Project Report

Document Digitization

Abhishek Masand

Sukrit Goel
Director
IT Advisory Services
EY

Data Digitisation

Introduction

Digitization is the process of converting information into a digital (i.e. computer-readable) format, in which the information is organized into bits. The result is the representation of an object, image, sound, document or signal (usually an analog signal) by generating a series of numbers that describe a discrete set of its points or samples. The result is called digital representation or, more specifically, a digital image, for the object, and digital form, for the signal. In modern practice, the digitized data is in the form of binary numbers, which facilitate computer processing and other 17 | Page operations, but, strictly speaking, digitizing simply means the conversion of analog source material into a numerical format; the decimal or any other number system that can be used instead. Digitization is of crucial importance to data processing, storage, and transmission because it "allows information of all kinds in all formats to be carried with the same efficiency and also intermingled". Unlike analog data, which typically suffers some loss of quality each time it is copied or transmitted, digital data can, in theory, be propagated indefinitely with absolutely no degradation. This is why it is a favored way of preserving information for many organizations around the world.

Problem Statement

Data Digitisation is the process by which physical or manual records such as text, images, video, and audio are converted into digital forms. This is of paramount importance when projects need directions based on already established facilities and the implementing agency needs to find the scope for expansion. Benefits include, digitised data offers the following benefits, long term preservation of documents, orderly archiving of documents, easy & customised access to information, easy information dissemination through images, text, CD-ROMs, internet, intranets, and extranets. To transform un-structured information to structured it is important that we are able to contextualise the information. Thus ICR tools leverage machine learning to infer context based on historical information.

Contract Digitisation and Automation for QRM

Problem Statement

Organizations now are struggling to handle the **unstructured information**. The volume and unstructured nature put them into a real big challenge when the processing takes place for such information.
Crucial documents such as contracts and emails are a form of unstructured data. Apart from legal obligations, there is a need to build a robust automated contract management system which will help the organization to comply with the new/ changing regulations and identify hidden risks and opportunities efficiently.
Contracts contain commercial terms, payment terms, pricing, renewal information, obligations, incentives, risk, liabilities, which can be used by procurement, sales, legal, M&A, regulatory, facilities, and other divisions of the company for better performance. Hence contracts have become a valuable and rich source of information which can help the organization in intelligent decision making.
Industry in focus: Financial Services, Legal, IT Services, Product companies, Healthcare. Contracts, compliance, and regulations practically form the basis of most of the transactions in these industries.

CHALLENGES FACED BY ORGANIZATIONS WITH MANUAL CONTRACT MANAGEMENT PROCESS

- Human error & misses causing millions of dollar penalties to the firms.
- Manual effort to rectify the errors.
- Manual Audit process need several hours of lawyers & legal advisors
- Manual effort to analyze the contract and delay in response to regulatory requirements.
- Unavailability of the information due to silo-ed approach.
- No visibility to the contract terms, legal clauses, values, commitments.
- Hard to understand the legal terms hence no visibility to impacts in case of non-compliant. Scenarios (such scenarios may arise in future due to change in external factor).
- Governance & Monitoring over time, loss of ownership.

Objective

To Build A Contract Management System that can be used to digitise paper contracts, identify certain Clauses and tables from the contract. It can also be used to validate the Contract by determining verification checks like Clause, signature and date existence.

Limitations

Due to the sensitive nature of contracts, most clients would now want to send their confidential data to the cloud services, hence an in-house solution needs to be generated

OCR, Handwritten Text
OCR doesn't remember the structure of the text, and can give a wrong output very easily. It also docent work properly on photos taken in extreme lighting conditions

Table Extraction

Table extraction is difficult on tables with no column dividers, orientation issues, curved lines in scanned pdf

Tools Used

Python - Programming Language
Pdf2image - Converting PDF to Images
Pytesseract - OCR Library
TFIDF Vecotriser
Multinomial Naive Bayes
Fastai
Regex
NLTK

Learnings, Mistakes and Previous Approaches

Simple Keyword Extraction does not work when searching for clauses, context matters a lot, and the result had comparatively less accuracy than our current method.

Random Forest, SVM, Logistic Regression provide lesser accuracy as compared to Multinomial Naive Bayes.

In general, Multinomial Naive Bayes provided a better accuracy on our text data. Also, I had tested a state of the art deep learning algorithm called ULMFiT using the fast.ai library, we got a fairly good accuracy, but it depended on a language model and hence I chose not to use it, as we're working on legal data.

Directly applying classification on our text data with None as a label along with the other 9 labels provided lesser accuracy. Instead our current classification approach with Elastic Search Works as a means to reduce the classification inputs works better.

For Clause existence, rule based approaches work a lot better rather than a learning approach.

Methodology And Approach

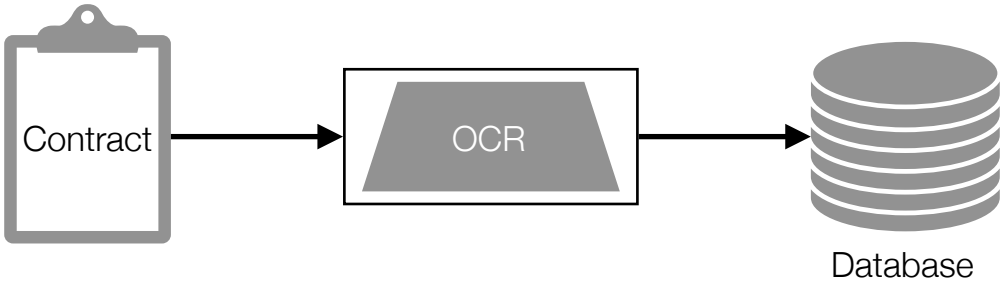
Making Contracts Digital

Contracts are generally received as hard copies, so even before applying any time of algorithm on it, we need to convert it into text format.

Our solution was to use google's local OCR library known as tesseract. Tesseract is open-source and free and works good on scanned documents with little-to-no noise.

The first step in the process was to convert the individual pages of the PDF to images, and then our OCR was applied to it.

The resulting OCR text output was stored to the database for further use.



Clause Classification

Collecting Data consisted of scraping a legal website for clause occurrences in different contracts. For the POC, 9 main clauses were chose in order to demonstrate the feasibility of the project. 9200 samples were scraped from that website and then labeled according to category and stored to the data frame.

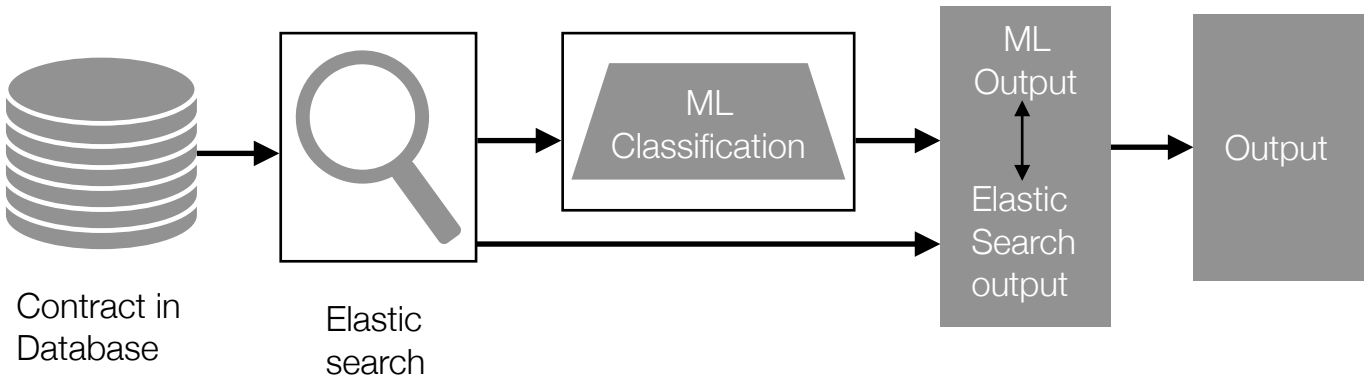
We trained a classification model based on 9200 samples of data across 9 categories using TF-IDF Vectorizer and Multinomial Naive Bayes for our classification problem. First step was to shorten our search, we used elastic search to get a handful of paragraphs which can contain our particular clause.

Second step was to preprocess our data using NLP techniques, but not lemmatising or stemming our words in order to preserve the words, this is important as some words can change the whole meaning of the paragraph in a legal document.

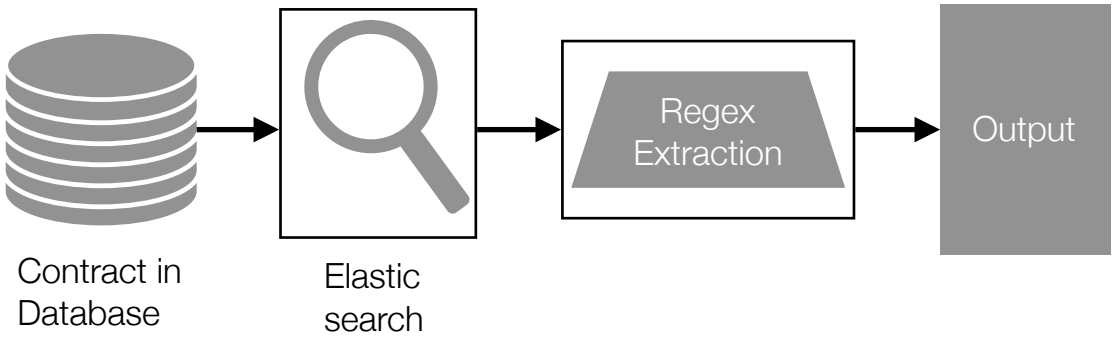
Third step was to predict the class of our extracted clauses.

We then matched the clauses with the extracted keyword belonging to our elastic search query.

In the end we got a data frame of results corresponding to the top 2 clauses that can pertain to our contract.

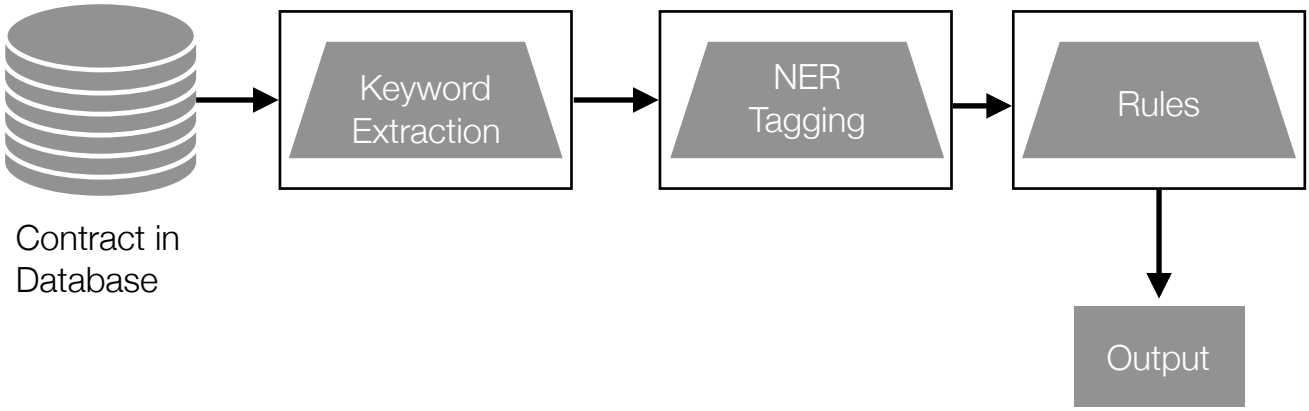


Clause Existence
For getting clause existence, a rule based approach using regex was applied. We applied elastic search again on our data and then in the paragraphs that were received, we applied different regex patterns in order to find out if our clause existed.



Party Extraction
For finding out the different parties in our document, we first split our document into paragraphs and then use keyword extraction to find out the paragraphs which can contain the parties.

After extracting the parties, we applied NER taggers to our paragraph and then using rule based approaches, extracted the parties from the NER output.



Results

There were three different tasks in our contract management system:

We achieve a staggering 96% accuracy on our Clause classification dataset, on 2 out of 9 clauses we had considerably less data than the others, as a result, the major misclassifications were only on those samples.

Second was Clause Existence, it was a rule based approach, as we only had to see the existence or certain headings, parties in the contract. Our algorithm correctly classified 94% of the data out of 11 contracts that we tested upon. It only went wrong when there were no clauses in the data, but it found the clauses to exist in some table headers,

Third was Party Extraction, we had to extract the two parties, whom the contract was between, we achieved an accuracy of 92% on our Party Extraction, some of the cases where our algorithm went wrong were when this clause was in between the contract and not in the first couple of pages. In the contract, if some other organisation name is listed, our algorithm misclassifies that particular party.

Future Work

Future work mainly consists of getting more data using actual contracts, since using text augmentation can give not so contextually accurate data on legal text and instead of helping, it might be hurtful .

Also, it would be better to use deep learning using glove or Elmo embeddings, in order to get more accurate results. Using state of the art algorithms with Bi-directional LSTM with attention units will keep in mind the context when classifying the clauses.

SSI Document Digitisation

Problem Statement

Standing Settlement Instructions (SSI) documents are an incredibly important part of the clearing and settlement process in the industry. They include information such as correspondent bank details and banks involved in cross-border payments which are used by the ordering and receiving institutions. It is the document to find out where to deliver cash or security to the accounts party. SSI process is only partly automated in the industry today and needs to be understood by everyone who is working with it in a timely manner.

Receiving information for the same which is timely and accurate in extremely important.

Objective

SSI documents contain tables in various formats that contain data like SWIFT code, name of account, country, correspondent bank, cash account, etc.

The data from the table has to be extracted and properly structured into key value pairs for further use.

This will help people further in the process from reopening the document, and hence reduce the bottleneck time.

Limitations

Although there are many tools available for table extraction, they all fail in multiple scenarios example skewed lines, no column dividers, wrapped text, gap between lines, etc.

When tables have a slight angle, it is harder for the tools to properly divide columns and sometimes even recognise the text inside

Tools Used

Amazon Textract
Python

Stanford NLTK Tagger
Pandas
Regex

Learnings, Mistakes and Previous Approaches

Libraries such as Camelot and Tabula depend upon the table lines for segmenting cells, and many SSI documents contain tables in which column dividers were not explicitly drawn, so it was difficult to use those libraries.

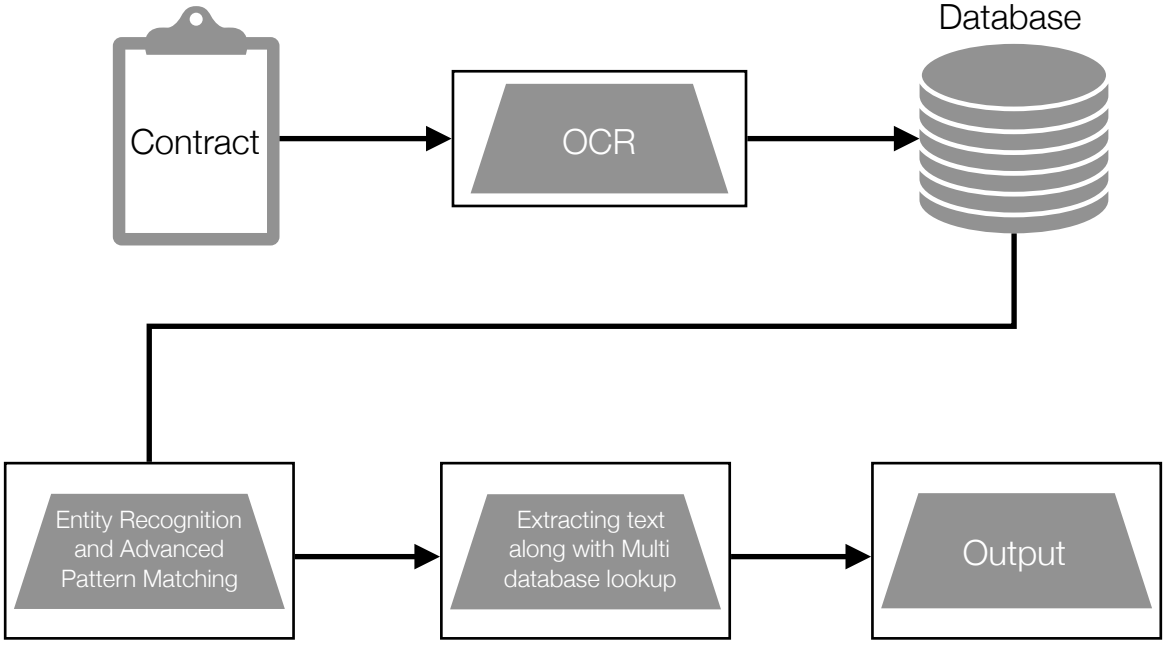
Methodology And Approach

First step in the process was to convert the document from a pdf to a tabular format. We used the amazon textract library to convert the tables from the document to structured csv files.

Next we converted the csv files to data frames. After converting them, we preprocessed our dataframes to remove anomalies such as empty rows and columns depending upon the table data.

We used a final information extraction and normalization layer that uses NLP techniques such as Entity Recognition and Advanced Pattern Matching to extract fields of interest. After identifying the fields of interest, we used a manually curated Swift database to correlate and find data associated with the particular swift code in each row, the other things that we picked out from the rows were account details, bank details, beneficiary, etc.

The final output is a normaliseds data structure containing fields such as BIC, Location, Currency, and Name of Correspondent bank that are ready for ingestion by downstream processes.



Results

We correctly identified all the data in the table, and some more as we used lookup from our self curated SWIFT database. So if even some of the data was misspelt in the table, our algorithm fixed that.

The only time our algorithm failed was when the table was not extracted properly, and had major mishaps, example - A SSI Document contained wrapped text which took around 3 rows per cell, Amazon Textract classified the 3rd line In the cell as a different row, leading to a wrong output.

Future Work

Future Work includes building an in-house table extractor that can handle the problems discussed in limitations also improve ways to dynamically extract more columns if required.

ID card Digitisation

Problem Statement

Verification of Identification cards such as Passport, Driving License is done manually at airports, on site KYC's.Companies often face issues with performing KYC operations. The data for KYC is usually images or scanned copies of the individual to be verified and the process of extracting data from PAN card / Passport / National Identity Card and storing in files and folders manually can be a strenuous task requiring considerable human effort. With this project in hand, we intended to automate the entire process by taking in any quality and type of identification document and extracting relevant fields from it.

Objective

In today's era, with the rise of mobile payment banks such as Airtel Bank and Paytm, The process of automating KYC is extremely important, so as to remove the repetitive work of verifying the id's and hence reduce processing time between user and the service, thereby creating a more streamlined experience for the user.

This project aims to create a fully automated system for extracting the details of the user and to verify it using existing data.

Limitations

ID cards have various formats and almost no two ID cards follow the same structure, therefore it is difficult to make a generalised solution using rule based approach.

Open-source OCR solutions don't work too well on handheld images of ID cards as well as they do on scanned ones.

Tools Used

Python
Regex
Opencv
Tesseract

Learnings, Mistakes and Previous Approaches

Simple keyword search is not beneficial for our use case. As it is rule based, a simple search for gender such as "male" can easily output wrong results when it encounters a name like "malek".

Hence a more ingenious approach, either using Regex or entity extraction is needed.

Methodology And Approach

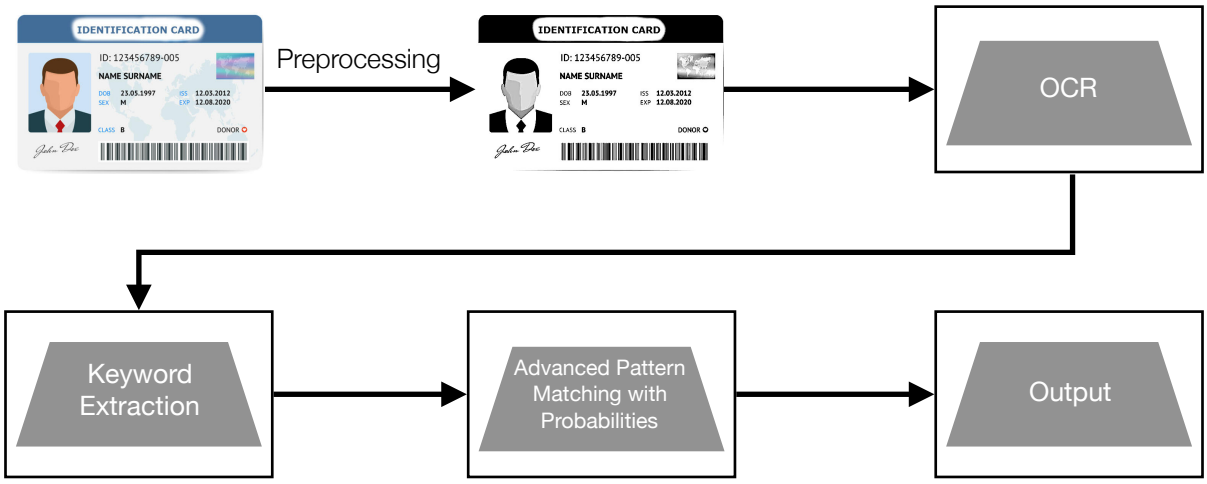
The first step was to preprocess our ID card image in order to increase the clarity of the text present in the ID card, remove small unwanted artefacts from the image and to correct if any skew was present .

After the preprocessing was complete , we used tesseract OCR to extract text from the documents. On the extracted text, we used NLP to clean the text of any unwanted characters.

As our text still contained unwanted items, we used a Keyword Extraction approach to find the fields that are of interest.

On the extracted text, we used regex along with a probabilistic matcher to extract the important Fields.

The final output is a normalised data structure containing fields such as name, fathers name, age, address, and ID card Number that are ready for ingestion by downstream processes.



Results

The algorithm was able to correctly predict the details of all the id cards properly, except when the image quality was subpar, leading to a bad our output that does not contain all or some of the text.

We tested our algorithm on Aadhar Cards, PAN Cards, Indian Passport, French ID cards along with African ID Cards.

Future Work

We can use deep learning to localise the important locations in the image of the ID card, segmenting the locations where important data can be found. That will help us remove unwanted text, improving our input data. We can also use a custom NER based approach to tag our output and then use the ww wa results to find the details.