Intel® Unnati Industrial Training 2024

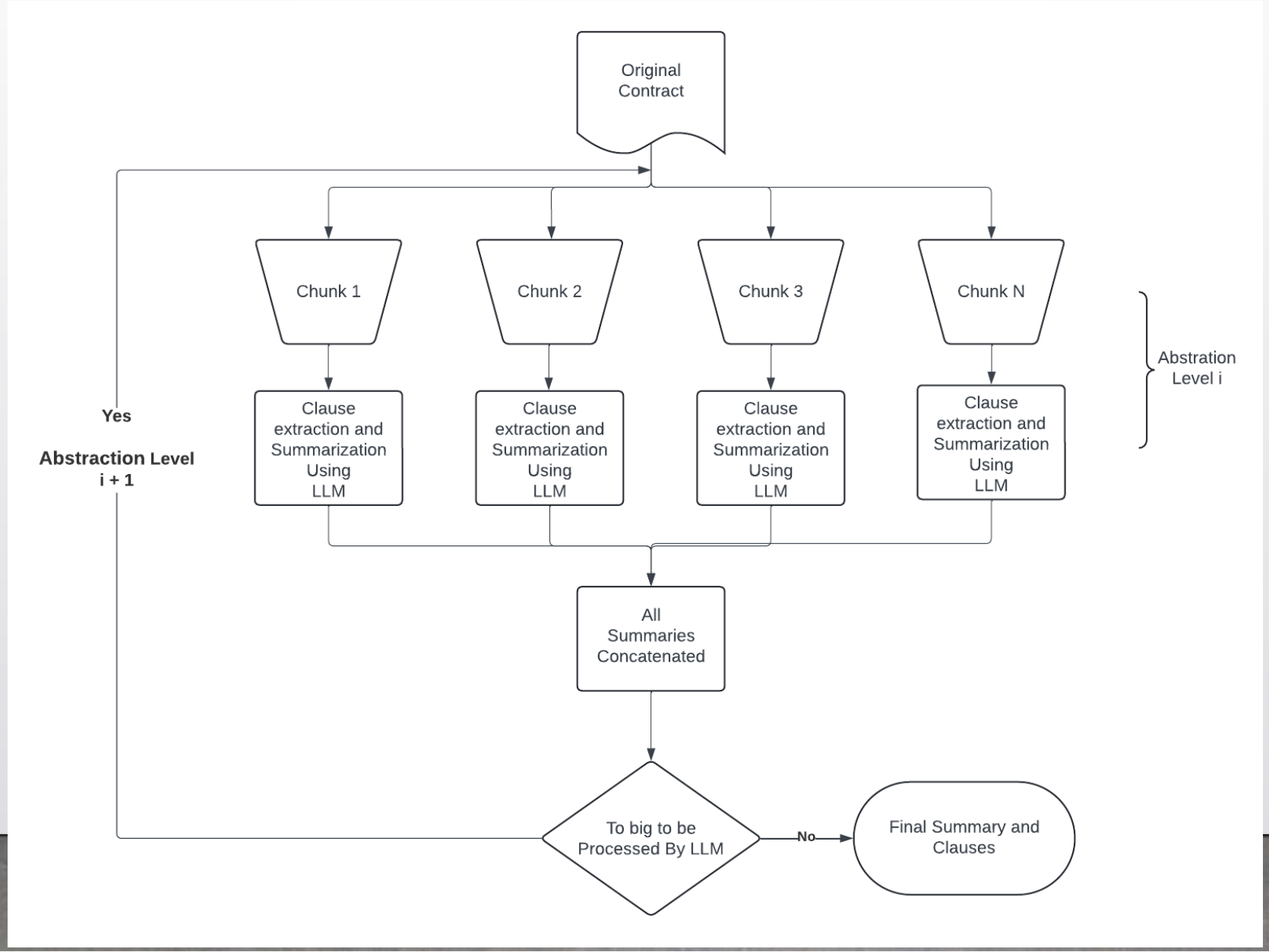
Business Contract Validation -To Classify Content within the Contract Clauses and Determine Deviations from Templates and highlight them.

By: Utkarsh Kumar College: Manipal Institute of Technology Email: utkarsh.kumar@learner.manipal.edu

Unique Idea

- Legal documents are crucial, and every line and word holds significant importance.
- Given the token limit of LLMs, the conventional approach involves breaking down extensive documents into smaller chunks and processing these fragments individually with the LLM.
- However, this method risks the loss of critical information.

Unique Idea


Original Contract
Chunk 1
Chunk 2
Chunk 3
Chunk N
Clause extraction and Summarization Using LLM
Clause extraction and Summarization Using LLM
Clause extraction and Summarization Using LLM
Clause extraction and Summarization Using LLM
Abstration Level i
Yes
Abstraction Level i + 1
All Summaries Concatenated
To big to be Processed By LLM
No
Final Summary and Clauses

Unique Idea

- To loose minimal information going through the contract the summarized chunks are concatenated and then summarized again.
- Each summary produced has a higher level of abstraction.
- Although this creates redundant information the user can skip to the final level of abstraction just to get a glimpse of the contract and If needed, they can go to lower abstraction levels to obtain more detailed information.

Features offered

1. Web Interface:
   - Provides a simple web interface for file upload and result display.
   - Displays summarization steps, including original text pieces and their summaries.
   - Shows the final summary for each recursion level.
   - Presents extracted clauses and subclauses in the first level of abstraction.
2. Error Handling and logging:
   - Includes basic error handling for file uploads and processing.
   - Displays flash messages for user feedback on errors or important information.
   - Implements logging to track the summarization process and any issues that arise.
3. Recursive Summarization:
   - Breaks down large texts into manageable pieces.
   - Performs multiple levels of summarization for comprehensive text reduction.
   - Maintains context across summarization levels using the Ollama chat API.

Features offered

4. Customizable Parameters:
   - Allows for adjustment of summarization parameters such as maximum length and recursion levels.
5. Natural Language Processing:
   - Uses NLTK for advanced text processing tasks like tokenization and stopword removal.
6. API Integration:
   - Utilizes the Ollama API (specifically the chat endpoint) for text summarization and clause extraction.
   - Maintains conversation context for more coherent and relevant summaries.

Process Flow

1. File Upload:
   - The user uploads a PDF or TXT file through the web interface.

2. File Validation:
   - The program checks if the file is provided and whether it is a supported format (PDF or TXT).

3. Text Extraction:
   - For a PDF file, text is extracted using PyPDF2.
   - For a TXT file, the text is read directly.

Process Flow

4. Text Preprocessing:
   - The text is tokenized into sentences and common stopwords are removed.
5. Recursive Summary:
   - The text is split into smaller pieces.
   - Each piece is summarized using the LLaMA API.
   - If the combined summary is too long, the process is repeated recursively up to three times to create a shorter summary.
6. Clause Extraction:
   - Main clauses are extracted from the preprocessed text using the LLaMA API.
   - Subclauses are extracted from each main clause using the LLaMA API.
7. Display Result:
   - The summarization steps and final summary are displayed on the web page.

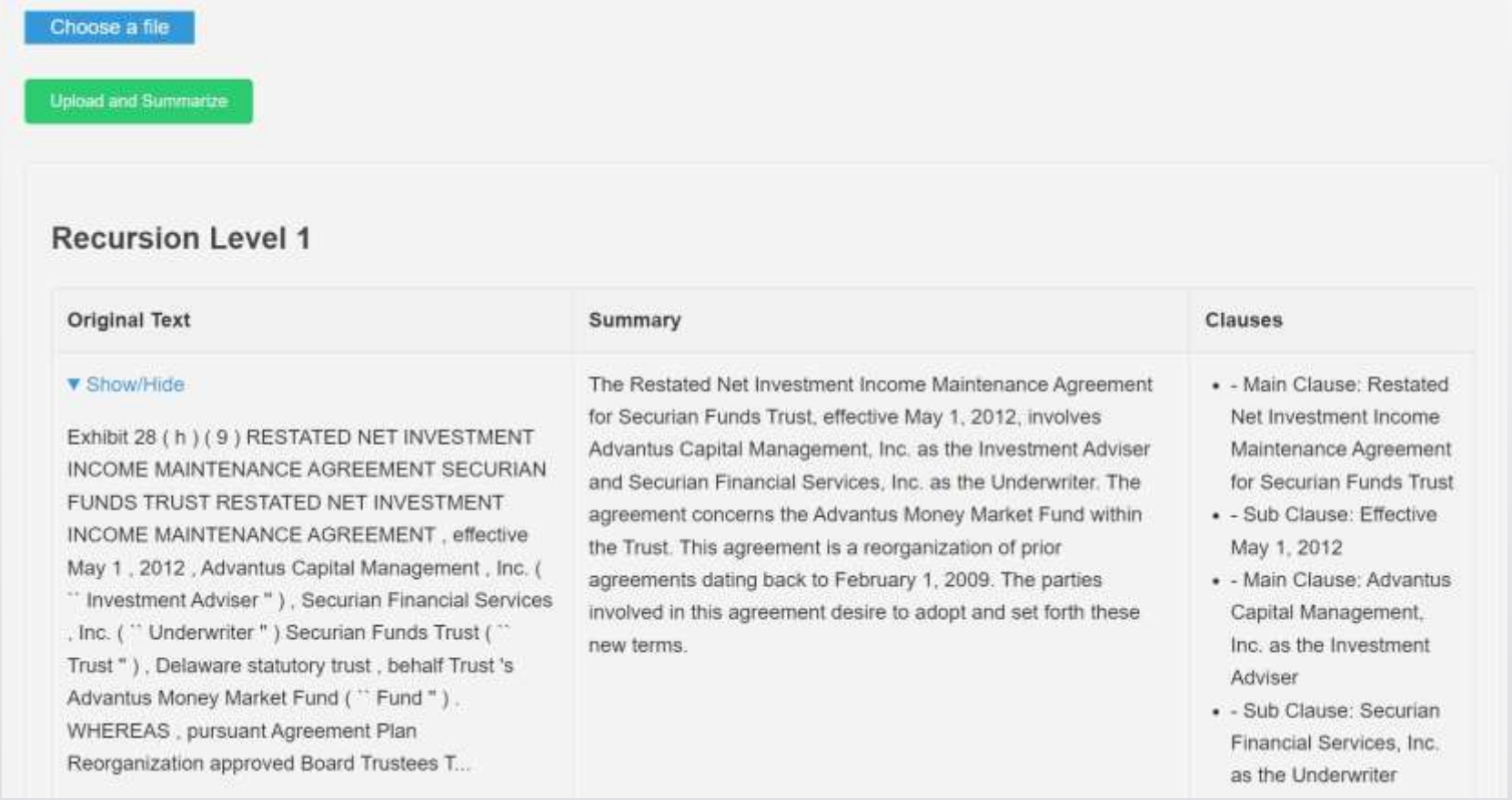
Technologies Used

- Flask: For creating the web application and handling HTTP requests.
- Requests: For making HTTP requests to the LLaMA API.
- Logging: For logging information and errors.
- JSON: For handling JSON data.
- PyPDF2: For extracting text from PDF files.

Technologies Used

- NLTK (Natural Language Toolkit):
  - sent_tokenize: For tokenizing text into sentences.
  - word_tokenize: For tokenizing sentences into words.
  - stopwords: For removing common stopwords.
- LLaMA API (Ollama Server): For generating summaries and extracting clauses.
- HTML and Jinja2: For rendering the web pages (via Flask's render_template function).

Output


Choose a file
Upload and Summarize
Recursion Level 1
Original Text
Summary
Clauses
▼ Show/Hide
Exhibit 28 (h) (9) RESTATED NET INVESTMENT INCOME MAINTENANCE AGREEMENT SECURIAN FUNDS TRUST RESTATED NET INVESTMENT INCOME MAINTENANCE AGREEMENT , effective May 1 , 2012 , Advantus Capital Management , Inc. (`` Investment Adviser '' ) , Securian Financial Services , Inc. ( `` Underwriter '') Securian Funds Trust (`` Trust '' ) , Delaware statutory trust , behalf Trust 's Advantus Money Market Fund ( `` Fund '') . WHEREAS , pursuant Agreement Plan Reorganization approved Board Trustees T...
The Restated Net Investment Income Maintenance Agreement for Securian Funds Trust, effective May 1, 2012, involves Advantus Capital Management, Inc. as the Investment Adviser and Securian Financial Services, Inc. as the Underwriter. The agreement concerns the Advantus Money Market Fund within the Trust. This agreement is a reorganization of prior agreements dating back to February 1, 2009. The parties involved in this agreement desire to adopt and set forth these new terms.
- Main Clause: Restated Net Investment Income Maintenance Agreement for Securian Funds Trust
- Sub Clause: Effective May 1, 2012
- Main Clause: Advantus Capital Management, Inc. as the Investment Adviser
- Sub Clause: Securian Financial Services, Inc. as the Underwriter

Output

- The summary along with the extracted clauses are represented in a table format for easy access in the first recursion level.
- This is done for providing the LLM with the most context.

- All stages are logged for better diagnosis when something goes wrong

```
127.0.0.1 - - [15/Jul/2024 17:05:01] "GET / HTTP/1.1" 200 -
Recursion Level: 1
Number of pieces: 6
Processing piece 1/6
Processing piece 2/6
Processing piece 3/6
Processing piece 4/6
Processing piece 5/6
Processing piece 6/6
Concatenated summary is too long. Starting next recursion level.
Recursion Level: 2
Number of pieces: 3
Processing piece 1/3
Processing piece 2/3
Processing piece 3/3
```

Team Members and their Contribution

This is an individual Project.

Name: Utkarsh Kumar

Registration Number: 210907246

College: Manipal Institute of Technology, Manipal

Email: utkarsh.kumar@learner.manipal.edu

Conclusion

The Text Summarizer project demonstrates a practical application of natural language processing techniques to solve the problem of information overload. By leveraging recursive summarization and clause extraction, it provides users with a powerful tool to quickly digest large volumes of text. While there are areas for improvement, the current implementation serves as a solid foundation for future development and expansion of capabilities.

Future Scope

- Develop a custom summarization model to reduce dependency on external APIs
- Implement user accounts for saving and managing multiple documents
- Add options for users to customize summarization parameters
- Develop a RESTful API for integration with other systems
- Implement multi-language support