Project Report

Overview

This invoice data extraction system consists of three key components:

PDF to Text Conversion

For normal PDFs (not scanned or hybrid with images), libraries like PyPDF2 are sufficient to extract text. For scanned PDFs, Optical Character Recognition (OCR) is required to extract text, necessitating the use of libraries like Tesseract.

Parsing Extracted Text

This is the most crucial step. The extracted text must be parsed to obtain the invoice details and form meaningful, structured data. Libraries such as Regex can be utilized for this purpose.

Accuracy Metrics and Trust Determination

This step involves comparing the extracted information with actual data. It determines the number of fields that the system has correctly identified in comparison to the actual data.

My Approach

The system begins by extracting text from PDF files using PyPDF2.

The extracted text is then parsed using Regex to retrieve:

- Recipient information
- Invoice details
- Customer details
- Item descriptions
- Total amount
- Payment information

This parsed output is organized in a YAML file format.

Regex leverages patterns found in all the invoice PDFs located in the test data folder.

Justification

Since all provided invoice documents are regular (not scanned or containing images) and follow a consistent pattern, a straightforward approach is adequate for data extraction. This method capitalizes on the regularity and simplicity of the data. Although simple, this approach is scalable, efficient, and cost-effective.

Hybrid Approach

If the invoice is in a scanned PDF format, the system should utilize OCR libraries such as **Tesseract** for text extraction. When there is a visible pattern in the extracted text or the invoice itself, the same **Regex** methods can be applied to extract meaningful data and structure it effectively.

Hybrid approaches combine both regular and OCR-based methods for text extraction. However, using OCR can introduce some errors. The confidence scores provided by the OCR can be leveraged to assess the reliability of the extracted information. This method may also result in a slight decline in accuracy compared to data extraction from regular PDF documents.

Necessity of GenAI and Drawbacks

When automating the system for various types of invoices—whether regular, scanned, or in other formats without visible patterns—extensive automation and dynamic decision-making are required. This can be achieved through the use of **Large Language Models** (LLMs) such as **GPT-4, 4o, Gemini** and open-source models like **LLama** models to obtain the desired results.

How to use GenAI in this context?

Multimodal LLMs can be employed to upload PDF documents and return the extracted data in the preferred format (either **JSON** or **YAML**).

To achieve the desired accuracy, effective prompt engineering is essential, which includes supplying a few examples of the expected output to the model (known as **Few-shot Learning**).

The accuracy of the response can be assessed by comparing the extracted information with the actual data (labeled data).

The automation needed for text extraction and parsing will be managed by the employed LLM.

Reliability Issues

GenAI-based approaches encounter various reliability issues that can affect their application in real-world scenarios where accurate information is critical. The model may produce hallucinations in its responses, and there is a risk that it could output irrelevant information related to the uploaded invoice PDF.

This concern is particularly significant in situations where trust determination is paramount, especially regarding whether the extracted information can be relied upon for downstream tasks.

Accuracy Score and Trust Determination

To assess the accuracy of the system, actual data is needed as a benchmark dataset to evaluate the performance of various implementations.

However, in addition to accuracy checks for trust determination, cross-validation methods—such as verifying the total amount against itemized information and detecting anomalies—can be employed. If the reliability of the output falls below a certain threshold, the invoice can be flagged for manual review.

Scalability and Cost-Effectiveness

My approach

This method uses PyPDF2 for text extraction and Regex for parsing, making it highly scalable and low-cost. It requires minimal computational resources and can efficiently handle increased invoice volumes due to the predictable nature of regular PDF formats. However, it cannot be used for scanned PDF or invoices with varying patterns.

Hybrid Approach

The hybrid approach incorporates OCR, like Tesseract, for scanned PDFs alongside traditional methods for regular invoices. While this broadens the system's applicability, it increases complexity and costs, as additional libraries and resources are needed. Despite potential inaccuracies, it remains scalable for various invoice types but may incur higher operational costs.

GenAI-Based Approach

The GenAI-based method employs Large Language Models (LLMs) to process diverse invoice formats. Although it offers advanced capabilities and adaptability, this approach involves higher costs for model inference and computational resources. While it enhances scalability by automating decision-making, reliability issues such as hallucinations can impact trustworthiness and require careful validation.