

Document image understanding and information extraction tasks

tan chay yee

*** COPY ***

OJC MARKETING SDN BHD

ROC NO: 538358-H

NO 2 & 4, JALAN BAYU 4,

BANDAR SERI ALAM,

81750 MASAI, JOHOR

Tel:07-388 2218 Fax:07-388 8218

Email: ng@ojcgroup.com

TAX INVOICE

Invoice No : PEGIV-1030765
Date : 15/01/2019 11:05:16 AM
Cashier : NG CHUAN MIN
Sales Person : FATIN
Bill To : **THE PEAK QUARRY WORKS**
Address : .

Description	Qty	Price	Amount
000000111	1	193.00	193.00 SR

KINGS SAFETY SHOES KWD 805

Qty: 1	Total Exclude GST:	193.00
	Total GST @6%:	0.00
	Total Inclusive GST:	193.00
	Round Amt:	0.00

TOTAL: 193.00

VISA CARD 193.00

xxxxxxxxxxxx4318

Approval Code:000

193.00

Goods Sold Are Not Returnable & Refundable

****Thank You. Please Come Again.****

Marie-Noël Lepage
April 6, 2023

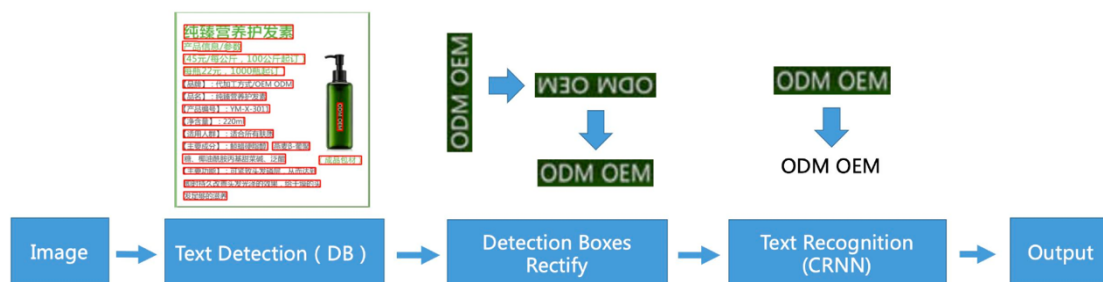
Problem Statement

Business documents (purchase orders, financial reports, letters, resume, invoice, ID cards, etc.) are critical to a company's efficiency and productivity. Understanding documents is a very challenging task due to the diversity of layout and formats, poor quality of scanned document image as well as the complexity of template structure but the information is usually presented in natural language. Many companies extract data from business documents through manual efforts that are time-consuming and expensive. To address these problems, using advanced deep learning are designed to automatically classify, extract, and structuralize information from documents.

Data Sources and Technique Used

The dataset used for this project was created for the ICDAR 2019 Robust Reading Challenge on Scanned Receipts OCR and Information Extraction (SROIE).¹ The dataset contains 973 scanned receipt written in English. It contains the pre-trained base model to extract information (for each receipt you have an .jpg file of the scanned receipt (image), a .txt file holding OCR information (box) and a .txt file holding the key information values (entities).

In case that our data are not pre-trained, I explore the machine learning model PaddleOCR² toolkits (PARallel Distributed Deep LEarning) that provide data annotation and synthesis tools support in more that 80 languages recognition. There are three major processing steps: character position detection, recognition of character orientation and correction character content recognition.



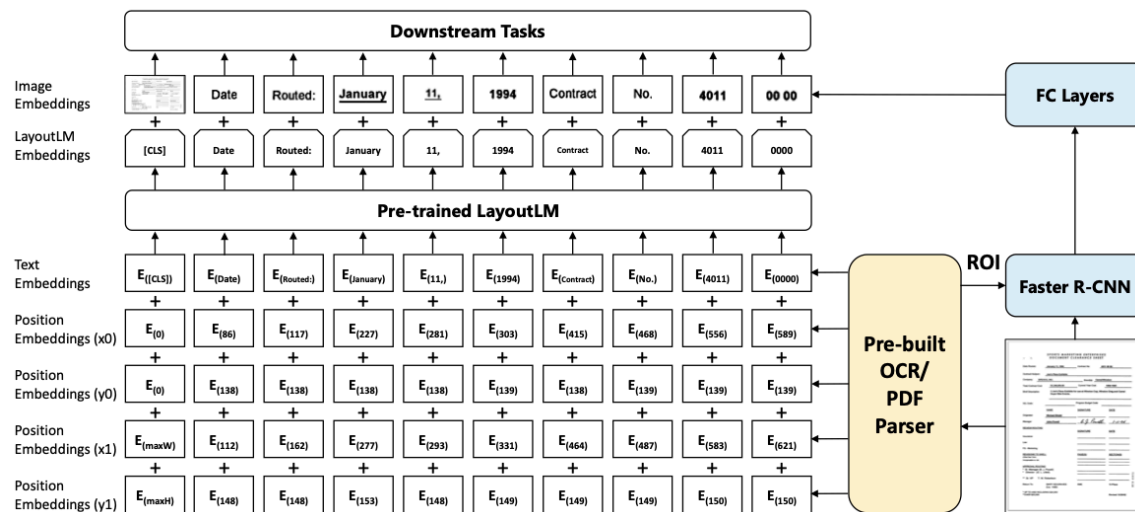
For assign labels correctly, I use the Hugging Face LayoutLM³ model. It was one of the first models that archived success combining the pieces to create a singular model that performs labelling using positional information, text-based information, and image information. Inspired by the BERT (attention-based bidirectional language modelling approach), where input textual information is mainly represented by text embedding and position embeddings, LayouLM adds two types of input embeddings:

¹ <https://rrc.cvc.uab.es/?ch=13>

² <https://github.com/PaddlePaddle/PaddleOCR>

³ https://huggingface.co/docs/transformers/model_doc/layoutlm
<https://arxiv.org/abs/1912.13318>

1. **Embedding 2-D position** features into the language representation can improve the alignment between layout information and semantic representation by utilizing the self-attention mechanism within the transformer.
2. By combining the **image embedding** with traditional text representation, documents can have richer semantic representations that capture appearance features such as font, types, colors, and direction.



Preprocessing the Dataset and Fine Tune LayoutLM

Annotation of document:

The first thing we do is to extract the text-based information from the document and find their respective locations of recognize text for training our model. By location, I refer to something called a bounding box (a rectangle that encapsulates the piece of text on the invoice). The entities file represents the label I chose associate with the text recognition. With help of PaddleOCR, I can easily create the box and entities text files as the same format of the SROIE dataset.

(x ₀ =166, y ₀ =317)	(x ₁ =419, y ₁ =317)	Fat Bar - Fatburger
(x ₃ =166, y ₃ =351)	(x ₂ =419, y ₂ =351)	3763 Las Vegas BLVD Las Vegas, NV 89109 702-736-4733
Server: Irma	12/03/2019	
Table 60/1	9:54 PM	
Guests: 1	20060	
XXL FAT	15.50	
ADD CHEESE	1.10	
only lettuce		
only tomato		
CHILI CHEESE FAT FRIES	6.35	
BUD LIGHT	7.00	
Subtotal	29.95	
Tax	2.47	
Total	32.42	

Box Information:

```
!head -n 10 /content/fatburger_box.txt
166.0,317.0,419.0,317.0,419.0,351.0,166.0,351.0,Fat Bar - Fatburger
164.0,351.0,419.0,351.0,419.0,382.0,164.0,382.0,3763 Las Vegas BLVD
164.0,385.0,422.0,385.0,422.0,416.0,164.0,416.0,Las Vegas, NV 89109
219.0,421.0,378.0,421.0,378.0,448.0,219.0,448.0,702-736-4733
13.0,479.0,184.0,484.0,182.0,518.0,12.0,513.0,Server: Irma
446.0,486.0,578.0,486.0,578.0,513.0,446.0,513.0,12/03/2019
17.0,515.0,157.0,515.0,157.0,549.0,17.0,549.0,Table 60/1
480.0,515.0,583.0,515.0,583.0,549.0,480.0,549.0,9:54 PM
14.0,549.0,145.0,549.0,145.0,583.0,14.0,583.0,Guests: 1
506.0,544.0,584.0,550.0,581.0,586.0,503.0,580.0,20060
```

Entities Information:

```
!head -n 10 /content/fatburger_entities.json
{
  "company": "Fat Bar - Fatburger",
  "date": "12/03/2019",
  "address": "3763 Las Vegas BLVD, Las Vegas, NV 89109",
  "total": "35.66"
}
```

Data Preprocessing:

The dataset needs to be formatted in some way in files before the training and testing step. From the annotation step, I get my box file (text lines) and entities file (text labels). I need to reorganize them by assembling each line of text with its associated labels. Since all text must be labeled, we create an 'O' label for the undefined text. I match the label with the text line by checking the similarity according to a match higher than 80%. In addition, I assign total and date to only largest bounding boxes for no duplicate labels (a receipt could have only one item on it). Finally, I divide the lines of text into separate words (based on word length) with their own bounding boxes.

- **train.txt / test.txt:** This is the information that our model tries to predict. We use this file in the LayoutLM for calculate our performance metric on the model.

test.txt [word, label]

```
1 SIN S-COMPANY
2 LIANHAP S-COMPANY
3 SDN S-COMPANY
4 BHD S-COMPANY
5 LOT S-ADDRESS
```

labels.txt ×

```
1 S-COMPANY
2 S-DATE
3 S-ADDRESS
4 S-TOTAL
5 O
```

- **labels.txt:** Define the label using 'S-format'. All words must be labeled, and the 'O' represent each words not labeled in the annotation step.

test_box.txt [word, x₀, y₀, x₂, y₂]

```
1 SIN 279 185 354 200
2 LIANHAP 362 185 522 200
3 SDN 531 185 605 200
4 BHD 613 185 688 200
5 LOT 301 202 370 215
```

← Box normalized

```
x0 = int(1000 * (x0 / width))
x2 = int(1000 * (x2 / width))
y0 = int(1000 * (y0 / height))
y2 = int(1000 * (y2 / height))
```

- **train_box.txt / test_box.txt:** Bonding box should be in (x₀, y₀, x₂, y₂) format by word and normalized to scale of 0-1000.

test_image.txt [word, x₀, y₀, x₂, y₂, width, height, filename]

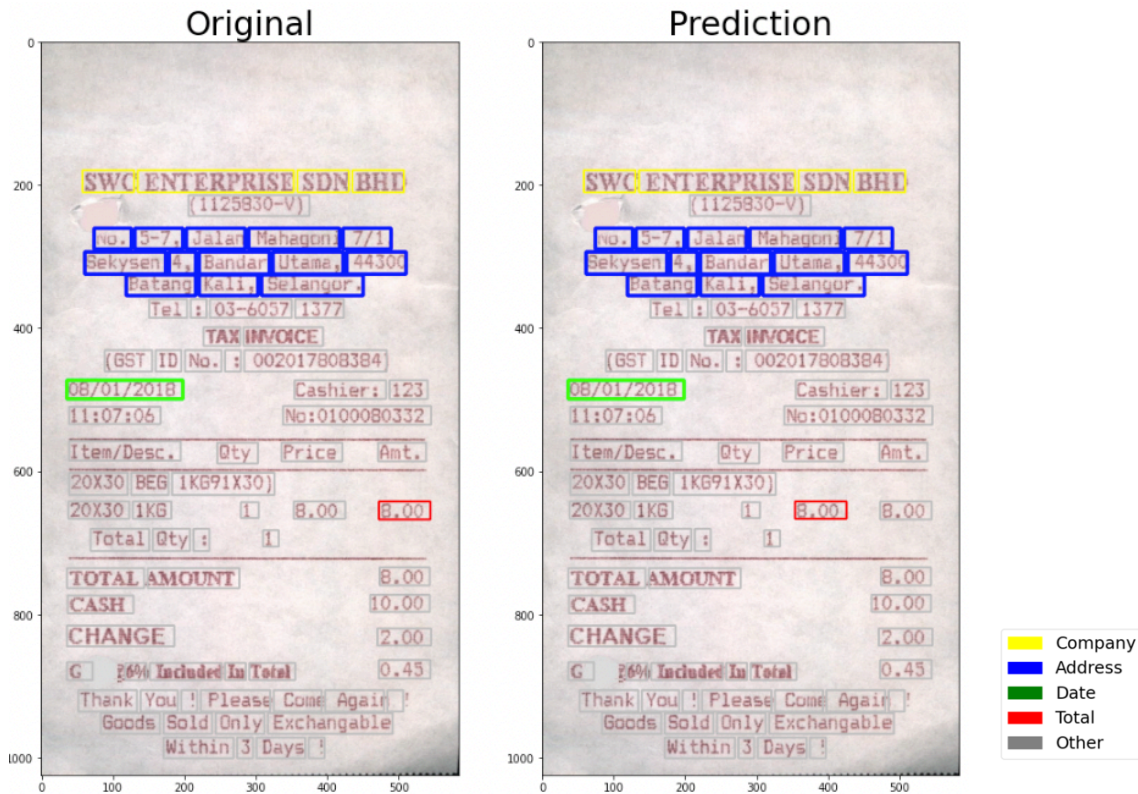
```
1 SIN 166 329 210 355 593 1769 X51005230621
2 LIANHAP 215 329 310 355 593 1769 X51005230621
3 SDN 315 329 359 355 593 1769 X51005230621
4 BHD 364 329 408 355 593 1769 X51005230621
5 LOT 179 358 220 382 593 1769 X51005230621
```

- **train_image.txt / test_image.txt:** All information that contain our document (word, bounding box, weight & height of the invoice and the filename).

Fine Tune LayoutLM:

As LayoutLM is a pre-trained model one won't need a large dataset to further train it. But it depends on the visual structure of your document. If you want to target a wide range of receipt formats you would need at least 200 images per targeted format to give you decent if not good performance. In my case, I have 626 invoices for the training step. I clone the LayoutLM GitHub project which contains the script and use the training and testing command related. These are the results after the training and testing on SROIE Dataset.

Metric	Score (%)	Definition
Loss	9.78	How many of the predictions may be not correct.
Precision	91.20	Ratio between the True Positives and all the Positives.
Recall	96.17	Ability of the classifier to correctly identifying True Positives.
F1-Score	93.62	Combining precision and recall. Described as the harmonic mean.



```
cat output/test_predictions.txt -n | head -n 504 | tail -n 10
```

```
495 SWC S-COMPANY
496 ENTERPRISE S-COMPANY
497 SDN S-COMPANY
498 BHD S-COMPANY
499 (1125830-V) O
500 NO. S-ADDRESS
501 5-7, S-ADDRESS
502 JALAN S-ADDRESS
503 MAHAGONI S-ADDRESS
504 7/1, S-ADDRESS
```

The model return a text file on the prediction of label for each word in invoices.

Conclusion

The model gives a decent prediction and shows how powerful LayoutLM is. To improve accuracy, it's possible to adjust hyperparameter or train on more data depending on your needs. To extract further information, you would need to create appropriate labels for each item row using the 'S-format'. You would also need to create custom training data consisting of your newly created labels. Notes about the project:

- You can use other tools for annotation. As example, Tesseract, EasyOCR, UBAI, AWS Textract or Google Cloud Vision. Label classification is done at the word level, so it is up to the OCR text extraction engine to ensure all words in a field are in a continuous sequence.
- Microsoft has released two news transformer model and should be improve in performance. The version 2 uses the detectron library to enable visual feature embeddings as well. The version 3 is the multi-modal transformer architecture that combines text and image embedding in a unified way. The main advantage of this approach is the reduction in parameters needed and overall lower computation.