# Data Science Assignment: Invoice Data Extraction

Ramesh Raj S (CH20B085)

October 20, 2024

This document provides a comprehensive technical overview of the approach, algorithms, and performance analysis for the Invoice Data Extraction system.

## 1 Technical Approach and Algorithms

### 1.1 PDF to Image Conversion Process

This process converts PDF files into images for OCR using the **PyMuPDF library**. A target resolution of **300 DPI** is used to ensure high-quality images for OCR, achieved by applying a zoom factor to the default **72 DPI** provided by PyMuPDF.

### 1.2 OCR Processing of Images

The system performs Optical Character Recognition (OCR) on images using **PaddleOCR** with angle classification enabled and set to process English-language text.

### 1.3 JSON Extraction from Invoice Text

This process uses a generative model **Gemini 1.5 Flash** to convert **semi-structured** extracted invoice text into a **structured JSON**. The model's **temperature** is set to **0.01** to ensure that the generated outputs are more deterministic and focused, minimizing randomness in responses.

The model processes each invoice's text and returns structured data, including fields such as invoice information, supplier and customer details, items, and payment details.

### 1.4 Accuracy Check Implementation

The accuracy check implementation compares extracted JSON with a **manually curated true JSON**. Both are nested, necessitating a comprehensive comparison approach. The core functionality of the accuracy check involves **recursively comparing the true JSON with the extracted JSON**. This recursive comparison allows for the evaluation of data at multiple levels of nesting, ensuring that all elements are adequately assessed. At the leaf nodes of the JSON trees, where the values are strings, we employ the **fuzzywuzzy library**. This library allows us to calculate a similarity score between the strings.

As a result of this comparison, a **match JSON** is generated, which retains the structure of the original JSON while providing **accuracy scores** that reflect the degree of similarity between the corresponding elements.

### 1.5 Calculating Average Accuracy Scores for Invoice Attributes

We calculate the average accuracy scores for various invoice attributes using the match JSONs generated from comparing extracted and true JSON data for all test data.

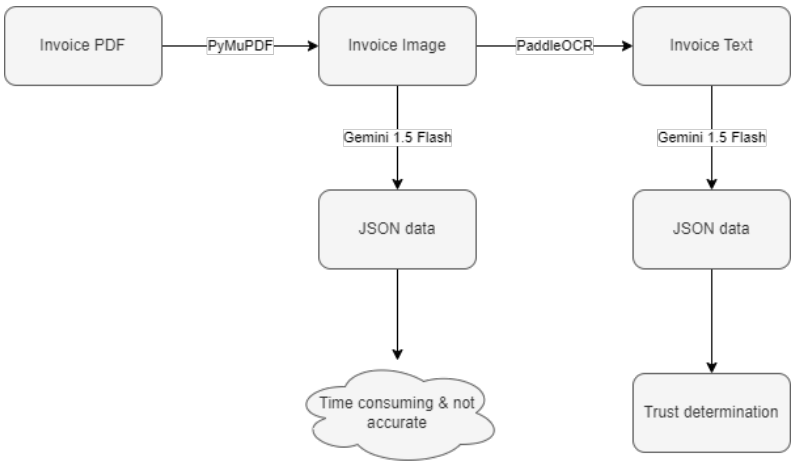


Figure 1: Overview of the System.

The process begins by iterating through the match JSONs for all test entries to compute the average accuracy scores. During this iteration, we identify the accuracy scores associated with each invoice attribute. These scores are accumulated for each attribute across all entries in the match JSONs.

Once the accuracy scores for each attribute are collected, the next step involves calculating the **average score**. This is achieved by summing the individual accuracy scores for each attribute and dividing by the total number of scores collected.

Additionally, we calculate the **overall average accuracy across all attributes** by summing the average scores of all individual attributes and dividing by the total number of attributes. This final total accuracy provides a comprehensive metric of performance for the entire data extraction process, allowing for an assessment of its overall effectiveness.

## 1.6 Trust Determination for Invoice Data

The trust determination process assesses the integrity and reliability of invoice data by performing a series of validation checks on its attributes.

The validation process comprises several key checks:

1. **Quantity:** Each item's quantity must be greater than zero.

2. **Taxable Value:** The taxable value is computed as the product of the rate per item and quantity, compared against the provided taxable value with a tolerance.

3. **Tax Amount:** The tax amount is calculated from the taxable value and tax percentage, then compared to the provided tax amount, considering a tolerance.

4. **Total Amount:** The total amount is verified as the sum of the taxable value and tax amount, ensuring it aligns with the provided total.

5. **Total Taxable Amount:** The total taxable amount is verified as the sum of the taxable values, ensuring it aligns with the provided total.

6. **Invoice Dates:** The invoice date must be earlier than or equal to the due date and not set in the future.

Upon completion of these checks, the function outputs whether the invoice data is trusted. If any discrepancies are found during the validation process, appropriate messages are generated, highlighting the specific issues identified.

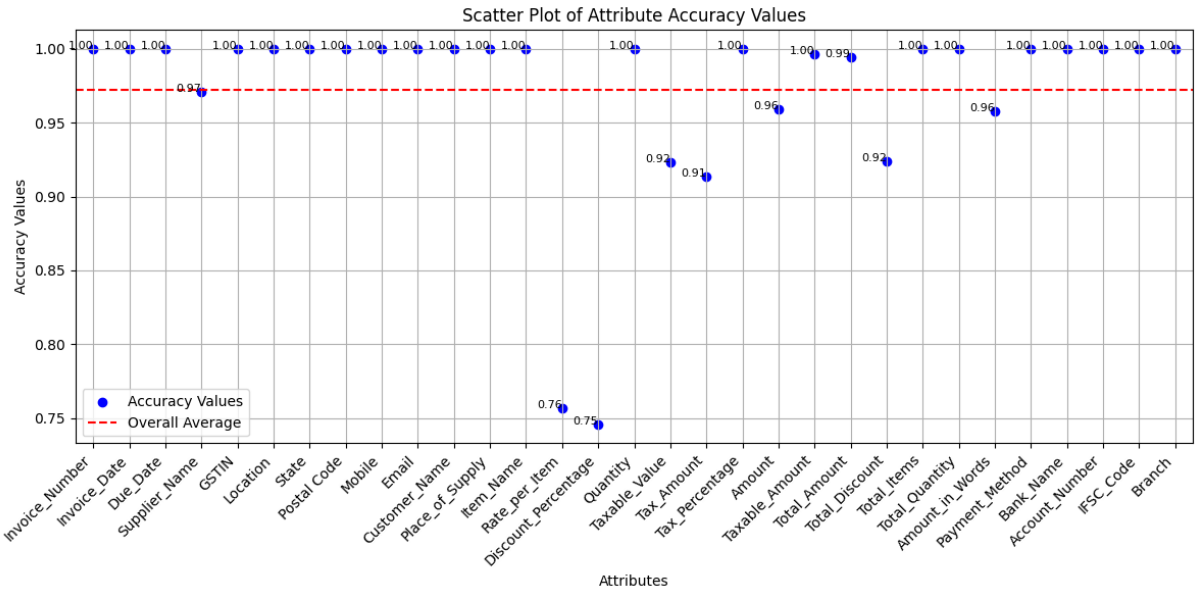


Figure 2: Scatter Plot Attribute Accuracy Values.

# 2 Justification for Chosen Methods

- **PaddleOCR**:
  - **High Accuracy:** Utilizes advanced deep learning techniques for superior text recognition accuracy across various languages and fonts.
  - **Open Source:** Customizable and flexible, allowing fine-tuning without licensing costs.
  - **Multilingual Support:** Supports multiple languages, making it suitable for international documents like invoices.
- **Gemini 1.5 Flash**:
  - **Structured Data Generation:** Excels in converting unstructured text, such as invoice details, into well-structured JSON formats.
  - **High Performance:** Offers rapid processing speeds, essential for handling large volumes of invoice data efficiently.
- **FuzzyWuzzy**:
  - **Context Insensitivity:** In the invoice data validation process, the exact context of words is less critical, allowing for the use of statistical scores for comparison.
  - **Statistical Scoring:** FuzzyWuzzy provides efficient string matching based on edit distance, measuring the number of single-character edits required to change one string into another.

# 3 Accuracy Report

## 3.1 Accuracy Breakdown by Invoice Type and Field

The following table provides a detailed breakdown of system accuracy by invoice type and key data fields:

**Overall Average Accuracy:** 97.23%

| Invoice Type | Field | Accuracy (%) |
|---|---|---|
| Invoice Info | Invoice Number | 100.0 |
| | Invoice Date | 100.0 |
| | Due Date | 100.0 |
| | Tax Percentage | 100.0 |
| | Total Amount | 99.42 |
| Supplier Info | Supplier Name | 97.08 |
| | GSTIN | 100.0 |
| | Location | 99.96 |
| | State | 100.0 |
| | Mobile | 100.0 |
| | Email | 100.0 |
| Customer Details | Customer Name | 100.0 |
| Item Details | Item Name | 99.98 |
| | Rate per Item | 75.65 |
| | Discount Percentage | 74.56 |
| | Quantity | 100.0 |
| | Taxable Value | 92.30 |
| | Tax Amount | 91.33 |
| Payment Details | Payment Method | 100.0 |
| | Bank Name | 100.0 |
| | Account Number | 100.0 |
| | IFSC Code | 100.0 |
| | Branch | 100.0 |

Table 1: Accuracy Breakdown by Invoice Type and Field

# 4 Performance Analysis

A total of **24** test PDFs were evaluated. The performance analysis for each stage of processing is as follows:

- **PyMuPDF:**
  - Time taken to convert PDFs to images: **8.01** seconds
  - Average time per PDF: **0.33** seconds
- **PaddleOCR:**
  - Time taken to perform OCR on images: **13.66** seconds
  - Average time per image: **0.57** seconds
- **Gemini 1.5 Flash:**
  - Total time for JSON generation: **159.33** seconds
  - Average time per invoice: **6.64** seconds
- **Overall Inference Time:**
  - Total time for inference per PDF: **4.17** seconds, which includes converting to image, performing OCR, generating JSON, and trust determination.

# 5 Comparison of Approaches

Several approaches were tested during the development process:

- **PaddleOCR vs. Tesseract:** PaddleOCR performed significantly better than Tesseract in terms of accuracy and processing time.
- **Direct Image Input to Gemini 1.5 Flash:** Feeding images directly into the Gemini 1.5 Flash resulted in JSON output; however, this method was less accurate compared to using PaddleOCR. Additionally, the total time for JSON generation from images was **212.45** seconds, with an average time per image of **8.85** seconds.

# 6 Conclusion

This report documents the approach, algorithms, and performance assessment of the Invoice Data Extraction system. The combination of **PaddleOCR and Gemini 1.5 Flash** ensures a high degree of accuracy and trustworthiness in data extraction, making the system reliable for large-scale use cases.