Event-based Timeline Generation for Document Analysis

Author: Maria Miskaryan
BS in Data Science
American University of Armenia

Supervisor: Aram Aghababyan
Lawyer, Co-Founder @ CaseLens
University of Amsterdam

***Abstract*—This paper details a capstone project that leverages OpenAI APIs and text processing tools to automate timeline generation in investment arbitration, using data from the FDI-MOOT competition. The project integrates multiple OpenAI APIs, such as GPT-3.5 Turbo and GPT-4 Turbo with vision capabilities, to extract, clean, and analyze textual data from documents, converting them into a structured timeline of events. This methodology addresses the complexities and inefficiencies associated with manual timeline construction in legal documents. Key outcomes demonstrate the potential of OpenAI-driven processes to enhance accuracy and efficiency in legal data analysis. This paper explores the methods used, discusses the results, and acknowledges limitations while proposing future research directions.**

***Index Terms*—Timeline Generation, OpenAI APIs, Document Analysis**

I. Introduction

Investment treaty arbitration involves legal proceedings where timely and accurate data analysis is crucial. Traditional methods of extracting relevant information from extensive documentation are labor-intensive and error-prone. This paper introduces a novel application of an AI-driven approach to streamline this process, focusing on timeline generation for arbitration cases based on the FDI-MOOT competition dataset.

The need for innovation in this field arises from the growing complexity and volume of legal documents in arbitration cases. Automating timeline generation can significantly reduce the time and effort required by legal professionals, allowing them to focus on more strategic aspects of case preparation. The methodology employs a combination of text extraction using PDF text extraction tools, optical character recognition (OCR) using the state-of-the-art GPT-4 Turbo model with vision capabilities, text cleaning with GPT-3.5 Turbo model, date and event extraction using spaCy's named entity recognition and GPT-3.5 Turbo models, and finally merging of duplicate events using OpenAI embedding model with cosine similarity and thresholding to process documents and produce an organized chronological timeline of relevant events.

The FDI-MOOT competition, simulating investment treaty arbitration, provides a rich dataset that mirrors the challenges and complexities of real-world legal documents. This dataset serves as the foundation for developing and testing the timeline generation tool, ensuring the project's relevance and applicability to actual arbitration scenarios.

The main contributions of this project include developing a scalable and efficient tool for streamlining the generation of a timeline of chronological events, providing insightful information to legal professionals for further strategic decisions. This paper will detail the implementation of timeline generation using various OpenAI API capabilities, providing a comprehensive technical overview of the process from document processing to timeline generation, focusing on the methodology and technical aspects.

II. Results and Discussions

A. Results

The project successfully implemented a timeline generation tool that processes and organizes information from legal documents into a structured timeline. The proof of concept was tested on the FDI Moot 2024 case dataset, which consisted of a single PDF file containing various case-related documents. The tool demonstrated its adaptability by efficiently handling both readable and non-readable PDFs. Text extraction from readable PDFs was performed using PDF extraction tools with text cleaning utilizing the GPT-3.5 model, resulting in high accuracy, while non-readable PDFs were processed using optical character recognition (OCR) with the state-of-the-art GPT-4 turbo model, ensuring accurate extraction of text.

As a result of efficient data processing and text extraction techniques, along with incorporating techniques such as Spacy's named entity recognition model for ”date” entity recognition, and GPT-3.5 Turbo model with prompt engineering's best practices, including a chain of thought and few-shot prompting, the tool achieved high accuracy in extracting dates and events from the documents. This successful outcome underscores the effectiveness of the implemented methods in generating a precise chronological timeline of events from legal materials.

B. Findings

The findings reveal that AI-driven timeline generation significantly reduces the time required to prepare for arbitration cases compared to traditional methods. Lawyers often spend weeks aggregating evidence from numerous submissions to construct a chronological timeline of events relevant to a case, with thousands of documents to process. Having a tool that streamlines this time-consuming task by analyzing these documents within minutes instead of weeks represents a significant improvement over traditional manual methods. As a result, this helps lawyers in quickly grasping the case and comprehending its key aspects and sequence of events. In contrast, manually reviewing multiple case-related documents can feel like solving a puzzle, where it takes considerable time to piece together the larger picture. Finally, by integrating sources alongside identified events, the tool enables easy access to the original documents, simplifying further investigation.

C. Limitations

Despite the successes achieved, the project encountered some limitations. One such limitation came from the variability in the quality and structure of PDF documents. Different types of PDFs may contain diverse symbols and structures, posing challenges during the text extraction process. To address this challenge, the project implemented a text cleaning approach leveraging the capabilities of a large language model (LLM). This method accommodates any type of redundant information and symbols commonly found in legal documents, making it more adaptable than rule-based approaches.

Another limitation was posed by non-readable PDFs, which require additional steps for text extraction. To address this limitation, the project incorporated optical character recognition (OCR) with GPT-4 Turbo, leveraging its advanced vision capabilities. This ensures that even non-readable PDFs can be processed effectively and with high accuracy, overcoming the limitations associated with traditional OCR tools.

Additionally, the reliance on OpenAI APIs for the project's implementation introduces a potential limitation. While these APIs offer advanced capabilities, there may be associated costs that could impact further development. However, the decision to utilize OpenAI APIs underscores the project's commitment to leveraging state-of-the-art technologies for efficient document analysis, while further steps will be taken for cost optimization to mitigate the impact of associated costs and ensure efficient resource utilization.

D. Future Directions

Future research will focus on optimizing the use of OpenAI APIs to streamline the document analysis process.

Currently, multiple requests are made for cleaning, OCR, date and event extraction, and handling of duplicate events. The plan is to incorporate all these aspects into one combined API request within a single prompt while utilizing prompt engineering best practices, such as again using a chain of thoughts approach and incorporating more examples in the prompt. For this combined approach, GPT-4 Turbo, the most up-to-date model, will be utilized to ensure comprehensive context and accuracy.

Additionally, there may be a process of fine-tuning the LLM model on legal case documents to adapt it to our specific legal use case.

To conclude, the aim for the future is to reduce costs associated with using OpenAI APIs while enhancing performance for legal document analysis.

III. Conclusion

The capstone project successfully demonstrated the effectiveness of AI-driven techniques in efficiently extracting and generating a timeline of events from extensive investment arbitration case legal documents to enhance legal professionals' workflow. By using OpenAI APIs for various tasks such as OCR for non-readable PDFs, text cleaning, and date and event extraction, the project has significantly enhanced the accuracy of analyzing documents. This innovation not only saves time but also improves the reliability of data extraction and event chronology, which are crucial for effective case preparation in arbitration scenarios. The integration of these technologies addresses a critical need in the legal field, providing a scalable solution that can handle the increasing complexity and volume of arbitration documents. The potential impact of this tool extends beyond simplifying procedures; it introduces a new approach in legal document analysis that could influence future developments in legal tech. This project serves as a proof of concept that can be refined and expanded in future work for improved and faster performance at a lower cost.

IV. Methods

This section outlines the approach taken to implement the AI-driven timeline generation tool. The sections showed in Fig. 1., from IV-A to IV-G provide a detailed overview of each development step's methodology, covering implementation tools' details, data preparation, text extraction, OCR processing, text chunking, text cleaning, date and event extraction, and handling of duplicated events.

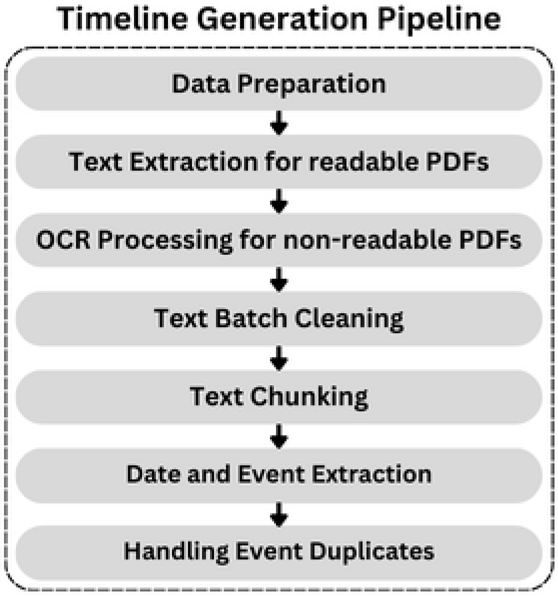
Timeline Generation Pipeline
Data Preparation
Text Extraction for readable PDFs
OCR Processing for non-readable PDFs
Text Batch Cleaning
Text Chunking
Date and Event Extraction
Handling Event Duplicates

Fig. 1. Pipeline of order of development steps.

A. Implementation

The implementation of the Timeline Generation Tool was conducted in Python, a programming language ideal for data manipulation and seamless integration with the OpenAI API.

The core functionality relied on the OpenAI API, which enabled various tasks, including OCR processing for non-readable PDFs and text cleaning from redundant symbols as a result of extracting text from readable PDFs using PDF extraction tool `fitz`, date and event extraction, and using OpenAI text embeddings for understanding event similarity and merging duplicate events.

The spaCy library played a crucial role in tasks such as named entity recognition for `date` entity recognition and segmenting text into manageable chunks. This approach was essential for efficiently handling the large volumes of text typically found in legal documents, employing state-of-the-art techniques rather than rule-based methods.

B. Data Preparation

Data preparation was a foundational step in the project, involving the conversion of the FDI Moot 2024 competition's case document from its original concatenated format into individual files. This step was crucial as it simulated the real-life scenario of processing where users typically upload separate documents instead of concatenating them, matching the typical user experience. The separation of documents was achieved using a custom script which identified document boundaries based on the table of contents page of the case dataset and separated them into individual documents according to these page ranges. As a result, this process gave a dataset suitable for proof-of-concept testing of the timeline generation idea matching the future user experience.

C. Text Extraction and OCR Processing

Text extraction and OCR processing were crucial steps in obtaining information from both readable and non-readable PDF files.

For all PDFs, text extraction was conducted using the Python library `fitz`, which provided robust support for navigating and extracting text. `fitz` was selected for its efficiency in handling large PDF files while preserving the original layout of the documents.

For PDFs with text length lower than a specified threshold of 30 characters, indicating minimal or no information, OCR processing was employed. This was accomplished using the GPT-4 Turbo model, known for its vision capabilities.

The OCR processing workflow involved several steps:

- First, documents suitable for OCR processing were identified based on their low text length.
- From the PDFs needing OCR, all pages were saved as separate images in JPEG format.
- These images were then encoded into base64 format for input to the GPT-4 Turbo model.
- The encoded images, along with a comprehensive prompt utilizing prompt engineering best practices, such as chain of thought prompting and one-shot format example, were provided to the model.
- Batch processing was utilized, typically with three pages per batch, ensuring consistent information extraction from the images and allowing for effective OCR processing even for longer documents. This approach ensures that the algorithm can handle documents of varying lengths, providing reliable extraction regardless of the document's size.
- The outputs from all batches were combined to obtain the complete text for each document.

This approach ensured efficient and accurate extraction of text from diverse PDF formats, both from readable PDFs with the help of the `fitz` library, and from non-readable PDFs with the help of accurate OCR using GPT-4 Turbo models' advanced vision capabilities.

D. Text Chunking and Batching

For text chunking and batching, a custom logic was implemented to split the text into manageable parts using spaCy paragraphs. By default, the function performs chunking, but with an additional argument, it also supported batching.

- Chunking Method: For text chunking, the text was split into chunks with a minimum length set to 1,000 characters. SpaCy paragraphs were combined together until this minimum threshold was met. Once met, the chunk was considered complete. Indexed chunks were generated for each document, and these indexes were later used to display the corresponding chunk if a date was found within it.
- Batching Method: If the batching argument was set to true, an alternative logic was implemented for batching. This method involved combining previously split chunks until reaching the maximum batch threshold, which was set by default to 25,000 characters. No indexing was used for batching; instead, the resulting big batches of text were simply saved as they were for each document. This batching logic was primarily used for the further text cleaning step, where the document text was processed in batches of three, with each batch close to 25,000 characters in size. This approach ensured that text cleaning would be effective even for very long documents, such as those spanning multiple pages. Once cleaned, the text was chunked into smaller parts for date and event extraction.

E. Text Cleaning

Once text was extracted, it underwent a cleaning process to ensure that only relevant textual content was retained for further processing. This involved the removal of redundant or irrelevant information such as headers, footers, line numbers, and legal references that were contextually out of place and irrelevant to the content being processed. The text underwent a cleaning process using OpenAI GPT-3.5 Turbo model, which was prompted to identify and remove various types of non-relevant text symbols and information and return the exact text. It was chosen for its advanced NLP capabilities and cost-effectiveness compared to the GPT-4 Turbo model. The cleaned text was generated batch by batch, where each batch underwent cleaning individually before being connected to form the full text for each document for handling long documents. The process takes a considerable amount of time, yet the final results are delivered in minutes, in contrast to the weeks typically required by legal professionals.

F. Date and Event Extraction

The extraction of dates and events involved a two-step process.

- Firstly, from all chunks of text dates were extracted using spaCy's named entity recognition (NER) for `date` entity recognition. SpaCy's model was good at recognizing all dates in various formats, including relative expressions like "tomorrow" or "next week."
- Following date extraction, the GPT-3.5 Turbo model was prompted to validate, identify, and standardize valid dates into a dd/mm/yyyy format while considering the surrounding context, that is the date chunk along with 2 preceding and 2 next chunks. If a date was invalid, the process was stopped to avoid inaccuracies.

Additionally, the GPT model was tasked with associating events with the identified dates. Alongside each date, a brief contextual prompt was provided to assist in identifying the event, ensuring that events were accurately linked to their corresponding dates. As a result, each extracted event consisted of a four or five-word event title describing the occurrences on each date, accompanied by a one-sentence event description/summary.

To enhance accuracy, the output from the GPT model was carefully parsed. Dates were validated to ensure consistency in format, mitigating any potential errors. Finally, all extracted dates and associated events were organized chronologically, providing a clear timeline of events from oldest to most recent.

By combining spaCy's NER capabilities with GPT-3.5 Turbo's contextual understanding, the extraction process gave precise and comprehensive results, generating accurate timelines.

G. Handling Event Duplicates

As some events and their corresponding dates were mentioned across multiple documents and chunks, there was a possibility of duplicate events. To address this, a custom logic was implemented to compare event titles and summaries using cosine similarity between the embeddings generated by OpenAI's `"text-embedding-3-small"` text embedding model. This approach ensured that each event appeared only once in the final timeline.

Using the embedding model, embeddings were obtained for event titles and compared for cosine similarity with a threshold of 0.75. Similarly, cosine similarity was calculated for event summaries with a threshold of 0.7. The higher threshold for title similarity accounted for their consistent nature, typically consisting of four or five words. In contrast, event summaries, despite being one sentence, could vary in wording and length, resulting in a slightly lower similarity threshold.

Events with similarity scores exceeding the thresholds were considered duplicates and merged. One of the duplicate events' titles and summaries were retained, while both chunks and documents where the events were mentioned were preserved in the tool for comprehensive review. This ensured that users could access all relevant information regarding to a particular event, despite of its occurrence in multiple documents or chunks.

Following the merging process, a final chronological timeline was generated, containing dates, event descriptions, document references, and document chunks where the events were found.

Ultimately, integrating this tool into broader legal tech platform, such as CaseLens, can revolutionize how legal professionals interact with extensive document sets, offering significant time savings and operational efficiencies. This project lays the groundwork for a transformative approach to legal document analysis, with potential applications that extend far beyond the scope of the initial study.