AI Engineer Assignment

Develop a System to Extract Claims from Records Using Large Language Models (LLMs)

Data

Problem Statement:

Your task is to create a system that can accurately extract specific parts of records discussing particular claims. The records contain various types and structures of textual data relating to user feedback, interaction logs, and other conversational data. Extracting these claims can be challenging due to the diversity and complexity of the textual content.

You are required to build a robust solution using Large Language Models (LLMs) to identify and extract relevant claims within these records. Following the development phase, you will encapsulate this functionality into a reusable and scalable library.

Data Overview:

You will be provided with a CSV file containing records with the following columns:

- **ID:** Unique identifier for each record.
- **Content:** The main content of the record, which may include conversations, feedback, or other textual data.
- **URL:** A link to the source of the record.
- **Source:** The platform or tool from which the record was obtained (e.g., Slack, Salesforce).
- **Record Type:** The type of record (e.g., conversation, survey).
- **Timestamp:** The date and time when the record was created.

Steps to be followed:

1. **Data Understanding and Preprocessing:**
    - Analyze the structure and content of the records.
    - Implement preprocessing steps to clean and prepare the data for analysis. This may include handling missing values, normalizing text, and splitting text into manageable units.
2. **Claim Identification:**
    - Develop a method to identify and extract claims from the records. This involves understanding the context in which claims are made and distinguishing them from other parts of the text.

- Use LLMs to enhance the accuracy of claim identification. Fine-tune the model if necessary to better suit the specific nature of the records.

3. **Text Splitting and Annotation:**
   - Implement functions to split the text into smaller units (e.g., sentences, paragraphs) and annotate these units with relevant metadata (e.g., speaker, timestamp, message index).
   - Ensure that the splitting mechanism preserves the context necessary for accurate claim extraction.
4. **Model Integration:**
   - Integrate the LLM into your system to process the preprocessed text and identify the units where claims are discussed.
   - Develop a mechanism to return the indices of these units in a structured format.
5. **Library Development:**
   - Encapsulate the developed functionality into a reusable and scalable library.
   - Ensure that the library is well-documented and easy to integrate with other systems.

Expected Output:

The system should output a structured format indicating where the claims are discussed within the records. For example:

```
[
  {
    "record_id": "a879cf1-120c-5a69-b059-5820f08abae3",
    "claim_indices": [2, 5, 7]
  },
  ...
]
```

Evaluation Criteria:

1. **Accuracy:**
    - The system should accurately identify and extract claims from the records.
    - The extracted claims should be relevant and contextually correct.
2. **Efficiency:**
    - The system should process records in a reasonable amount of time.
    - The preprocessing and claim identification steps should be optimized for performance.
3. **Code Quality:**
    - The code should be well-structured, modular, and follow best practices.
    - The library should be well-documented, with clear instructions on how to use it.
4. **Innovation:**
    - The solution should demonstrate innovative use of LLMs and other NLP techniques.
    - The approach should be creative and effective in addressing the complexities of the task.

Submission:

- Submit the complete codebase along with a README file explaining the setup and usage of the library.
- Provide a brief report detailing your approach, challenges faced, and how you addressed them.
- Include examples of input records and the corresponding extracted claims to demonstrate the effectiveness of your solution.