Information Extraction from Images using Optical Character Recognition and Named Entity Recognition

Anish Dixit
andixit@uw.edu

Neel Shah
nshah23@uw.edu

Prerit Chaudhary
prerit16@uw.edu

Abstract

Digitization of printed receipts and invoices is a major presence in most shops or restaurants of late. This is done by parsing the scanned images to extract text from within. Information Extraction is a huge topic in Deep Learning where currently there are various approaches to extract information from images. Most commonly used techniques are Optical Character Recognition and Tagging. In this paper, we implement and compare two models for text extraction and contextual mapping, and try to interpret the results in a meaningful manner. We have used PaddleOCR, DonutProcessor and Graph CNNs in our analysis. We have trained and tuned the models for our use case and leveraged certain quantifiable metrics for comparison of the models. The aim was to compare the performance of the models against baseline approaches and give reasoning as to why certain models perform better than others and try to gauge which models are best suited for receipts and invoices.

1. Introduction

Information Extraction still remains very relevant in everyday life and remains a challenge across different use cases. There has been tremendous evolution and growth in the accuracy of the techniques used to solve this problem. Optical Character Recognition is used to extract characters or words from an image. OCR finds the coordinates of the character and also has to take in consideration spatial details, like baselining, spacing, variations in fonts, etc. Currently, many Deep learning-based text-extractors are being developed and have tackled even more variability in the input like noisy backgrounds and hand-written characters.

With the advent of increased computational power, deep learning techniques have become exponentially powerful and accurate. Computer vision and image classification have been one of the cornerstone use cases of deep neural networks. Detection of useful information from an image can be a great use case of deep learning techniques and identification and accurate recognition of text blobs from images can provide a lot of meaningful information and improve existing systems in vision based uses. Reading text from natural images is still a challenging problem because of complicated background, size and space variations, and irregular arrangements of texts. While most traditional text classification techniques employ character segmentation and classification, this model can be inefficient and cause computational overhead. We intend to focus on identification of a word as a whole, a significantly tougher but more meaningful problem. And beyond this, we can also tag the word in a contextual manner, and understand what entity it can be representing.

Information Extraction from Invoices/Receipts is still relevant because receipts are still being used for corporate card reimbursements and many other use cases. There is still scope for automating this aspect for expense management problems and thus we aim to focus on tagging entities based on context using features such as the position of the word relative to the other words in the document and plotting a relational association between layout information and not based solely on the machine readable words. At the end, we would be comparing the two methods for text extraction and tagging entities such as company, address, date and total.

2. Related Work

There has been a significant amount of work done in the field of OCR and text extraction. In recent years, several methods have been given by researchers for text recognition. [1] There are usually 2 styles of recognition: character-based recognition and word-based recognition. In traditional approaches, they recognize texts character by character. Recent methods have been proposed for directly recognizing the word.

Zhanzhan Cheng et al. [4] described the end-to end trainable network called arbitrary oriented network (AON) which can be trained using only images and word-level annotations. This model is based on CNN and LSTM architecture. The AON module is mainly used for extracting image text's deep features in all four directions and four placement clues also. Output of this network is given to filter gate module for generating relevant integrated se-

quence of features. Then, these features work as input for attention-decoder module for generating predicted character sequences.

Baoguang Shi et al. [5] has given a model based on CNN and LSTM architecture. Initially, they have rectified the input image using TPS (thin-plate spline) transformation, which handles various irregularities of text and makes it new one. Then, they have recognized the text from new image using attention-based sequence to sequence model which predicts character sequence.

Improving accuracy by using Ceep models, currently the research has reached to engines such as Tessaract . R Smith et al. [6] describes that Tesseract is now behind the leading commercial engines in terms of its accuracy. Its key strength is probably its unusual choice of features. Its key weakness is probably its use of a polygonal approximation as input to the classifier instead of the raw outlines.

With internationalization done, accuracy could probably be improved significantly with the judicious addition of a Hidden-Markov-Model-based character n-gram model, and possibly an improved chopper.

In terms of text classification and contextualization also, there is tremendous research done in NLP techniques to correctly classify the extracted text and also look at its contextual importance.

Kowsari K et. al [7] describes a clear and detailed process of basic text classification involving important steps like tokenization, lemmatization, n-grams, Bag of Words, TF-IDF,etc. The main classification takes place after finding Word embedding using models like word2vec, GloVe, FastText which are LSTM or CNN based models to define an embedding for a word. Contextual embeddings can also be generated using models like context2vec which generate these vectors in accordance to the context of words around. The paper [8], authored by researchers at Stanford University, introduce Gibbs sampling, a simple Monte Carlo method used to perform approximate inference in factored probabilistic models. By using simulated annealing in place of Viterbi decoding in sequence models such as HMMs, CMMs, and CRFs, it is possible to incorporate non-local structure while preserving tractable inference.

There are also Transformer based models which might perform better than LSTM or CNN based models. Zian Liu et.al. [10] has considered a pretrained NER-BERT model training the pretrained BERT model for NER and contextual classification. They conclude that it is essential to leverage various entity categories for pre-training, and NER-BERT is able to significantly outperform BERT as well as other strong baselines. Additionally, NER-BERT is effective when only a few pretraining examples are available in target domains. Moreover, the visualization further indicates that NER-BERT possesses good pre-learned knowledge for categorizing a variety of entities. This can be thus used for text classification after extraction.

3. Proposed Methodology

As we have seen in the previous section, there has been quite a bit of existing research in our domain. Existing methods have proven to have a good performance, however the aim of the project was to implement different approaches which have different architectures and compare and contrast the performance metrics on test data and compare it with the baseline performance. The choice for first method was done to select a pre-trained model that has explicitly built and developed for information extraction tasks. In contrast, the second method leveraged a framework that can improved the performance of an already set up architecture for information extraction tasks. The major challenge in information extraction has been the variation in layout and structures of receipts which have different positions for key information such as total, date and address. In addition to that, different receipts can have a variation of font and font size can make matters worse. Hence, the methodology we intend to follow involves carrying out a comparative analysis of two model architectures that are prevalent for our use cases of text extraction and contextual mapping. The first method uses Paddle OCR and Donut processor which were developed for information extraction. The second method aims to improve upon the legacy architecture by incorporating Graph Convolution Networks as a catalyst in the framework with the aim of capturing structural relationships between different entities inside and outside a table in receipt and invoices that can help improve the performance of downstream task of information extraction.

3.1. Dataset

The dataset used was Scanned receipts OCR and information extraction (SROIE) data [16] which consisted of 1000 scanned receipt images published by the International Conference on Document Analysis and Recognition where each image had a corresponding tsv and txt file that had coordinates for bounding boxes and text annotations. Each image in the data had coordinates for the bounding boxes and the transcript of each bounding box. Coordinates are represented as rectangles with four vertices, which are in clockwise order starting from the top and text files had annotations for an image. Each receipt image had elements such as name of the company, address, product name, price and total cost etc with the text mainly consisting of digits and English characters. The below two figures represent an example of the images available in the dataset. There were receipts from the same company but having different product names and dates. The dataset was divided into a training/validation set and a test set with a 80:20 split as we wanted to have more images in training given the comparatively lesser number of images available for training.

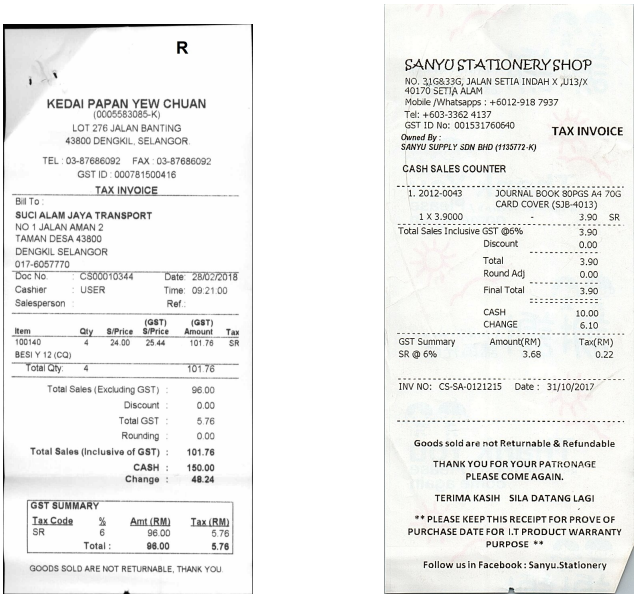

Figure 1. Sample Images from the Dataset

3.2. Approach 1

In this approach, we use a combination of an Optical Character Recognition model - PaddleOCR, and a Transformer which has image encodings and text decodings in order to identify text as well as classify it accurately into pre decided entities present in the dataset.

3.2.1 PaddleOCR

PP-OCR is an practical ultra lightweight OCR System developed by researchers at Baidu [14]. The framework of the OCR model involves 3 major steps: Text Detection, Rectification of Detected Boxes and Text Recognition.

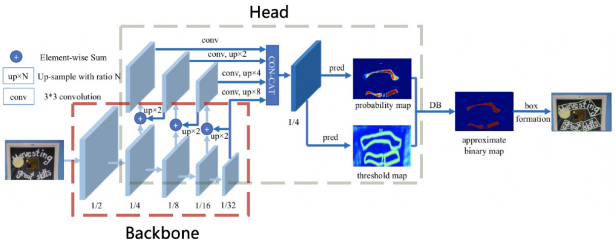

Figure 2. PaddleOCR Architecture

The purpose of text detection is to locate the text area in the image. PaddleOCR uses Differentiable Binarization (DB) as text detector which is based on a simple segmentation network. As can be seen in Fig. 2, it consists of a light backbone, with a number of 3*3 convolution filters, increasing from 2,4,8,16 to 32 filters. It also employs deep learning techniques such as learning rate warmup and a cosine scheduler for learning rate delay. The MobileNet model is used as a Backbone. This is fed to a 'head' which consists of sequential Up-Conv Layers, the output of which is binarized into a prediction map.

Next, the text box needs to be transformed into a horizontal rectangle box for subsequent text recognition, which can be achieved by geometric transformation as the detection frame is composed of four points. Training a text direction classifier is a simple image classification task. We adopt the following four strategies to enhance the model ability and reduce the model size: light backbone, data augmentation, input resolution and PACT quantization.

Finally, CRNN is used as the text recognizer. CRNN integrates feature extraction and sequence modeling. It adopts the Connectionist Temporal Classification(CTC) loss to avoid the inconsistency between prediction and label. It consists of a sequence of Deep Bi-directional LSTMS and Convolutional Feature Maps which give good accuracy for character recognition.

3.2.2 DonutProcessor

Donut is a transformer offering from HuggingFace [15]. Donut consists of an image Transformer encoder and an autoregressive text Transformer decoder to perform document understanding tasks such as document image classification, form understanding and visual question answering. The general purpose flow can be seen in the figure 3.

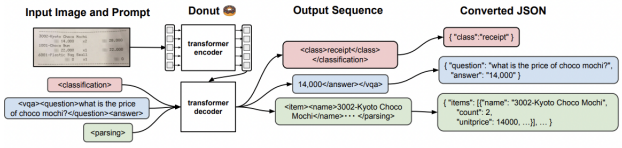

Figure 3. Donut Processor Architecture

In this approach, we use the VisionEncoderDecoder model, which consists of multiple layers of deep convolutional networks. The architecture builds off multiple blocks of the following layers:

- 1. LayerNorm
- 2. 2D Convolutional Layer
- 3. Dropout Layer
- 4. Fully Connected Layer

LayerNorm is used instead of the conventional BatchNorm since it normalized the features independently across batches, useful for smaller data sizes. The activation function used is GeLU (Gaussian Error Linear Units), in consistency with common transformer practices. BeRT model is also used in the text embeddings based decoder model.

The Donut Model is trained for 20 epochs on the SROIE dataset explained earlier, and because of the availability of annotated bounding box coordinates and expected textual categories, we notice good results.

3.3. Approach 2

The major gap in extracting information is that there is difficulty in incorporating contextual information within the different elements of the receipts. Graph Convolution Networks (GCN) is a concept that can be utilized to overcome this challenge. GCNs can be used to represent and capture the contextual and structural relationship between different entities as most receipts and invoices have a hierarchical and interconnected structure, where different elements like total amount and quantity are associated to each other together. GCNs are able to emphasise the relationship between entities by considering the local and global neighborhoods of each node in the graph. This allows the model to leverage the surrounding context when extracting information from individual elements.

3.3.1 CNN + Transformer + Graph GCN

The overall architecture follows the same process of Information Extraction through OCR, using Graph Convolution Networks as a tool to improve the downstream process of Named Entity Recognition and Text Classification.

The overall architecture contains 3 modules:

- Encoder: This module encodes text segments using Transformer to get text embeddings and image segments using CNN to get image embeddings. The text segments and image segments stand for textual and morphology information individually. Then these two types of embeddings are combined into a new local representation X, which will be used as node input to the Graph Module.

- Graph Module: This module can catch the latent relation between nodes and get richer graph embeddings representation of nodes through improved graph learning convolutional operation. Meanwhile, bounding boxes containing layout context of the document are also modeled into the graph embeddings so that graph module can get non-local and non-sequential features.

- Decoder: After obtaining the graph embeddings of the document, this module performs sequence tagging on the union non-local sentence at character-level using BiLSTM and CRF, respectively. In this way, our model transforms key information extraction tasks into a sequence tagging problem by considering the layout information and the global information of the document.

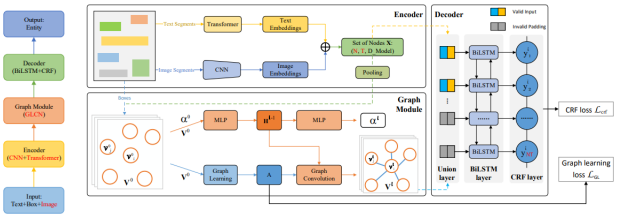

Figure 4. Overall Architecture

The input image goes to a number of image preprocessing steps such as cropping a receipt from image, histogram adjustment, brightness adjustment etc. Next this image goes through OCR systems like Google's cloud APIs or Tesseract or any OCR system for text detection. The image is passed through a CNN model (Resnet) to generate image embedding and through a Transformer model to generate text embeddings. The outputs of OCR and the bounding boxes on receipts are also used to create the input graph and corresponding adjacency matrix for the image to be used by graph neural networks.

The generated embedding are then combined to be stored in an node feature matrix along with the adjacency matrix which act as node input to the Graph neural network. Combining with the image embedding show significant improvement in model as they contain useful information like text font, text curvatures etc. For example: Text in category STORE NAME has larger font size than the other text present on receipts, thus helping the model for right predictions. The adjacency matrix (A), feature matrix (x) and the labels (y) are inputs to the final classification model and graph networks. A, x and y will be used to train a graph based neural network models which will learn to classify each node in the possible classes. The GCN, Graph Convolution Neural Network learns to embed node feature vector by generating a vector of real numbers that represents the input node as a point in an N-dimensional space, and similar nodes will be mapped to close neighboring points in the embedding space, allowing to train a model able to classify the nodes.

Finally a Bi-LSTM and CRF based decoder is used to get the final class predictions. CRF (Conditional Random Fields) help in getting contextual predictions based on neighbouring graph nodes for the specified class and then classifying the specific text using Bi-LSTM. The model was tuned to improve the predictions with hyper parameter tuning for graph aggregation functions using mean and max, optimizers like SGD and Adam. 15 epoch were run in total and metrics such as CRF Loss, Mean Precision, Mean Recall, Mean F1 and Mean Accuracy were tracked during the training process.

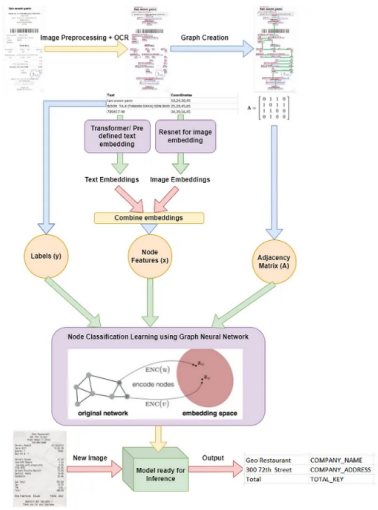

Figure 5. GCN Pipeline

```
[Epoch Validation] Epoch:[15/100] Total Loss: 108.737311 GL_Loss: 0.014841 CRF_Loss: 107.253217
+---------+----------+----------+----------+----------+
| name    |      mEP |      mER |      mEF |      mEA |
+=========+==========+==========+==========+==========+
| company | 0.511628 | 0.553073 | 0.531544 | 0.553073 |
+---------+----------+----------+----------+----------+
| date    | 0.378295 | 0.410084 | 0.393548 | 0.410084 |
+---------+----------+----------+----------+----------+
| address | 0.488176 | 0.146924 | 0.225869 | 0.146924 |
+---------+----------+----------+----------+----------+
| total   | 0.571596 | 0.404485 | 0.473735 | 0.404485 |
+---------+----------+----------+----------+----------+
| overall | 0.491922 | 0.295344 | 0.369091 | 0.295344 |
+---------+----------+----------+----------+----------+
```

Figure 6. Metrics during Training

3.4. Performance Metrics

Metrics such as Mean Average Precision, Mean Average Recall, Mean Average F1, Mean Average Accuracy and Intersection over Union(IoU) were used to evaluate performance of the object detection component of the model. Intersection over Union gives an assessment of the overlap between the ground truth and the predicted bounding box. The coordinates for the ground truth bounding box is a part of the input data and that allowed us to measure how well the predicted bounding box was identifying the key text elements in the image.

Levenshtein Distance was leveraged to compare the text extracted and tagged after named entity recognition as it is a commonly used metric for string matching tasks such as information retrieval and pattern recognition. Levenshtein distance is a lexical similarity measure which identifies the distance between a pair of strings by counting the number of operations (insert, delete or replace) at a character level that need to be made in one string to make it identical to the other string. Mean Levenshtein Ratio which is a metric based on Levenshtein Distance. The interpretation of Levenshtein Ratio is inverse of the Levenshtein Distance. A higher Levenshtein Ratio implies that the strings are similar and the results have been shown for the two approaches on the test data.

4. Results and Discussion

We evaluated the metrics over the test data having 100 images. A sample image has been shown below to show how the text was identified, extracted and tagged using the two methods.

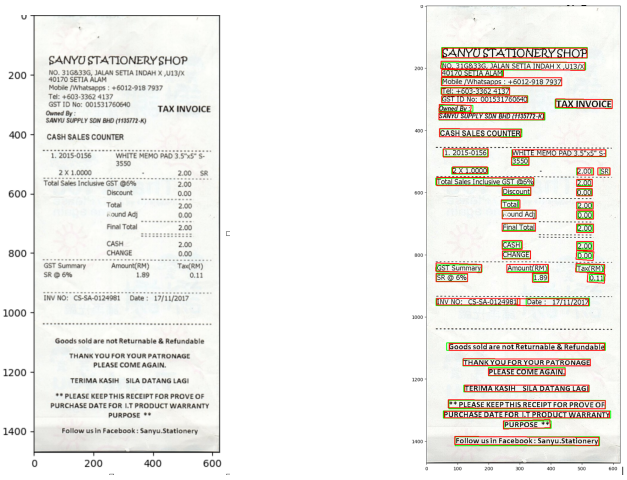

Figure 7. Original Image and Annotated Image

4.1. Approach 1

Mean Levenshtein Ratio for test data: **0.934**

Mean Average Precision: **0.827**

Intersection over Union: **0.7432**

{ "address": "NO. 31G&33G, JALAN SETIA INDAH X,U13/X 40170 SETIA ALAM",
"company": "SANYU STATIONERY SHOP",
"date": "17/11/2017",
"total": "2.00"}

Figure 8. Output for Sample Image using Approach 1

4.2. Approach 2

Mean Levenshtein Ratio for test data: **0.672**

Mean Average Precision: **0.49**

{"company": "SANYU STATIONERY SHOP,company",
"address": "NO. 31G&33G\\, JALAN SETIA INDAH X \\,U13/X,address4",
"date": "17/11/2017"}

Figure 9. Output for Sample Image using Approach 2

The performance for both the models can be attributed the fact that the input data had receipt images with similar layout, structure and lighting and hence even with limited epochs, performance metrics for both models were good.

We saw that the overall performance of the first approach was better than the second one as PaddleOCR and Donut Processor have been primarily developed with a specific use case i.e. information extraction. They have been pre-trained on a variety of templates to extract text and contextual embeddings and hence are more optimized for this task.

GCNs can be used across a variety of use cases in computer vision but are primarily used for social network analysis and recommender systems where it is able to capture relational information in structured data. The GCN approach also performs good as it is able to classify correctly and able to distinguish the classes using the graph networks which is more efficient and fast.

We also tried other methods but the biggest challenge we faced was computation power. As most of the models used for OCR+Text Classification are Neural Networks with multiple layers, activation function, etc, they need large GPUs to train the data which we were not able to provide completely. Thus for some of our other methods, the results were comparatively very bad due to lack of training. Thus we chose these 2 methods to perform information extraction.

5. References

[1] A. Shrivastava, J. Amudha, D. Gupta and K. Sharma, "Deep Learning Model for Text Recognition in Images," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-6, doi: 10.1109/ICCCNT45670.2019.8944593.

[2] MVV Prasad Kantipudi, Sandeep Kumar, Ashish Kumar Jha, "Scene Text Recognition Based on Bidirectional LSTM and Deep Neural Network", Computational Intelligence and Neuroscience, vol. 2021, Article ID 2676780, 11 pages, 2021.

[3] Ha, Hien Thi, and Ales Horak. "Information extrac- ´ tion from scanned invoice images using text analysis and layout features." Signal Processing: Image Communication 102 (2022): 116601.

[4] Zhanzhan Cheng, Yangliu Xu, Fan Bai, Yi Niu, Shiliang Pu and Shuigeng Zhou, "AON: Towards Arbitrarily-Oriented Text-Recognition", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5571-5579, 2018.

[5] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao and Xiang Bai, "ASTER: An Attentional Scene Text Recognizer with Flexible Rectification", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PP, no. 99, pp. 1-1, June 2018.

[6] R. Smith, "An Overview of the Tesseract OCR Engine," Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil, 2007, pp. 629-633, doi: 10.1109/ICDAR.2007.4376991.

[7] Kowsari K, Jafari Meimandi K, Heidarysafa M, Mendu S, Barnes L, Brown D. Text Classification Algorithms: A Survey. Information. 2019; 10(4):150. https://doi.org/10.3390/info10040150

[8] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370.

[9] www.expressexpense.com

[10] Liu, Z., Jiang, F., Hu, Y., Shi, C., and Fung, P. (2021, December 1). Ner-Bert: A pre-trained model for low-resource entity tagging. arXiv.org. https://arxiv.org/abs/2112.00405

[11] Anh Duc Le, Dung Van Pham, Tuan Anh Nguyen Deep Learning Approach for Receipt Recognition https://arxiv.org/ftp/arxiv/papers/1905/1905.12817.pdf

[12] https://arxiv.org/abs/2004.07464

[13] https://prakhargurawa.medium.com/how-graph-neural-networks-are-used for-information-extraction-32e9ae0e7acc

[14] arXiv:2009.09941

[15] arXiv:2111.15664

[16] https://rrc.cvc.uab.es/?ch=13