification report shows moderate overall performance with a weighted F1-score of 0.67, accuracy of 0.66, and macro F1 of 0.53, it shows that the model is doing well across the categories which are common like Franchise, Co-Branding, License, Development. But it really struggles with rare or unseen classes like Intellectual Property, Construction, Sponsorship where the precision and recall are both 0.00. This is common in RAG system as the classes which are rare, and unseen tend to be retrieved less and generalizes poorly on unseen data. So, performance is good on frequently retrieved or well-represented categories.

3. **Evaluation of Risk Assessment Task:**

Like the clause classification evaluation, we have performed the standard F1-Score, Precision, recall metrics on each of the Risk clause category. It is being shown in the below classification report.

	precision	recall	f1-score	support
Termination Risk	1.00	0.89	0.94	18
Confidentiality Risk	1.00	1.00	1.00	10
Payment Risk	1.00	0.47	0.64	19
Liability Risk	0.87	1.00	0.93	20
IP Risk	0.59	1.00	0.74	10
No Risk	0.00	0.00	0.00	10
micro avg	0.72	0.75	0.73	87
macro avg	0.74	0.73	0.71	87
weighted avg	0.81	0.75	0.75	87
samples avg	0.67	0.67	0.67	87

Interpretation of results:

The model is great at catching Termination & Confidentiality Risk classes, it rarely makes mistakes and finds almost all of them. Model missed a few actual payment risks, probably due to varied wording. The model catches all liability-related clauses but sometimes tags others incorrectly and it Likely because common words like “damages” appear in many places. The model couldn’t detect safe clauses at all, possibly because there weren’t enough examples in the training data, or they weren’t distinct enough.

4. **Evaluation of Legal Clause Summarization Task:**

For the Legal Clause Summarization task, we have performed multiple metrics, and these will be discussed below:

a) **ROUGE-1 (Score: 0.5015):**

This measures the overlap of individual words called unigrams between the generated summary and the reference summary.

Interpretation: We have achieved a score of 0.5015 while testing the model. The Score Indicates that good unigram overlaps between generated and reference summaries, suggesting decent content coverage.

b) ROUGE-2 (Score: 0.3251):

This goes one level higher—it checks for overlap of two-word phrases or called bigrams. A solid score here means our summaries are not just word salads, rather they preserve meaningful phrase structures.

Interpretation: We have achieved a score of 0. 3251 while testing the model. It captures bigram overlap; shows that the summaries preserve some local phrase structure, avoiding word salad issues.

c) ROUGE-L (Score: 0.4569):

This looks at the longest matching sequence of words between the two texts. It's useful for understanding whether the overall structure and flow are being maintained.

Interpretation: On testing data, the model was able to score of 0. 4569. The score suggests the generated summaries retain logical sequence and fluency.

d) ROUGE-Lsum (Score: 0.4569):

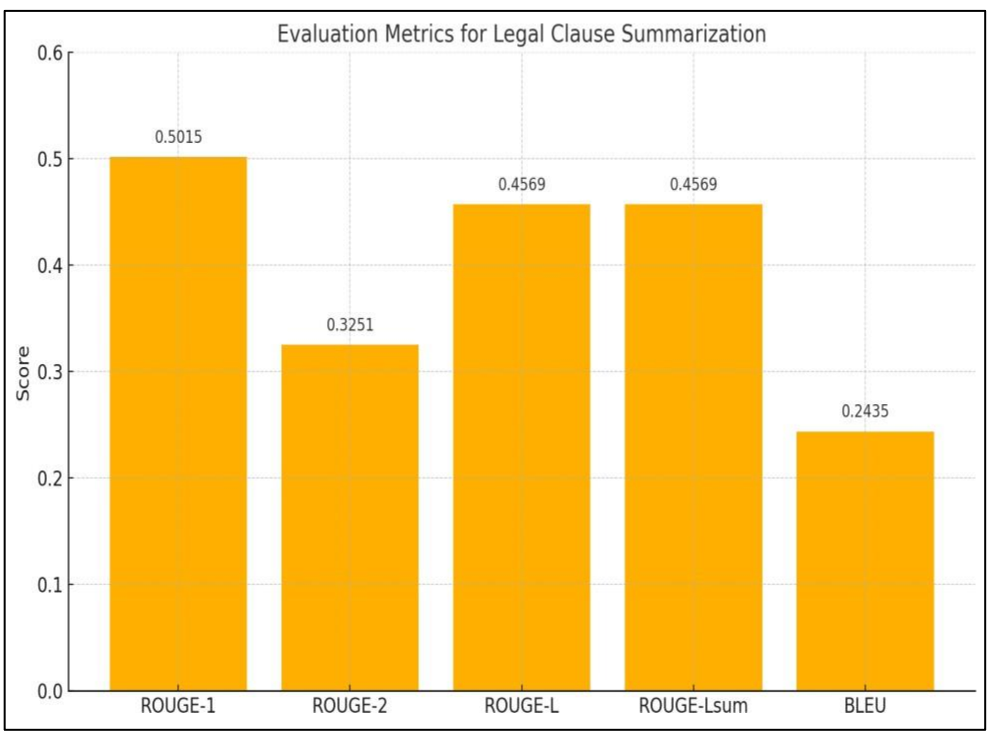
While ROUGE-L works at single sentence level, the ROUGE-Lsum is specially designed for multi-sentence summaries and measures sequence at paragraph/summary level.

Interpretation: The Score confirms that model is designed for multi-sentence summaries; confirms structural and flow at the paragraph level.

e) BLEU (Score: 0.2435):

BLEU measures the precision of n-grams, penalizing overly short summaries. BLEU is very strict metric, and even good summaries usually don't get very high BLEU scores.

Interpretation: A score above 0.2 is considered reasonable in summarization tasks, the moderate score reflects some phrase alignment but penalizes briefness.



5. SLA-CARA Pipeline Evaluation:

The Pipeline has two possible ways of getting the output, the detailed output will be shown below:

1. Single Clause Analysis with CARA:

Here we passed the clause

```
"This Agreement is made and entered into on April 1, 2025, by and between Acme
Corporation ("Company"), a Delaware corporation, and Beta Solutions LLC
("Contractor"), a Texas limited liability company.
2. Confidentiality
Each party agrees to keep confidential all non-public information disclosed by the
other party during the term of this Agreement and to use such information solely for
the purposes of performing under this Agreement. This clause shall survive the
termination of this Agreement for a period of three (3) years."
```

and was able to get the results in the table below

<u>Output Response:</u>

--- Classification ---

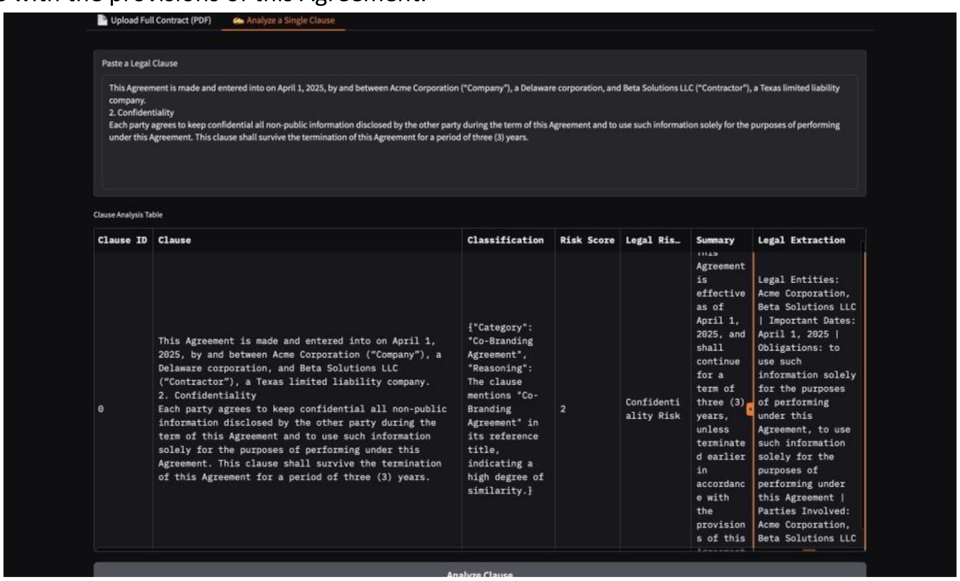
- Category: Co-Branding Agreement
- Reasoning: The clause mentions "Co-Branding Agreement" in the reference title, indicating a high similarity.

--- Risk Analysis ---

{'Risk Score': 2, 'Legal Risks': ['Confidentiality Risk']}

--- Summarization ---

The agreement commences on April 1, 2025, and will continue for a term of three years, unless terminated earlier in accordance with the provisions of this Agreement.



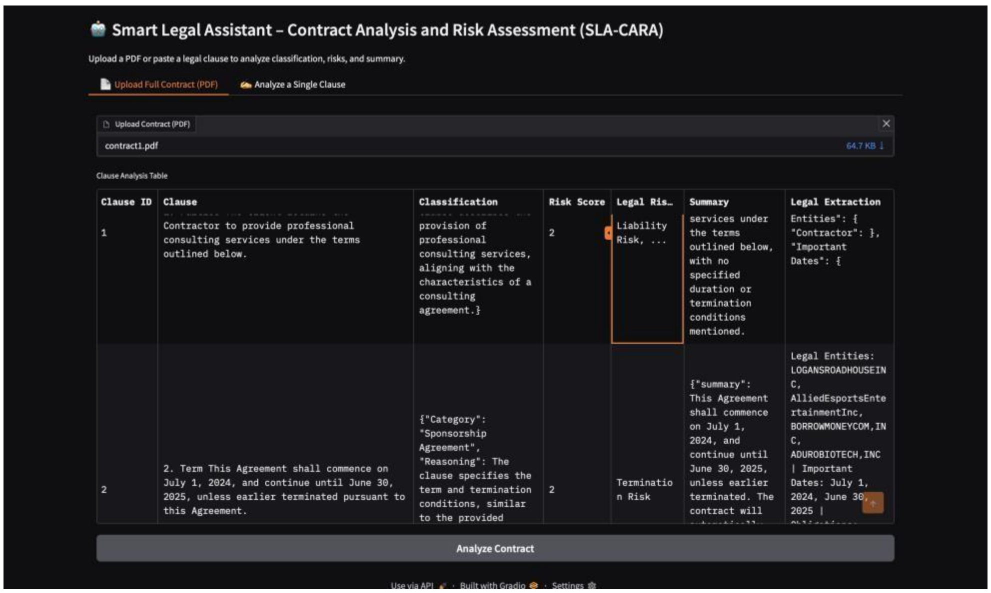
--- Key Legal Extraction ---

- **Key Legal Entities**:
  - Acme Corporation
  - Beta Solutions LLC
- **Important Dates**:
  - April 1, 2025
- **Obligations**:

- Each party agrees to keep confidential all non-public information disclosed by the other party during the term of this Agreement.
  - Each party agrees to use such information solely for the purposes of performing under this Agreement.
- **Parties Involved**:
  - Acme Corporation
  - Beta Solutions LLC

2. Full Contract Analysis with CARA:

Here we passed the contract “contract1.pdf” and was able to get the output for all the clauses from the contract. It is a list of all the response displayed in a table.



<u>**Response Output:**</u>

Chunk 1: SERVICE AGREEMENT

This Service Agreement ("Agreement") is made effective as of July 1, 2024, by and between:

- Client: AlphaTech Solutions Inc., a Delaware corporation with offices at 123 Innovation Drive, Wilmington, DE 19801.
- Contractor: NovaBridge Consulting LLC, a New York limited liability company with offices at 789 Enterprise Lane, New York, NY 10001.

In consideration of the mutual promises and covenants herein, the parties agree as follows:

--- Classification ---

- Category: Service Agreement
- Reasoning: The clause contains the exact title "SERVICE AGREEMENT" and has similarities to other contract clauses titled "UNDR AGMT-SERVICES AGREEMENT".

--- Risk Analysis ---

{'Risk Score': 0, 'Legal Risks': ['No Risk']}

--- Summarization ---

This Service Agreement is effective as of July 1, 2024, between AlphaTech Solutions Inc. and NovaBridge Consulting LLC, outlining the terms and conditions of their mutual promises and covenants.

--- Key Legal Extraction ---

- **Key Legal Entities**:
  - AlphaTech Solutions Inc.
  - NovaBridge Consulting LLC
- **Important Dates**:
  - July 1, 2024
- **Obligations**:
  - In consideration of the mutual promises and covenants herein, the parties agree as follows:
- **Parties Involved**:
  - The Client
  - The Contractor

……

The above response is only for one clause, content is too long to be included here as there are a lot of chunks.

Discussions:

Across all the evaluation and results, we can see that metrics indicate that our legal analysis system can handle complex clause processing and legal analysis. Although some areas need some improvement which was an expected behavior as we are using a dataset which was tailored to be used for LLM finetuning. The retrieval system can be reliably used to provide reference context for legal tasks prompting. The clause classification system has shown moderate performance, performing good with common classes and while rare classes really struggled. This is likely due to class imbalance and lack of training examples for those categories. For the Risk Assessment task, the model really performed well with high-risk clauses but struggles with safe or ambiguous clauses. And with Clause Summarization Task, All the scores indicate that summaries had a good lexical overlap and structure preservation similar reference context.

Challenges we faced:

1. Limited Computing Resources - LLaMA and Legal-BERT models required GPU memory and runtime beyond standard local systems
2. Rule Creation for Risk Assessment – We had to really go through some websites and use online tools for coming up with the rules we used.
3. Dataset Not Tailored for Inference - The CUAD dataset was mainly created for fine-tuning models, not for plug-and-play inference.
4. Prompt Engineering Difficulties - Crafting task-specific prompts that produce accurate and structured responses was time-consuming.

Future Enhancements:

1. Multilingual Legal Document Support – we can extend the project to include text translation and provide multi language support.
2. We can expand Dataset with Expert-Annotated from SEC Files and EDGAR Contracts. Increase the number of different contracts in each clause type.
3. We can perform Legal Compliance Engine Integration to compare our clauses against jurisdictional compliance requirements.

Conclusion:

Our SLA-CARA project successfully demonstrates the power of combining retrieval-augmented generation (RAG) with large language models to automate complex legal contract analysis tasks. Through the FIASS + Cosine similarity hybrid retrieval system, we can retrieve task specific context for each clause and supply it to the LLM as part of the custom tuned prompt to enable accurate classification, summarization, risk assessment, and entity extraction.

The pipeline performed strong with Context retrieval (Recall@5 = 0.77), solid F1 Score in clause classification for common clause categories and robust ROUGE and BLEU scores in summarization task. While the performance really dropped in rare or low risk clauses, and this is expected with the nature of the CUAD dataset.

The project is scalable, modular and efficient for real-world contract parsing and analysis. Despite the challenges we discussed, the SLA-CARA has shown true potential and provides a foundational Legal RAG baseline for the future improvement and enhancements.

References:

[1] Callidus Staff, "LLMs in Legal Practice: A Game Changer for Law Firms," 25 04 2025. [Online]. Available: https://callidusai.com/blog/the-rise-of-llms-in-legal-practice-a-game-changer-for-law-firms/.

[2] Lindsay MacDonald, "RAG vs Fine Tuning: How to Choose the Right Method," 31 07 2024. [Online].

Available: https://www.montecarlodata.com/blog-rag-vs-fine-tuning/#:~:text=RAG%20involves%20augmenting%20an%20LLM,i

ts%20performance%20on%20specific%20tasks..

[3] Arunim Samat, 23 03 2024. [Online]. Available: https://www.truelaw.ai/blog/legal-rag-vs-rag-a-technical-exploration-of-retrieval-systems.

[4] The Atticus Project, Inc., "Contract Understanding Atticus Dataset (CUAD)," [Online]. Available: https://www.atticusprojectai.org/cuad.

[5] "Lambda Inference API for LLaMA 3.2 3B," [Online]. Available: https://lambda.ai/inference.

[6] I. a. F. M. a. M. P. a. A. N. a. A. I. Chalkidis, "nlpaueb/legal-bert-base-uncased," [Online]. Available: https://huggingface.co/nlpaueb/legal-bert-base-uncased.

[7] We used the Spacy's en_core_web_md. Available:https://spacy.io/models/en#en_core_web_md.