Quantum Assessment

Maxime Marteau

Image processing algorithm - La Cuentax

- La Cuentax is a mobile app that makes it easy to split the total amount of a bar or restaurant bill
- Developed an image processing algorithm to extract structured elements from an image
- Leveraging ML models to improve the end-to-end algorithm accuracy

La Cuentax

How does La Cuentax works

1. Take a photo of the bill, the app analyzes the photo and extracts all the elements
2. Add people from your table
3. Select what each person has eaten and drunk, it is possible to share a dish between several people
4. The app displays the total per person, everyone can now pay or refund their share

Image processing algorithm

High level explanation of the current image processing algorithm


```
"items":[
  {
    "quantity":1,
    "name":"COCA ZERO",
    "price":250
  },
  {
    "quantity":1,
    "name":"COCA COLA",
    "price":250
  },
  {
    "quantity":1,
    "name":"PULLED PORK Burger",
    "price":1250
  },
  {
    "quantity":1,
    "name":"ROCKEFELLER 250g",
    "price":1250
  },
  {
    "quantity":1,
    "name":"SOHO 250g",
    "price":1250
  }
]
```

High level explanation of the image processing algorithm

1. Run OCR algorithm to extract images texts and boxes
2. Clean texts to remove noise and unused characters
3. Regroup boxes per lines
4. Identify the lines containing the bill items
5. Convert the text lines to structured list

Improving the algorithm through an iterative process

Prerequisite:

- Label images using a semi automatic process
- Create a scoring function

Identify common issues and improve the algorithm

Check the improvements and the potential negative impacts

Leveraging LayoutLMv3 model

Introduction of the LayoutLMv3 model

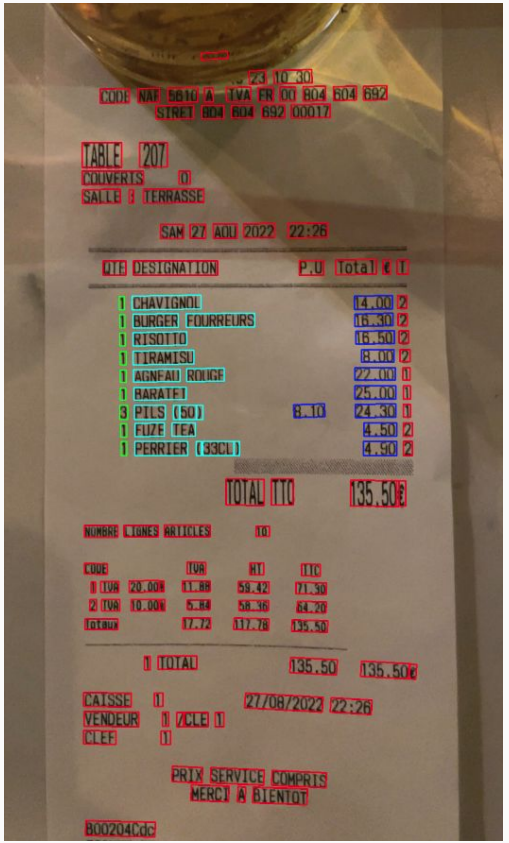
- A transformer-based model by Microsoft for Document AI
- Integrates text, layout, and visual (image) information through a multimodal encoder
- Various applications:
  - Document Classification
  - Token Classification
  - Visual Question Answering

Improve the end-to-end algorithm by labelling the image texts

- Label the image texts to 4 labels: item name, item quantity, item price and other
- Train a LayoutLMv3 model to predict the label using as input the images with their texts and boxes
- Incorporate the label prediction into the process to identify directly the relevant texts and focus on retrieving the final structured list of items

Labelling data in semi automatic process

1. Download random images from the S3 bucket where all images uploaded by the app users are stored
2. Save the OCR outputs
3. For each image get a list of labels generated by a LLM chatbot
4. Check - and correct if needed - the labels given by the chatbot by visualizing them



Data preprocessing

- Convert box coordinates from the OCR format to the LayoultML format
  - OCR boxes: [[x0, y0], [x1, y1], [x2, y2], [x3, y3]]
  - LayoutML boxes: [x_min, y_min, x_max, y_max]
- Normalize the box coordinates from 0 to 1000
- Prepare the inputs with the LayoutLMv3Processor which internally wraps:
  - LayoutLMv3FeatureExtractor for the image modality
  - LayoutLMv3Tokenizer for the text modality

Model training and inference

- Define classification metrics and use F1 to select the best model to as the classes can be imbalanced
- Train the model on Google Colab GPU

Epoch	Training Loss	Validation Loss	Precision	Recall	F1	Accuracy
1	No log	0.741414	0.234742	0.182149	0.205128	0.699683
2	No log	0.454798	0.616438	0.737705	0.671642	0.874921
3	No log	0.338610	0.702532	0.808743	0.751905	0.909206

- On test dataset the F1 score is equal to **0.38** which clearly means that the model is overfitting

Next steps

- Use much more labelled data
- Finetune the hyperparameters to prevent the model from overfitting
- Try using the LayoutXLM models which is an extension of the LayouLMv2 model trained on several languages
- Try moving to a classic NLP model as the images and boxes may not bring useful information

Thank you for your attention