Text Extraction for KYC Processing using teX.ai for a Financial Services Firm

- Increase the number of applications processed in a single day by 87%.
- Project was completed in a span of 3-4 months.

teX.ai™

Customer Background

Client is one of the fast-growing firms in India providing their services in the financial domain by embracing digital transformation to their advantage. Client provides quick and efficient Credit Score Rating to their customers and assist many banks in assessing their customers' credit score. They intend to change the way credit is delivered in India by channeling the power of technology and digital platform. In the process of validating the applicants to provide the Credit Score, client had to process thousands of scanned bank statements to fulfill the KYC (Know Your Customer) requirements for the applicants.

Business

teX.ai, Text extraction

Technology

Table Detection and Extraction: TensorFlow Object Detection API, Pytesseract, Tabula, PDFplumber, Camelot

Header Information Extraction: Anago, Pycrfsuite, CRF (conditional random fields), LSTM (type of RNNs), REGEX

Pre-processing and Post Processing Tools: openCV, xPDF, Poppler, pandas, json

Application Deployment and Access: Flask, Requests

Domain

Financial Service

Key Highlights

- Challenges texts were extracted using combined RNN model called LSTM-CRF.
- Despite the inherited challenges of the PDF documents, the project was completed in a span of 3-4 months.
- The time taken to process single file was less than minute. This was a big leap from existing process, and it enabled them to increase the number of applications processed in a single day by 87%.
- Client's challenge of extracting the right information from bank statements was resolved forever.

Business Requirement

- The documents that were to be extracted were of two types:

  o Scanned images

  o Digital PDFs

- 5 fields were required to be extracted from the documents and these fields were situated in both the Table section (Tabular Data) and outside the tables (Peripheral Data). The 5 fields were:

  o Account holder name

  o Date

  o Name of the bank

  o Transaction details

  o Address

- An initial corpus of 2000 bank statements were used to train the models needed for the extraction.
- The system needed to be scalable for a large inflow of documents on a daily basis.

Challenges

- Client was already using text conversion tools to extract the data from the customers' Bank statement and using its JSON output in their in-house application for further processing. However, the client was not content with the accuracy level of the output for the text extraction.
- The fieldnames to be extracted varied from bank to bank. For example, "Credit amount" in the transaction details can be "Cr", "Credited", "Amount Credited" etc.,
- Location of Text to be extracted varied from bank to bank, with some residing within tables and some outside tables.

Objective

- To build a text extraction model for bank statements employing the teX.ai product and build a pipeline for easy orchestration of future bank statements for text extraction.

Solution Overview

- We methodically classified the bank statements first by different types of bank and then by quality.
- The tables were located leveraging the bounding box method that works based on Deep Learning Network and texts were extracted using teX.ai.
- Texts that were present outside table were extracted deploying a combined RNN model.

Approach & Implementation

Types of Bank Statements: One of the first things identified was that there were 200 different types of bank statements amongst 2000 files provided.

Quality of the Bank Statements: 830 files were classified as good quality and this set was majorly utilized to create the initial NLP model.

Extraction of Data from Tables (Tabular Data):

- Scanned Bank Statements:

Table Identification: Leveraged Deep Learning Network to detect the tables by drawing the bounding boxes around them. Object Detection Neural Network was used at this stage of the process.

Table Contents: The required text from the detected tables were extracted Tabula and Camelot which are embedded in teX.ai.

- Digital PDFs:

Tabula and Camelot were used to instantly extract the textual data from the tables.

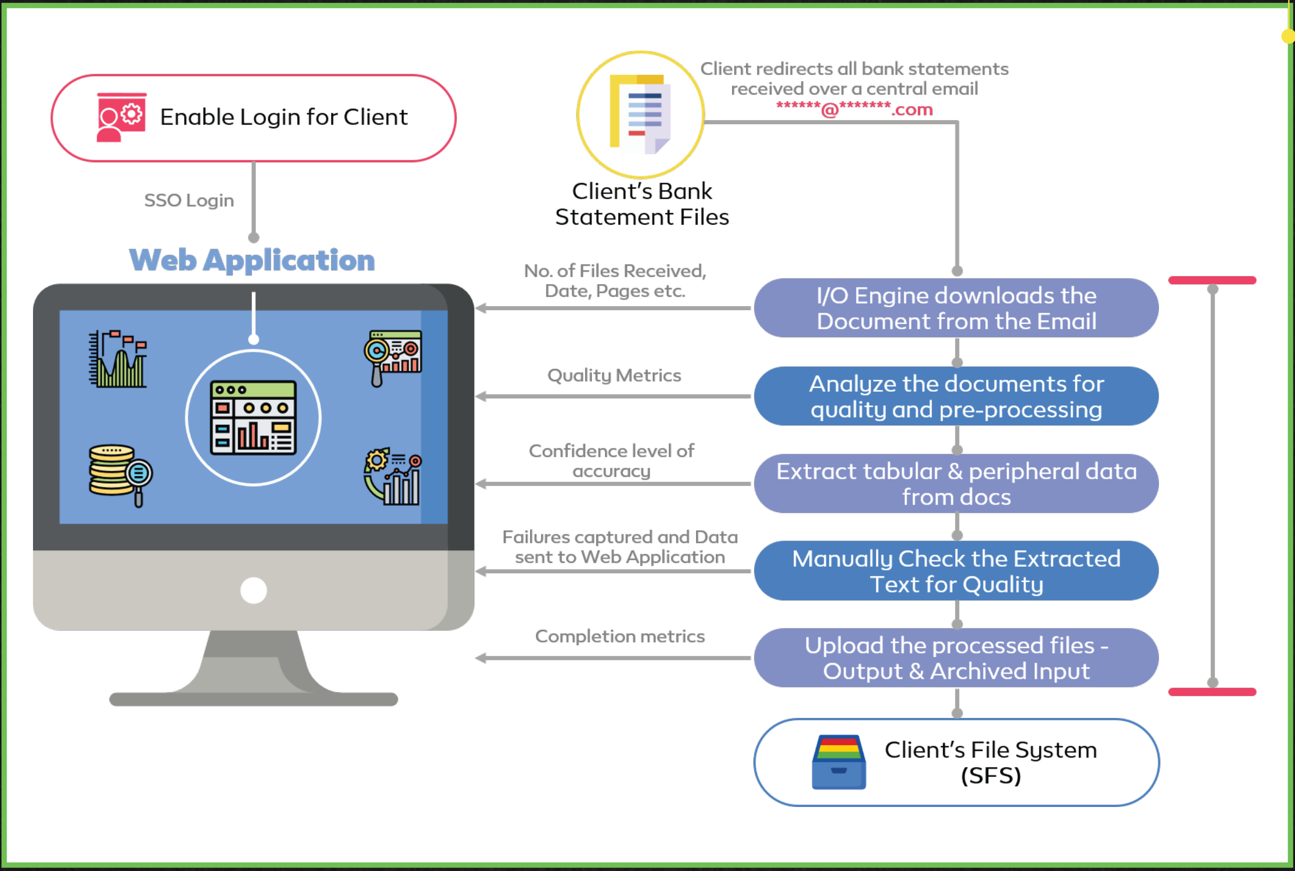
Extraction of Data Outside Tables (Peripheral Data):

- Texts were extracted from both digital and scanned PDFs by using a combined RNN model called LSTM-CRF.
- This model decisively extracts all required peripheral data regardless the type (Scanned or Digital) of PDF document.

Output: The outputs were generated as JSON file, which client used on their in-house product.

Set-up: teX.ai was setup on-prem in the client's datacentre and training was provided to their in-house team. Flask & Requests were leveraged to build the pipeline.

Process Workflow:


Enable Login for Client
SSO Login
Web Application
Client's Bank Statement Files
Client redirects all bank statements received over a central email
******@*******.com
No. of Files Received, Date, Pages etc.
I/O Engine downloads the Document from the Email
Quality Metrics
Analyze the documents for quality and pre-processing
Confidence level of accuracy
Extract tabular & peripheral data from docs
Failures captured and Data sent to Web Application
Manually Check the Extracted Text for Quality
Completion metrics
Upload the processed files - Output & Archived Input
Client's File System (SFS)

Business Impact

Time: The time taken to process a single file was less than minute. This was a big leap from existing process, and it enabled them to increase the number of applications processed in a single day by 80%.

Accuracy: The accuracy was closed to 90% and with a larger data set, it was designed to increase.

Get in touch to see how teX.ai can help you!

✉ info@tex-ai.com

☎ +1 (888) 207 5969 (Toll-free)

🌐 www.tex-ai.com

Suite 100, 19925 Stevens Creek Blvd,
Cupertino, CA — 95014

teX.ai ™