Documentation for IP

MOTIVATION:

Legal data is one of the key determinants in forming decisions, policies, and dispensing justice; however, the vastness and complexity that surround it often act as a barrier to accessibility and effective utilization. The motivation for this project of collecting and processing legal data using large language models is because of the immense potential these models hold to transform things. The analysis of statutes, case laws, and contracts can be streamlined by utilizing the advanced natural language understanding and processing capabilities of LLMs. By using LLMs, this project attempts to improve domain knowledge by allowing more profound insights into legal precedents, providing better access to information for legal professionals, and facilitating innovations such as automated legal advice or monitoring compliance. The project also allowed us to learn and develop various new skills, such as data collection and preprocessing, prompt engineering, fine-tuning LLMs for domain-specific tasks, and understanding legal terminologies and frameworks. These skills not only enriched our technical expertise but also bridged the gap between technology and the intricate world of legal systems.

RESEARCH QUESTION:

The primary goal of this project is to address the challenge of making a comprehensive global dataset within the legal domain that can encapsulate meaningful insights obtained from legal cases. We will use advanced large language models to automate the analysis and annotation of legal texts into entities that are active agents and passive agents, together with moral decisions, legal decisions, and their relations. This way, this work will try to contribute toward a structured, enriched, and universally accessible repository of legal data that goes beyond both linguistic and jurisdictional frontiers. This dataset will be of use for deeper cross-jurisdictional pattern discovery, comparative legal studies, and applications

such as predicting legal decisions, compliance analysis, and AI-based legal reasoning. In doing so, we not only hope to push the boundaries further of integrating AI in the legal domain but also equip professionals, policymakers, and researchers with a powerful tool for enriching understanding, efficiency, and innovation in legal practices.

LITERATURE REVIEW:

1. **A Survey on Dataset generation and Augmentation in LLMs**

   This data synthesis and augmentation study for LLMs explores the state-of-the-art techniques and methodologies, with the intention to make the large language model more efficient and effective. Some of the methods related to domain-specific data generation in the process have been brought under the umbrella to further demonstrate the potential for preparing specialty datasets to enable the comprehension and processing of more intricate information by LLMs. Additionally, automated data labeling and reformatted data will streamline dataset preparation to have consistency, accuracy, and relevance.

   Key areas of focus include the role of prompt engineering to guide LLM outputs toward specific tasks and contexts, as well as the incorporation of multi-step generation processes to produce varied and high-quality synthetic data. The survey explores how such advancements strengthen the core functionalities of LLMs like comprehension, reasoning, and adaptability while addressing limitations by data scarcity or domain-specific challenges.

   In addition, the paper evaluates the effectiveness of these approaches with robust benchmarks and performance metrics, thus providing valuable insights into their practical applications. It also looks into the use of external resources, such as knowledge graphs and ontologies, to improve the quality of the data generated and its alignment with real-world scenarios. This comprehensive review not only underlines the transformative potential of

data synthesis and augmentation but also draws a roadmap for deploying these techniques across diverse fields, such as legal analysis, scientific research, and more.
(https://ar5iv.org/html/2410.12896)

2. **LegalLens: Leveraging LLMs for legal violation Identification in Unstructured Text.**
   This paper describes a methodology for building datasets to detect legal infringements in unstructured text and link them to parties affected. It applies LLMs to process and annotate legal documents, employing NER and NLI models. The datasets were produced from legal complaints and news articles by summarizing them with GPT-4 and custom prompts to make sure that they are relevant to the domain. This work focuses on the significance of creating structured datasets for violation identification and victim association, which have achieved significant results with fine-tuned and few-shot models.
   (https://ar5iv.org/html/2402.04335)

METHODOLOGY

1. **Domain Understanding:**
   During our meeting with Ms. Aisha, we were presented with a comprehensive PowerPoint presentation that significantly enhanced our understanding of the legal domain, particularly the intricate role of ontologies in structuring and organizing legal knowledge. The session provided deep insights into how legal concepts and relationships are systematically represented, enabling a clearer understanding of how datasets could be structured for AI applications.

We were also provided with sample cases and prompts, which would act as practical examples in order to help us understand the kind of deliverables that our project would require. Such examples illustrated the subtleties of legal reasoning, identifying active and passive agents, and drawing a distinction between moral and legal judgments.

To further solidify our knowledge, we did a great amount of research across the platforms of Quora, Reddit, and many other specialized forums on the law. This has allowed us to look at so many different types of real-world legal cases, including civil disputes and corporate compliance. We looked at what was done legally in each case and how it fit within legal standards, thus furthering our understanding of the laws that guide legal decisions.

Through these activities, we also gained insights into the variability of legal interpretations across jurisdictions and contexts. We also explored how user-generated discussions on forums often highlight common pain points and ethical dilemmas, which could be instrumental in enhancing the relevance and inclusivity of our dataset. These efforts collectively advanced our ability to structure and analyze legal data effectively, aligning with the project's objectives to bridge technology and legal expertise.

2. **Web Scraping Raw data**

   After analysing various websites we realised that reddit and various subreddit pages were the best source to extract raw cases. We extracted and scraped data from sub reddit pages of various countries like India, Canada, UK, etc.

   In order to web scrape the data we had to create a python script.

   pseudoCode :

   /* Base Configuration */

```
base-url: " ";
endpoint: " ";
category: " ";
url: base-url + endpoint + category + ".json";
after-post-id: null;
dataset: [];
/* Fetch Data Loop */
loop (maximum-iterations) {
   /* Request Parameters */
   params {
      limit: 100;
      t: "year";
      if (after-post-id != null) {
         after: after-post-id;
      }
}

   /* API Request */
   response: GET(url, params, headers: {
      user-agent: "Mozilla/5.0"
   });
   /* Validate Response */
   if (response.status-code != 200) {
      throw error("Failed to fetch data. Status code: " + response.status-code);
   }
   /* Parse Response */
   json-data: parse(response.json);
   if (json-data == invalid) {
      throw error("Invalid JSON response.");
```

```
    }
    /* Extract and Filter Data */
    for-each (record in json-data.data.children) {
        post-data: record.data;
        filtered-data {
            title: post-data.title || "N/A";
            selftext: post-data.selftext || "N/A";
            upvote-ratio: post-data.upvote_ratio || "N/A";
        }
        dataset.append(filtered-data);
    }

    /* Update Pagination */
    after-post-id: json-data.data.after;
    if (after-post-id == null) {
        break;
    }
    /* Respect Rate Limits */
    wait: 0.5 seconds;
}
/* Convert and Export */
output-file: "reddit_legal_advice_OffTopic.xlsx";
dataframe: convert(dataset-to-dataframe);
export(dataframe, output-file);
print("Data written to " + output-file);
```

In total we extracted raw data from 6 subreddit pages getting a total of around 6k raw data which is of the format :

Title, Case text , Upvote Ratio

3. **Feature Extraction**

   From the cases we web scraped we tried to extract all the relevant features which can help us get a deep insight into the legal aspects of the cases.

   Extracted features:

   1. Active agent
   2. Passive agent
   3. Action done by active agent
   4. Domain
   5. Ethical issue(s)
   6. Consequence
   7. Severity of consequence
   8. Utility of consequence
   9. Duration of consequence
   10. Moral intention of active agent
   11. Ethical principles upheld and ethical principles violated the relationship between active agent and passive agent.

Prompt used :

Analyze the following sentence: {selftext} to extract the following features from the cases:

- Active agent: The individual or entity that performs an action or initiates a process within an scenario.

- Passive agent: The individual or entity that is affected or impacted by the action performed by the active agent

- Action done by an active agent: The specific act or behavior undertaken by the active agent that influences the passive agent.

- *Domain : The context or area (e.g., healthcare, business, technology) in which the action takes place, influencing the ethical implications.*

- *Ethical issue(s) : the moral conflicts that arise from the action, questioning what is right or wrong in the scenario.*

- *Consequence : The outcome or effect that results from the action of the active agent on the passive agent or the environment.*

- *Severity of consequence : The degree of harm or benefit caused by the consequence, ranging from mild to severe.*

- *Utility of consequence: determining whether it benefits or harms stakeholders.*

- *Duration of consequence : The length of time for which the consequence persists, either immediately or over the long term.*

- *Moral intention of active agent: The ethical purpose or goal that the active agent aims to achieve through their actions.*

- *Ethical principles upheld: The moral values or standards that are supported and respected by the active agent's actions.*

- *Ethical principles violated: The moral values or standards that are disregarded or harmed by the active agent's actions*

- *Relationship between active agent and passive agent: The nature of the interaction or connection between the individuals or entities involved, which may affect the ethical dynamics.*

Understand the below example for clarity :

A company decides to lay off 100 employees to cut costs and increase profits. The CEO, as the active agent, implements this decision, leading to the job loss of several employees (the passive agents). The action results in some employees struggling financially, while the company profits in the short term. The ethical issue involves whether the company should prioritize profit over the well-being of its employees. The consequence is the emotional and financial hardship faced by the laid-off employees, which is severe in some cases. The utility of this consequence is questioned, as it

benefits the company but harms the affected employees. The consequence's duration could last long-term for the employees. The CEO's moral intention might have been to save the company from financial ruin. The action upholds the ethical principle of utilitarianism (maximizing overall profit) but violates principles of fairness and respect for individuals. The relationship between the CEO and the employees is one of employer-employee, with a power imbalance.

Identification of Ethical Elements:

Active agent: The CEO (person implementing the layoff decision).

Passive agent: The laid-off employees (those impacted by the layoffs).

Action done by active agent: The CEO decides to lay off 100 employees to cut costs.

Domain: Business and corporate ethics (focused on decisions related to employment and profit).

Ethical issue(s): Should the company prioritize profit over the well-being of employees?

Consequence: Financial and emotional hardship for the employees; increased profits for the company.

Severity of consequence: Severe for the employees, especially those facing long-term financial struggles.

Utility of consequence: Positive for the company (increased profit) but negative for the employees (financial and emotional strain).

Duration of consequence: Long-term for the affected employees, potentially leading to long-lasting financial insecurity.

Moral intention of active agent: To save the company from financial distress and maintain profitability.

Ethical principles upheld: Utilitarianism (maximizing overall profit).

Ethical principles violated: Fairness, respect for individuals, and care for the well-being of employees.

Relationship between active agent and passive agent: Employer-employee relationship with a significant power imbalance.

Provide your response in JSON format with the keys matching the above features.

Pseudocode:

 # 1. Setup Environment

environment {

api-key: "Your Together API Key"

client: Initialize Together(api-key)

model: "meta-llama/Meta-Llama-3.1-8B-Instruct-Turbo"

}

 # 2. Define Prompt Generation

function generate_prompt(selftext):

prompt: """

write the prompt

”””

return prompt

 # 3. Define LLM Interaction

function run_agent(client, prompt, model, context):

try:

response: client.chat.completions.create(

model=model,

messages=[

{"role": "assistant", "content": context},

{"role": "user", "content": prompt}

```
        ]
    )
    raw-response: response.choices[0].message.content.strip()
    if raw-response.isEmpty():
        throw error("Empty response from LLM")
    return parse(raw-response as JSON)
  except Exception as e:
    return {"Error": str(e)}
# 4. Load Input Dataset
data: LoadJSON("input_file.json")
columns: [
  "selftext",
  "Active agent", "Passive agent", "Action done by active agent", "Domain",
  "Ethical issue(s)", "Consequence", "Severity of consequence",
    "Utility of consequence", "Duration of consequence", "Moral intention of active agent",
  "Ethical principles upheld", "Ethical principles violated",
  "Relationship between active agent and passive agent", "Error"
]

# 5. Initialize Results Structure
rows: []
# 6. Process Each Row
for each row in data:
  selftext: Extract(row.selftext)
  prompt: generate_prompt(selftext)
   response: run_agent(client, prompt, model, "Role: Text analyst and legal domain expert")
  processed-row: {column: response.get(column, None) for column in columns}
```

processed-row["selftext"] = selftext

Rows.appen (processed-row)

 # 7. Export Results

results_df: CreateDataFrame(rows, columns)

results_df.to_csv("results_dataset.csv", index=False)

 # 8. Completion Message

print("Results saved as 'results_dataset.csv'")

4. LLM Summarisation

We wrote a code to summarise the cases and give a brief insight into these cases. We did the summarization part after the feature extraction so that we get good and well structured summarised data.

Prompt Used:

Summarize the case text using this template as accurately as possible while maintaining correct English grammar. Do not add extra information:

"The <active agent> did <action> to <passive agent> which led to <consequence>.The <active agent> had <good/bad/neutral> moral intention,

however, the <action> violated <ethical principle> ethical principle which

caused <ethical issue>."

Case text is as follows: "{case_text}"

give the output in comma separated format

PseudoCode:

 # 1. Setup Environment

environment {

api-key: "Your Together API Key"

client: Initialize Together(api-key)

```
    model: "meta-llama/Meta-Llama-3.1-8B-Instruct-Turbo"
}
# 2. Define Prompt Generation
function generate_prompt(case_text):
    prompt: (As mentioned above)
    return prompt
# 3. Define LLM Interaction
function run_agent(client, prompt, model, context):
    try:
        response: client.chat.completions.create(
            model=model,
            messages=[
                {"role": "assistant", "content": context},
                {"role": "user", "content": prompt}
            ]
        )
        raw_response: response.choices[0].message.content.strip()
        if raw_response.isEmpty():
            throw error("Empty response from LLM")
        return raw-response
    except Exception as e:
        return {"Error": str(e)}
# 4. Load Input Dataset
data: LoadJSON("practice.json")
columns: [
    "case_index", "summary", "selftext", "Error"
]
# 5. Initialize Results Structure
rows: []
```

 # 6. Process Each Row

for each row in data:

case_index: Extract(row.index)

selftext: Extract(row.selftext)

prompt: generate_prompt(selftext)

response: run_agent(client, prompt, model, "Role: Legal domain expert generating case summaries")

if "Error" in response:

processed_row: {

"case_index": case_index,

"summary": None,

"selftext": selftext,

"Error": response["Error"]

}

else:

processed-row: {

"case_index": case_index,

"summary": response,

"selftext": selftext,

"Error": None

}

rows.append(processed_row)

 # 7. Export Results

results_df: CreateDataFrame(rows, columns)

results_df.to_csv("summary.csv", index=False)

 # 8. Completion Message

print("Results saved as 'summary.csv'")

Augmentation:

In this segment our aim is to augment and create more instances of legal data with the help of LLMs. This in turn helps in diversifying our dataset to more variety of legal cases. We try to generate 5 cases from each case we have web scraped.

Prompt Used:

Behave as an instance generator. I will provide an example in JSON format. I need answer only in JSON format no unnecessary comments needed in the output
For each example, generate five new instances that are similar in structure but distinct in the following aspects:
- Context: Use a different setting or scenario that aligns with real-world situations.
- Agents: Identify distinct active and passive agents with clearly defined roles and relationships.
- Ethical issues: Introduce new and realistic ethical dilemmas relevant to the scenario.
- Features: Maintain the same set of features as the original example, providing detailed, well-structured, and contextually accurate values.
But make sure that these features are not more than 2-3 words

1. Ensure each generated instance explores diverse domains (e.g., healthcare, technology, education, business, law, etc.).
2. Clearly differentiate between the active agent (initiator of the action) and the passive agent (affected party).
3. Ensure the ethical issue is thought-provoking and aligns with the action and consequence.
4. Provide detailed descriptions for the consequence, its severity, utility, and duration.
5. Avoid repetitive or overly similar cases. Each instance should introduce fresh perspectives.

Here is an example in JSON format:
{example_json}
give the output as a comma separated value format

PseudoCode:

```
# 1. Define Prompt Generation
function generate_case_prompt(example):
    example-json: Convert(example to JSON format with indentation)
    prompt:(As mentioned above)
    return prompt
# 2. Extract JSON Blocks from Response
function extract_json_from_response(response_text):
    try:
        json-blocks: Find all patterns matching valid JSON in response_text
        parsed-data: []
        for each block in json-blocks:
            try:
                parsed-json: Parse(block as JSON)
                parsed-data.append(parsed-json)
            except JSONDecodeError:
                continue  # Ignore invalid JSON blocks
        return parsed-data if parsed-data exists else None
    except Exception as e:
        return {"Error": str(e)}
# 3. Define LLM Interaction
function run_agent(client, prompt, model, context):
    try:
        response: client.chat.completions.create(
```

```
            model=model,
            messages=[
                {"role": "assistant", "content": context},
                {"role": "user", "content": prompt}
            ]
        )
        response-text: Extract response content and strip whitespace
        return extract_json_from_response(response-text)
    except Exception as e:
        return {"Error": str(e)}

# 4. Load Input Dataset
data: LoadJSON("practice.json")
# 5. Initialize Results Structure
all-generated-cases: []
invalid-responses: []
# 6. Process Each Row
for each example in data:
    example-json: Convert example row to dictionary
    prompt: generate_case_prompt(example-json)
        response-data: run_agent(client, prompt, model, "You are a JSON instance generator and legal domain expert.")

    if response-data exists:
        all-generated-cases.append(response-data)  # Append generated cases
    else:
       invalid-responses.append({"example": example-json})
# 7. Export Results
if all-generated-cases exists:
```

```
    try:
        results-df: CreateDataFrame(all-generated-cases)
        results-df.to_csv("augmentation.csv", index=False)
        print("Generated cases saved as 'augmentation.csv'")
    except Exception as e:
        print("Error saving generated cases: {e}")
else:
    print("No valid cases generated.")

if invalid-responses exists:
    try:
        Save invalid-responses to "invalid_responses.json"
        print("Invalid responses saved as 'invalid_responses.json'")
    except Exception as e:
        print("Error saving invalid responses: {e}")
else:
    print("No invalid responses.")
```

Validation/Evaluation:

We tried to change the model to **Gemma.**

Prompt Used:

Behave as an instance generator. I will provide an example in JSON format. I need answer only in JSON format no unnecessary comments needed in the output

For each example, generate five new instances that are similar in structure but distinct in the following aspects:

- Context: Use a different setting or scenario that aligns with real-world situations.
- Agents: Identify distinct active and passive agents with clearly defined roles and relationships.

- Ethical issues: Introduce new and realistic ethical dilemmas relevant to the scenario.
- Features: Maintain the same set of features as the original example, providing detailed, well-structured, and contextually accurate values.
But make sure that these features are not more than 2-3 words

1. Ensure each generated instance explores diverse domains (e.g., healthcare, technology, education, business, law, etc.).
2. Clearly differentiate between the active agent (initiator of the action) and the passive agent (affected party).
3. Ensure the ethical issue is thought-provoking and aligns with the action and consequence.
4. Provide detailed descriptions for the consequence, its severity, utility, and duration.
5. Avoid repetitive or overly similar cases. Each instance should introduce fresh perspectives.

Here is an example in JSON format:
{example_json}
give the output as a comma separated value format

Problems and Solution:

1. **Problem:** While doing this project under the guidance of Dr. Raghava and Ms. Aisha we learnt a lot of new things which enhanced our brainstorming skills. As we were running heavy LLM models it was very difficult to execute them on our local systems. We tried arranging for the GPU access but being B.Tech students we were not given access for the same.
   Solution: After a lot of research and exploration we found Kaggle Platform which had 30 hrs. Free virtual GPU. Using this platform we accessed the **GPU P100.** This helped us a lot to reduce the computation time for the large datasets we had.

2. **Problem:** Since we were making a LLM aided legal domain dataset, it involved a lot of experimentation with various models for which we required their API keys. Initially we thought of using GPT 4.0 version because of its high accuracy but it did not have any free credits available. After facing this issue, we tried various lighter LLM models like GPT 2.0,etc. But they didn't give us high accuracy. We also tried to adapt an NLP based approach but they were also not very accurate.
   Solution: After a lot of brainstorming and researching for about 10 days we found a platform called **Together.ai.** This platform provided us with Free credits for API keys up to $1. We built multiple gmail accounts to access the same and populate our legal domain dataset with extracted features and summarised content.

3. **Problem:** Since we are using Large Language Models, Prompt engineering and enhancement plays a very crucial role. Initially, we were not able to get appropriate outputs because of poor prompts.

Solution: We tried to enhance our knowledge in the legal domain so that we have a better idea of the relevant features we could extract. We studied about 100 cases and manually extracted their relevant features and summarised its text. This helped us gain more ideas about the dataset we are dealing with. Atlast, after a lot of experimentation we came up with very refined prompts and also added various examples for the model to better comprehend the data.

4. **Problem:** Since the dataset is very large while running the code, the LLM was hallucinating and not giving appropriate answers in between.
   Solution: We decided to send the raw data with title and selftext in small packets. After a packet is executed the LLM gets refreshed clearing its cache. Hence, giving us appropriate results.

5. **Problem:** During data augmentation we were not able to generate a large variety of cases belonging to the legal domain. We were getting augmented data of similar types.
   Solution: We tried to enhance the prompts. We removed the example that it used to take to train the model. Rather now, we explicitly mentioned a few diverse domains in which it can generate relevant cases.

6. **Problem**: Initially we decided to scrap the first 20 comments of each case to validate the results but a major problem we faced was that the comments were usually not very relevant to the case mentioned hence they were not providing us any useful and relevant results.
   Solution: We were initially using **Llama 3.** Version to populate the dataset. Now for validation as well we thought of using another LLM. Hence, we tried to validate the results of one LLm by the other. For validation we used **Gemma** which also generated feedback where ever required.As a part of our

discussion with Dr. Raghava, both of us took the same 100 cases and tried to generate a list of their extracted features. Then we both compared our results with each other and the Gemma LLM. We could conclude that the results were usually very accurate apart from a few outlier instances.

CONCLUSION:

At the end we were able to produce a legal domain LLM aided dataset which could be used in the real world scenarios for research purposes. We get clearly laid out feature extracted columns and summary for each case. We also have implemented the code for augmenting cases into the dataset.

usage - kaggle + together.ai

Overview