Invoice Simplifier -Productivity Application to simplify your invoice analysis

Samik Biswas, Shivan Desai, Shubhangi Yadav and Rajeev Sebastian
Department of Software Engineering, San José State University
San José, CA
samik.biswas@sjsu.edu, shivan.desai@sjsu.edu, shubhangi.yadav@sjsu.edu, rajeev.sebastian@sjsu.edu

***Abstract*— In today's scenario, where time is of prime importance, decoding a complex invoice and reconciliation can be tedious and time consuming. In addition, in this cumbersome process, missing crucial details and spotting errors in the invoice can be an added overhead. To take care of all these day to day issues Invoice-Simplifier extracts key elements from your invoices, and gives you a simplified and informative summary, while also analyzing and categorizing the different products / services that drain your wallet the most. Pie charts and graphs has been included as an added advantage to get a better visual understanding of the expenditures. Our application plans to reduce all the overheads from the users' perspective and make invoice analysis a seamless experience.**

The user participation can be as simple as just uploading a snapshot of the invoice and hit the submit button. The invoice will be directly sent to our backend python data processing service that will take care of all the complex stuffs. Once done the processed data will be displayed to the user for a final confirmation. That's it, it will be ingested into the database summary reports and charts will pop up on the user dashboard.)

Index Terms — Invoice ,OCR ,Image Processing, Named Entity Recognition

I. Introduction

Invoice Simplifier is a productivity application which allows business users to easily manage their invoices and budget through the use of optical character recognition,image processing and data analytics to produce a variety rich information through various graphs.It reduces the need for manual entry of invoices as the content are extracted by the machine itself.Thus the application improves the productivity of the user and allows for easy analysis.

II. Architecture

There are 8 major components of the project. Image Processing,Optical Character Recognition,Name Entity Recognition,Categorization,Data Analysis,Web application,Backend server,Docker Container.

III. Image Processing

The input to the application is either as image or a PDF of the invoice.In the image processing stage we perform the following steps

1) Noise Removal
2) Contrast Enhancement
3) Sharpness Improvement

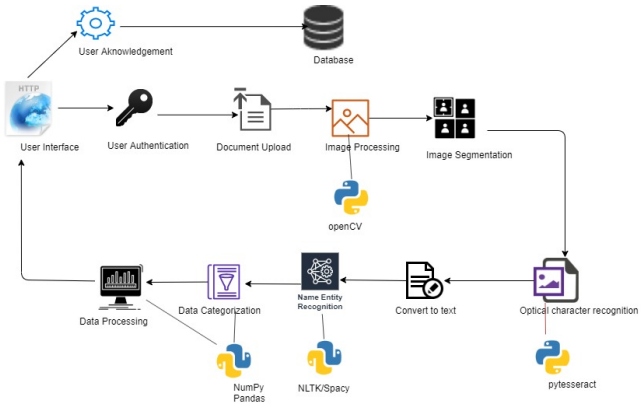

Fig. 1. Project Architecture

A. Noise Removal

Digital images are prone to various types of noise. Noise is the result of errors in the image acquisition process that result in pixel values that do not reflect the true intensities of the real scene. There are several ways that noise can be introduced into an image, depending on how the image is created. For example: If the image is scanned from a photograph made on film, the film grain is a source of noise. Noise can also be the result of damage to the film, or be introduced by the scanner itself. If the image is acquired directly in a digital format, the mechanism for gathering the data (such as a CCD detector) can introduce noise. Electronic transmission of image data can introduce noise.[1] We made use of the OpenCV library for noise removal

B. Contrast Enhancement

Contrast enhancement is a process that makes the image features stand out more clearly by making optimal use of the colors available on the display or output device. Contrast manipulations involve changing the range of values in an image in order to increase contrast.

C. Sharpness Improvement

Sharpening an image increases the contrast between bright and dark regions to bring out features.The sharpening process is basically the application of a high pass filter to an image. OpenCV was used for image sharpening.

IV. Optical Character Recognition

Optical character recognition or optical character reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image[2].The tesseract module for python allows us to extract the words present in the invoice and the n pass it on to the next stage which is the NER(Named Entity Recognition) engine.The image is first scanned and the text and graphics elements are converted into a bitmap, which is essentially a matrix of black and white dots. The image is then pre-processed where the brightness and contrast are adjusted to enhance the accuracy of the process.

The image is now split into zones identifying the areas of interest such as where the images or text are and this helps kickoff the extraction process. The areas containing text can now be broken down further into lines and words and characters and now the software is able to match the characters through comparison and various detection algorithms. The final result is the text in the image that we're given.

The process is not accurate and might need human intervention to correct some elements that were not scanned correctly. Error correction can also be achieved using a dictionary or even Natural Language Processing.Here we have made use of dictionary for extracting the items from the bill or invoice.

V. Named Entity Recognition

Named-entity recognition (NER) is a task of information extraction that seeks to locate and classify named entity mentions in unstructured text into pre-defined categories such as the person names, organizations, locations. Stanford NLTK library is used for the purpose NER.However it was found that it was a generalized engine which did not work for the specific application on invoices.For example Tesla was listed as a organization in the Corpora whereas in our use case it is supposed to be a product.[3]We managed to handle this problem by building dictionary model which associated each brand with a specific category.The basic steps in NER are the following tokenizing, part-of-speech tagging, stemming, sentiment analysis, topic segmentation, and named entity recognition.Tokenization is the process of demarcating and possibly classifying sections of a string of input characters. The resulting tokens are then passed on to some other form of processing. The process can be considered a sub-task of parsing input.Part of speech tagging is the process of identify whether a given token is a noun verb organization or type of clothing etc. Stemming is the process of finding the root word of a token. The final step of the process is the entity recognition based on the dictionary provide by Stanford NLTK library.

VI. Data Categorization

Once the invoice has been analysed and the data extracted it was grouped into particular category based on who the issuing vendor was for the invoice,the items in a particular invoice the date it was issued,the discount applied and the taxed levied on it.This allows one to create various graphs and diagram giving a quick view to the user about the expenses in his/her business. These graphs include those which show the monthly expenses that the user has had and then the discounts applied.

VII. Web Application

In order to provide an interface to the user for uploading the invoices and see the results a web application was made using ReactJS.React is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications, as it is optimal for fetching rapidly changing data that needs to be recorded. However, fetching data is only the beginning of what happens on a web page, which is why complex React applications usually require the use of additional libraries for state management, routing, and interaction with an API. The frontend actually interacts with the backend server on the basis of REST apis and JSON objects.

VIII. Backend Server

The backend server is made using NodeJS.Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts.It also hosts the MongoDb Database which stores all the information that is needed for the user interaction and for maintaining state.The NodeJs server also runs the code which is used to analyse the invoices present in the database and those uploaded.The server code exposes REST Apis for logging in , uploading invoices updating them and storing them in the database.

IX. Docker Containerization

Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels All containers are run by a single operating-system kernel and are thus more lightweight than virtual machines.The application runs on a docker container which is deployed on Amazon EC2 cluster.

X. Product Impact

1) Reduce the cost of manual effort: The application allows for automatic categorization of invoices and does not require any manual effort for digitization of paper invoices.

2) Automate processes: This involves automatic processing of invoices and reducing the errors that are caused by manual entry.

3) Easy Access and Real time monitoring: With the dashboard business owners can have access to all the payments processed and greater visibility of where the money is being sent.

4) Analytics: With all the invoices stored in one place analysis and accounting becomes easier.

XI. Conclusion

Invoice analysis is made easier by allowing the user to upload the pdfs of the invoices and the system then takes over and produces results in a matter of seconds.Small Business owners who cannot afford manual labour and time to analyze their purchases. Example Persona: John is small scale business owner who has just started his computer business. Due to the current crunch in his workforce, proper purchase management is getting ignored. John decides to use Invoice Simplifier to get his work done which simplifies his work as well as saves crucial manhours.Decoding a complex invoice and reconciliation can be tedious and time consuming. In addition, in this cumbersome process, missing crucial details and spotting errors in the invoice can be an added overhead. To take care of all these day to day issues Invoice-Simplifier extracts key elements from your invoices, and gives you a simplified and informative summary.

A. Future Improvements

There can be improvements to the application with the following.

1) Merge with existing Apps : Integrate Invoice Simplifier with existing applications like QuickBooks by Intuit which can lead to a complete end to end experience for the user.

2) Improved Analytics: Make use of better analytics by considering more of keywords and categories present in the invoice or bills.

XII. Deliveries

A. GitHub Repository

`https://github.com/SJSUFall2019-CMPE272/Invoice-Simplifier`

XIII. REFERENCES

[1] Image Noise Reduction and Filtering Techniques Abdalla Mohamed Hambal1 , Dr. Zhijun Pei2, Faustini Libent Ishabailu3 1, 2, 3Tianjin University of Technology and Education, Department of Electrical and Electronics Hexi District Tianjin, China 1310 N0

[2] Design of an Optical Character Recognition System for Camerabased Handheld Devices Ayatullah Faruk Mollah1 , Nabamita Majumder2 , Subhadip Basu3 and Mita Nasipuri

[3]Named Entity Recognition with Bilingual Constraints Wanxiang Che† Mengqiu Wang‡ Christopher D. Manning‡ Ting Liu†