Invoice Data Extraction: Project Report

- **Objective:**
  The goal of this project is to develop a solution for extracting data from invoices in PDF format. The invoices may include regular PDFs, scanned documents, and PDFs with both text and images. We aim for an accuracy rate of over 90%.

Project Details:

- **Data Extraction**:
  - The code supports data extraction from images, PDFs and folders containing both.
  - **PyPDF** is used for PDF extraction and **Pytesseract** is used for image extraction.
  - **Llama-Extraction**, an AI agentic tool gives the best accuracy for extraction from PDFs.

- **Accuracy Check and Trust Determination**:
  - Accuracy is calculated based on the number of attributes extracted v/s the total number of attributes present in the PDF.
  - The attributes were decided manually, meaning the trust-worthiness.

- **Cost-Effectiveness vs. Accuracy**:
  - Even though PyPDF extraction was faster, there was a need to use Regular Expressions to parse the data from the extracted text, leading to lesser accuracy, whereas Llama-Extract was computationally expensive, but gave an accuracy of more than 90%.

- **Performance Metrics**:
  - The accuracy has been explained below. Attributes like Round_off, CGST and SGST were not retrieved in certain cases, whereas using Llama Extract, all the other items were extracted with an accuracy of **100%.**

- **Scalability and Efficiency**:
  - The code can handle a large amount of data and is scalable.
  - To optimise the speed, the data will be splitted into image files and pdf files initially and then processed together, reducing the computation time.

- **Error Handling and Reporting**:
  - A single folder can handle '.pdf', '.jpg', '.png', and '.jpeg', files at the same time.
  - The code fails at certain times, giving the error: Key-error:"Round_off" in case of Llama-Extract.

Implementation Details

The main data extraction script (`model.py`) was developed to extract relevant information from the invoices. It uses the following libraries:

- **PyPDF2**: For extracting text from regular PDFs.
- **Llama-Extract**: AI-method to extract data from PDFs.
- **pytesseract**: For Optical Character Recognition (OCR) on scanned documents.

The script allows users to specify the target and save folders as arguments. The correct way to run the code is:

python model.py –pdf_directory 'pdf_dir' –save_directory 'sav_dir' –use_llama

- **use_llama**: This Argument is used only if Llama Extract is needed. Otherwise, the default option is PyPDF.
- These are the three arguments supported while running the code in the command line.

If `use_llama` is not specified, it will default to using PyPDF2 for extraction.

The code is scalable and can handle any number of files without needing an API, making it a cost-free solution.

It also supports multiple arguments and can process folders containing image files (.jpg, .jpeg, etc.) and scanned documents.

Perform the below installations:

- pip install PyPDF2 pdfplumber pytesseract pandas pytz
- sudo apt-get install tesseract-ocr (For Ubuntu users)
- # For Windows, download and install from https://github.com/UB-Mannheim/tesseract/wiki
- pip install pandas
- pip install llama-extractor
- pip install pydantic

Run the requirements.txt file given with the code as given below:
Pip insall -r requirements.txt to install the required dependencies.

Create a Conda Environment before installing(Recommended):
conda env –name 'name of environment'

Performance analysis:

- From the results obtained, by manual inspection, the obtained results were 100% accurate for PDFs for almost all the attributes, using LLama_Extract.
- The code is scalable to any number of pdfs and texts and can be processed at the same time.
- The extracted data is saved into a **.json** file and the code supports specifying the save-directory in the command line itself.
- The code can support the change in format of Invoice since Llama-extractor is used, which is an AI-based Agentic parser, but in case of images, there should be updates in the Regex Format, which is a limitation of this code.
- Llama-Extractor itself can be used for this, but it uses a paid Google OCR binding on top.
- The code uses a blend of manual and AI-approach towards solving the problem.

Accuracy Calculation:

Here, a method is adopted exclusively for the given invoices:
The total number of important attributes were calculated:

1. Company Name
2. Company GSTIN
3. Company Address
4. Company Mobile
5. Company Email
6. Invoice Number
7. Invoice Date
8. Due Date
9. Customer Name
10. Customer Shipping Address
11. Customer Phone
12. Place of Supply
13. Item : Description
14. Item : Rate / Item
15. Item : Discounted Price
16. Item : Quantity
17. Item : Taxable Value
18. Item : Tax Amount
19. Item : Total Amount
20. Taxable Amount
21. CGST
22. SGST
23. Round off
24. Taxable Amount
25. Total (Final Amount)
26. Total Discount
27. Total Amount in Words
28. Bank Details

Using Llama Extract from PDfs:

Given below is one example extraction using Llama Extract from the PDFs:

Company_Name: "UNCUE DERMACARE PRIVATE LIMITED"
GSTIN: "23AADCU2395N1ZY"
Company_Address: "C/o KARUNA GUPTA KURELE, 1st Floor S.P Bungalow Ke Pichhe, Shoagpur Shahdol, Shahdol Shahdol, MADHYA PRADESH, 484001"
Company_Phone: "+91 8585960963"
Company_Email: "ruhi@dermaq.in"
Invoice_Number: "INV-138"
Invoice_Date: "06 Mar 2024"
Due_Date: "06 Mar 2024"
Customer_Name: "Agrani Kandele"
Customer_Phone: "8120482988"
Shipping_address: "vadandana beauty parlour Murawara Katni, MADHYA PRADESH, 483501"
Place_of_supply: "23-MADHYA PRADESH"

Items:

1. **Item_Number**: "1"
   Item_Name: "Cetaphil Gentle Cleansing Lotion - Oily Skin 125 ml"
   Rate_per_Item: 415.68
   Quantity: 1
   Taxable_Value: 415.68
   Tax_Amount: 74.82
   Total_Amount: 490.50
2. **Item_Number**: "2"
   Item_Name: "Solasafe sunscreen gel spf 50"
   Rate_per_Item: 480.68
   Quantity: 1
   Taxable_Value: 480.68
   Tax_Amount: 86.52
   Total_Amount: 567.20
3. **Item_Number**: "3"
   Item_Name: "IPCA Acne OC Moisturizer Cream - 75 gm"
   Rate_per_Item: 378.98
   Quantity: 1
   Taxable_Value: 378.98
   Tax_Amount: 68.22
   Total_Amount: 447.20

Totals:

- **Taxable_Amount**: 1275.34
- **Total_Amount**: 1505
- **Total_Discount**: 308.1
- **Total_amount_in_words**: "INR One Thousand, Five Hundred And Five Rupees Only."

Payment_Details:

- **Bank**: "Kotak Mahindra Bank"
- **Account_Number**: "1146860541"
- **IFSC_Code**: "kkbk0000725"
- **Authorized_Signatory**: "UnCue Dermacare Pvt Ltd"

The attributes that are not extracted from the PDF: CGST, SGST, Round_Off.
Accuracy = {(Number of correct retrievals)/(Total number of attributes)}*100
Here, Accuracy = {25/28}*100 = 89.28%

On an average for the 24 PDFs, Round_off was getting parsed for many items, which adds up to a best case average of >= 90%!

Except these three attributes, all the other attributes were giving 100% accuracy for all the files.

EXTRACTING DATA USING PYPDF:

1. **Company**: "UNCUE DERMACARE PRIVATE LIMITED"
2. **Company_GSTIN**: "23AADCU2395N1ZY"
3. **Company_Address**: "C/o KARUNA GUPTA KURELE, 1st Floor S.P Bungalow Ke Pichhe, Shoagpur Shahdol, Shahdol Shahdol, MADHYA PRADESH, 484001"
4. **Company_Mobile**: "91 8585960963"
5. **Company_Email**: "ruhi@dermaq.in"
6. **Invoice_Number**: "INV-138"
7. **Invoice_Date**: "06 Mar 2024"
8. **Due_Date**: "06 Mar 2024"
9. **Customer_Name**: "Agrani Kandele"
10. **Customer_Phone**: "8120482988"
11. **Customer_Shipping/Billing_Address**: "vadandana beauty parlour Murawara Katni, MADHYA PRADESH, 483501"
12. **Place_of_Supply**: "23-MADHYA PRADESH"
13. I**tem_List**: "\n1 Cetaphil Gentle Cleansing Lotion - Oily Skin 125 ml 415.68\n461.86 (-10%) 1 BTL 415.68 74.82 (18%) 490.50\n2 Solasafe sunscreen gel spf 50 480.68\n600.85 (-20%) 1480.68 86.52 (18%) 567.20\n3 IPCA Acne OC Moisturizer Cream - 75 gm 378.98\n473.73 (-20%) 1378.98 68.22 (18%) 447.20\n"
14. **Total_Amount**: "1,505.00"
15. **Total_Discount:** "308.10"
16. **Bank_Details**: "Kotak Mahindra Bank"

17. **Account#**: "1146860541"
18. **IFSC_Code**: "kkbk0000725"
19. **Branch**: "PUNE - CHINCHWAD"

Since a Regular Expression had to be written to extract the data in this case, the important attributes were selected and the results gave an 100% accuracy for all the selected attributes.

Github Link: https://github.com/ananthu2014/PDF-Extraction

References:

1. https://ieeexplore.ieee.org/document/7991564
2. https://www.llamaindex.ai/blog/mastering-pdfs-extracting-sections-headings-paragraphs-and-tables-with-cutting-edge-parser-faea18870125
3. https://dev.to/wmisingo/create-the-fastest-and-precise-invoice-data-extractor-for-structural-output-using-ai-pe1
4. https://pypi.org/project/pytesseract/

Tools used: VS-Code, Python and libraries, GPT-4(for syntax, and doubts)