Providing Structure to Unstructured Financial Documents Utilizing AI/ML

Javier Tordable, Technical Director, Office of the CTO, Google
Shengyang Dai, Engineering Manager, Vision API, Google Cloud
Ross Biro, Chief Technology Officer, The Interface Financial Group
Vishnu Kumar, Chief Financial Engineer, The Interface Financial Group
Michael Cave, Senior Data Scientist, The Interface Financial Group
Zheng Xin Yong, Junior Data Scientist, The Interface Financial Group

Abstract—This project describes the use of Google AI technology approach for unstructured b2b invoices. Features are generated to capture layout and textual properties for each field of significance, and weighted to reveal key factors that identify a field on an invoice. Feature selection, threshold tuning, and model comparison are evaluated. Overall, we have preliminary achieved 99% accuracy on a field level basis.

INTRODUCTION

Today commercial and consumer-lending solutions require the ability to acquire and leverage third-party data, to offer funding opportunities to streamline all aspects of today's business. New approaches have been developed to address major financial service tasks like originations, on-boarding, underwriting, structuring, servicing, collection and compliance.

One of the main areas of development in underwriting, structuring and compliance for commercial lending is related to integration of the financial institutions with potential or current clients' accounting systems. This allows financial service providers to instantly capture any information regarding banking activity, balance sheets, income statements, Account Receivables and Account Payable reports into data feed formats. Enabling instant analysis by the decision engines to provide qualitative and quantitative provisions for credit limits and approval of all types of loans and funding in order to increase inclusivity and to offer funding for a wider range of businesses than before.

However, borrowers are reluctant to enable third parties to access internal data, which creates a barrier for adoption of this kind of system. Hence, clients submit unstructured financial documents such as bank statements, audited or interim financial statements and reports, invoices, etc., for example via a drag & drop method on a client portal or email. Many funding companies are using OCR and/or Virtual Printers technologies in the background to extract the data but the results are still far from being consistent or reliable. This solution still requires manual effort to achieve acceptable accuracy, which may cause additional inconsistency and provides an unsatisfactory resolution.

We started to explore Google AI, and Google Cloud infrastructure for machine learning to understand how to apply these tools to unstructured financial documents and to create an intelligent data extraction tool.

We have implemented a system using Google AI and its Vision & Machine Learning products as a base, coupled with customized algorithms developed by IFG to replace current technologies. Allowing financial service companies and banks to ingest unstructured documents as a data feed, consequently broadening the scope and providing more reliable commercial and consumer lending worldwide.

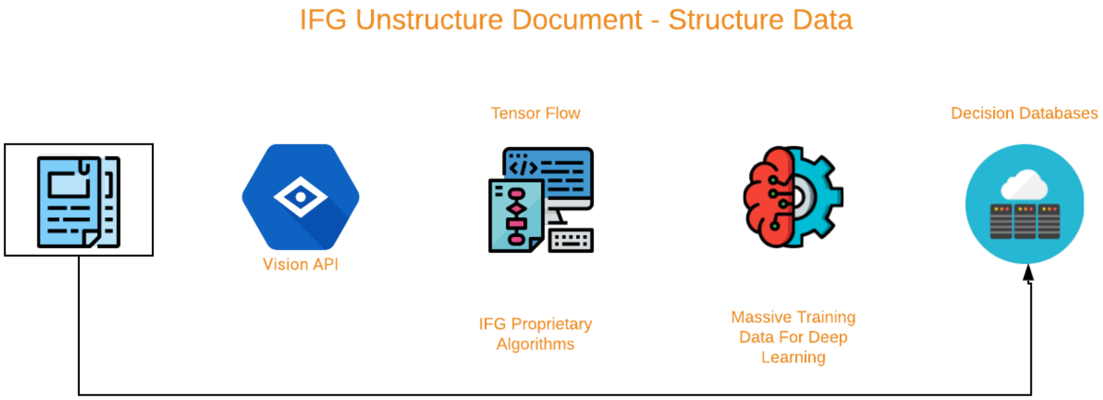
A joint Research & Development project between Google & The Interface Financial Group, designed and developed a method to extract key data fields from unstructured financial documents. As part of this project Google invited IFG's Data Team to be part of the alpha community for Google AI. Our case study is based on business to business invoices but the algorithms and methodology can be applied to other types of unstructured business and financial documents.

Invoice processing is one of the most critical tasks for the financial department of any organization. In many organizations, invoices are still examined and entered manually, a process that is slow, costly, prone to human errors, and has become a bottleneck of high-speed data processes. Especially as the number of invoices grows dramatically with the development of today's economy. While a standard list of critical fields is usually visible in almost all invoices, the choice of keywords and layout can vary largely from vendor to vendor and country to country, creating a significant challenge.

Extracting invoices will benefit the fast growing e-invoicing industry, and financiers such as Trade Finance, Asset Base Lending and Supply Chain finance platforms, connecting buyers and suppliers in a synchronized ecosystem. This environment creates transparency, which is essential for regulators and tax authorities. Ecosystems would benefit from suppliers who submit financial documents in various formats via supplier's portals; allowing these documents to be converted into data feed format and instantly.

GOALS

The goal of the invoice recognition project is to extract all the useful information from images of invoices regardless of the format. The majority of current commercially available invoice recognition techniques rely on invoices that have been directly rendered to PDF by software and that match one of a set of predefined templates. In contrast, this project starts with images of invoices that could originate from scans or photographs of paper invoices or be directly generated from software.


IFG Unstructure Document - Structure Data
Tensor Flow
Decision Databases
Vision API
IFG Proprietary Algorithms
Massive Training Data For Deep Learning

The machine learning models of our current invoice recognition system recognize, identify, and extract 26 fields of interest. These fields include invoice number, invoice date, invoice total, due date, terms, full name and address for the buyer & supplier, the party to whom payment on the invoice is made, and tabular information containing items descriptions, price, quantity, line total, and subtotal.

While a true generalized address recognition system is beyond the scope of this project, we attempt to split the address fields into subfields. We created a system that should work for most of the United States as well as several other countries. Our address recognition system attempts to extract the name of the company, the street, the street number, zip code and the city, state, and phone number.

An extra "other" field includes both information we cannot yet categorize as well as information that we do not currently believe is useful. An example of the former is the quantity field in the details table. An example of the latter would be a closing such as "Thank You for Your Business."

PROCESS

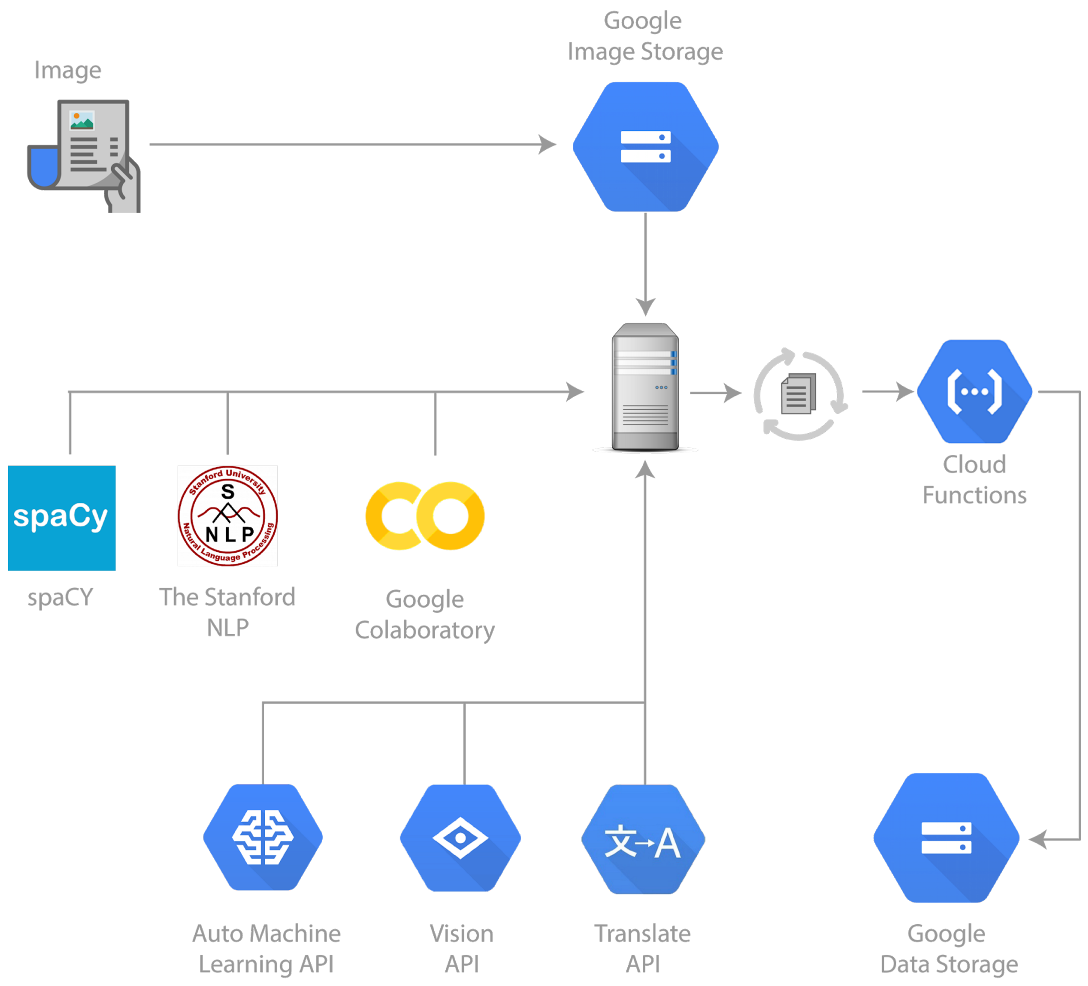
The first step in any invoice recognition project is to acquire images of invoices. Many companies consider their supply chains and hence their invoices to be confidential. Others simply do not see a benefit to maintaining scans of their invoices. The net effect is that we were unable to locate a large publicly available repository of invoice images.

What we were able to acquire was a set of line item data from invoices. With this data, we were able to synthetically generate a set of 25,011 invoices with different styles, formats, logos, and address formats. We use 20% of the invoices to train our models and then validate the models on the remaining 80%. We have extended our sample set to 38 million invoices for future training and validation. However, those additional invoices were not used in the preparation of this paper. Unfortunately, the original data set we used to generate these invoices does not belong to us, so we are not able to make our dataset publicly available.

A synthetic dataset such as ours will be representative but cover only a restricted subset of the invoices used in business today. However, since the core of our system uses machine learning instead of templates, it should work reasonably well with real invoices regardless of the format. We have restricted the numbers in our sample set to US standards for grouping and we have restricted the addresses in our dataset to portions of the US. Hence, our address splitting system will only function properly with addresses from a majority of the US and portions of a few other countries. A dataset that included other number and address formats would quickly allow us to overcome this limitation. As mentioned above, it is beyond the scope of this project to attempt to create a truly universal address recognition and splitting system.

As shown in the diagram below, the invoice recognition process is made of several distinct steps using tools developed by several different groups.

IFG Unstructured Documents Recognition Management



The first step in processing an invoice is to translate the image into text using optical character recognition (OCR). We use Google AI (Cloud Vision) for this step. Google AI outputs text grouped into phrases and their bounding boxes as well as individual words and numbers and their bounding box.

As part of our partnership with Google, we had access to alpha versions of Google AI and contact with members of the technical team. The collaboration helped improve the Vision API, in particular the processing of tabular data. In the debugging of tabular data processing it was very useful to have access to IFGs database of invoices, which helped improve tabular data processing for all Google Cloud customers in the future. Being able to identify tables has the potential to solve most of the issues identifying data in the details table included in most invoices. We expect to be able to extract the tabular information with high accuracy in the near future.

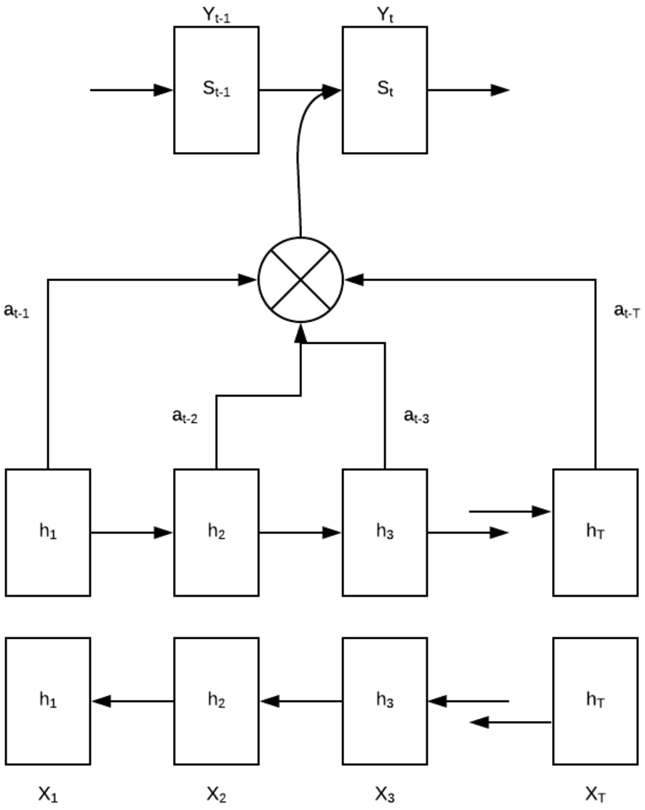
Google AI splits words on punctuation. However, some of the important fields in an invoice contain punctuation. Invoice numbers are a prime example as they can contain virtually any characters or symbols. We found it useful to split the text ourselves to avoid splitting single fields when possible. The splitting process is generally called tokenization and we used two different tokenizers from the Stanford Natural Language Processing Group. The first tokenizer splits on punctuation and keeps the punctuation with the word. This tokenizer is called wordtokenizer and is useful when the punctuation is likely part of the field. The second tokenizer also splits on punctuation, but it returns the punctuation as separate tokens. This tokenizer is called wordpunktokenizer.

Our dataset of 25,011 invoices produces approximately 8.7 million tokens. We use several modules from Keras to preprocess the token stream and reduce it to a size that is more manageable in the hardware that we used for this initial version of the system.

One particular Keras module that is worth noting is the embedding module. This module uses a dense vector representation for the data. A dense vector representation allows us to take advantage of the semantic relationships present in the data. This reduces the dimensionality of the input far more than a naive approach.

After preprocessing, the data is fed into several different machine learning models. All of the neural net models are running on TensorFlow. The non-neural net models use scikit-learn. The machine learning systems used are sequence to sequence, naive Bayes, and a decision tree algorithm. Each system has its own strengths and weaknesses and each system is used to extract different subsets of the data we are interested in. Using this ensemble model allowed us to achieve higher accuracy than any individual model.

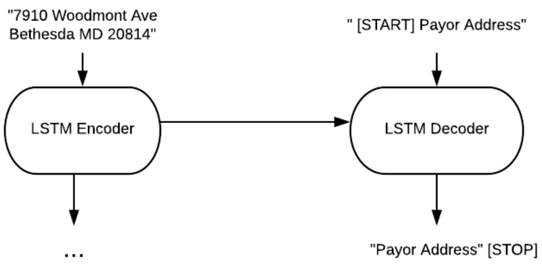
Sequence to Sequence (Seq2Seq) models use a recurrent neural network to map input sequences to output sequences of possibly different lengths. A common use of Seq2Seq neural nets is machine translation. Sequences of words in one language are the input and sequences of words in another language are the output. When all goes well, the input and output sequences have the same meaning.



Predicting the nth target word given an input

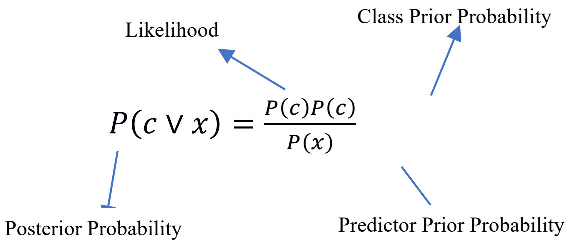
We implemented a character level sequence to sequence model for invoice id parsing. The reasoning behind utilizing character by character when word level models are more common was due to the uniqueness of the invoice number problem set as invoice numbers can be numeric, alphanumeric, alphanumeric with punctuation and so on.

We found Seq2Seq to perform very well at identifying invoice numbers. Because invoice numbers can consist of virtually arbitrary sequences of characters, we abandoned the tokenized input and focused on the text as a character string. When applied to the character stream, our Seq2Seq model is able to match invoice numbers with approximately 99% accuracy. Seq2Seq performed particularly poorly at distinguishing two letter state abbreviations from two letter street abbreviations. The best we were able to achieve was approximately 70% accuracy. This is not surprising since it can be difficult for a neural net to distinguish court (CT) from Connecticut (CT) when both are located in an address.



Because the Seq2Seq model was unable to distinguish street abbreviations from state abbreviations, a naive Bayes model was added. This model is able to distinguish state abbreviations from street abbreviations with approximately 97% accuracy.

Naive Bayes has been used by combining with n-grams to reconstruct the document and place the appropriate features in their appropriate fields at the end of the process. Even though an address is identified it must be associated with either the payor or payee in this particular use case. What precedes the actual address is of utmost importance in this instance.


Likelihood
Class Prior Probability
$P(c \vee x) = \frac{P(c)P(c)}{P(x)}$
Posterior Probability
Predictor Prior Probability

Neither Seq2Seq nor naive Bayes models were able to use the bounding box information to distinguish nearly identical fields such as payor address and payee address. A decision tree model was added to distinguish such data.

Finally, the output was post processed into a Pandas data frame to compare the output to the test data. We used cross entropy as a loss function for both accuracy and validity. As expected in a case like this accuracy varied base on the number of epochs used in training. An optimum number of epochs was discovered during testing to reach 99% accuracy or higher of element recognition in most invoices.

CONCLUSION AND FUTURE WORK

The Google AI performs exceptionally well capturing raw data from an image. Our collaboration allowed the team to focus on a highly accuracy machine learning model that processes a variety of business documents. Additionally, several well-known NLP libraries were utilized exhaustively to help prepare the output received from Google Cloud Vision for the aforementioned models. We have found the Sequence to Sequence techniques provide us the flexibility and appropriate algorithmic models that can be used for multiple applications in various vertical markets. We are looking to integrate this technology into our product offering.

Going forward we will take advantage of advances in Google AI and our extended sample set to properly process tabular data. Once all necessary fields are recognized and captured to an acceptable level of accuracy, we will extend the invoice recognition project to other forms of financial documents. Eventually we expect to be able to process any sort of structured or unstructured financial document from an image into a data feed with enough accuracy to eliminate the need for regular human intervention in the process.