Coding Assignment: AI Engineer Intern Summer 2024

Assignment: Building an AI-Powered Document Understanding and Processing Pipeline

Objective:

This assignment assesses your ability to design and develop an AI-powered pipeline for understanding and processing documents using advanced NLP techniques, Large Language Models (LLMs), and Optical Character Recognition (OCR). You will tackle real-world challenges in document comprehension, information extraction, and automation.

Overview:

Your task is to build a pipeline that uses an LLM to intelligently process documents (especially invoice/receipt type PDFs that contain both text and image content where the image content might have some text), extract essential information (key-value pairs most importantly), and enable user interaction through a chatbot interface. The pipeline should be capable of handling various document formats and adapting to different information extraction tasks, including scenarios where OCR is required to extract text from images within documents.

Test Documents:

ZIP **test data.zip** 483.05 kB

Part 1: Document Conversion, OCR, and Preprocessing

Task:

1. **Document Conversion:** Develop a system to handle document format PDF and convert it into a standardized format (e.g., .TXT) suitable for LLM processing.
2. **OCR Integration:** Implement a system to identify and extract image sections within documents. Integrate OCR to convert these images into machine-readable text.
3. **Preprocessing:** Implement preprocessing steps to prepare the converted document text for optimal LLM performance. This may include:
   - Text cleaning: Removing irrelevant characters, whitespace, and formatting.
   - Sentence segmentation: Dividing the text into individual sentences.
   - Tokenization: Breaking down sentences into individual words or sub-word units.

Deliverables:

- Python scripts or modules for document conversion, OCR, and preprocessing.
- Documentation detailing your approach, including the libraries used for conversion, the specific preprocessing steps implemented, and your OCR engine selection and evaluation process.

Part 2: LLM-Powered Understanding and Actions

Task:

1. **LLM Integration:** Integrate a Large Language Model into the pipeline to enable advanced document understanding. Your choice of LLM should be justified based on its capabilities and suitability for the tasks outlined below. Consider factors like model size, training data, and potential for customization or fine-tuning.
2. **Information Extraction:** Design and implement an LLM system to extract essential information from documents. This could include:
    - Identifying and extracting entities like names, dates, locations, and organizations.
    - Extracting relationships between entities.
    - Summarizing key information from the document.
3. **Document Classification:** Develop a mechanism using the LLM to classify documents into predefined categories based on their content.
4. **Internal Translation:** Implement a feature using the LLM to translate the text within the documents into different languages.

Deliverables:

- Python scripts or modules for LLM integration, information extraction, document classification, and internal translation.
- Detailed documentation outlining your chosen LLM, its integration with the pipeline, and the techniques employed for information extraction, classification, and translation.

Part 3: User Interaction via a Chatbot Interface (Optional/Additional)

Task:

1. **Chatbot UI:** Create a user-friendly chatbot interface (e.g., using Streamlit) that allows users to interact with the LLM and processed documents. The chatbot should enable users to:
    - Upload documents for processing.
    - Ask questions about the document's content.
    - Request specific information to be extracted.
    - Review extracted data and provide feedback.
2. **Pipeline Integration:** Ensure smooth integration between the chatbot interface, LLM, and document processing components.

Deliverables:

- Source code for the chatbot UI and its integration with the document processing pipeline.
- A README file containing setup instructions, instructions for interacting with the chatbot, and a clear explanation of the design choices and technologies used.

Technical Requirements:

- Choose appropriate Python libraries and frameworks for document processing, OCR, LLM integration, and chatbot development.
- Ensure code readability, proper documentation, and efficient error handling.

- Host your code in a public GitHub repository.

Evaluation Criteria:

- Understanding and implementation of LLM-powered document understanding techniques.
- Ability to design and implement a robust and adaptable document processing pipeline.
- Quality and user-friendliness of the chatbot interface.
- Clarity and completeness of documentation.
- Code quality, including readability, structure, and comments.

Submission Details:

- GitHub repository link containing your code and documentation.
- A video demonstration of your pipeline and chatbot UI, showcasing its features and functionalities.

Deadline:

Please submit your assignment within 72 hours of receiving this task.

We are excited to see your innovative solutions and how you approach the challenges of building an intelligent and adaptable document processing pipeline powered by Large Language Models!

> Please submit your assignments in the below given link:
> https://forms.gle/gYKvky81RgwEeGZT7