Free Form Document Based Extraction Using ML

Mona Deshmukh[1] , Shruti Maheshwari[2]

[1] Associate Professor, Master of Computer Application, Vivekanand Education Society's Institute of Technology, Mumbai, India

[2] Student, Master of Computer Application, Vivekanand Education Society's Institute of Technology, Mumbai, India

Abstract: ***Information extraction is concerned with applying natural language processing to automatically extract required information from free form based text documents. Several machine learning techniques have been applied in order to facilitate the portability of the information extraction systems. The challenge is not just to extract data from scanned documents but also to extract it accurately. This paper describes a general method for building an information extraction system using properties such as tokenization, POS tagging, entity detection and dependency parsing along with supervised learning algorithms. In this method, the extraction decisions are lead by a set of classifiers instead of sophisticated linguistic analyses. A major problem incurred by many businesses today is insufficiency to leverage data from scanned documents and images. Whenever a business makes use of data which is to be captured from paper documents, manually entering data can impact the efficiency, system vulnerability and speed of carrying out of business. In such business cases, we need data entry automation that helps to extract data from scanned documents and automate document based business processes.***

Keywords: spaCy, POS tagging, tokenization, OCR engine, open NLP

1. Introduction

In manual data extraction, businesses have a data entry operator whose job is to manually read data from one document, scanned document in this case, and enter it in another desired format. This process is problematic for the following reasons: It is time-consuming, prone to error, expensive as businesses need to hire someone for the job and no real-time tracking of the data. Some businesses outsource this aspect of their business process but while outsourcing only removes the overhead from their business line, it doesn't overcome the challenges listed above. Trade companies, retail companies, service-based industries, and government-based businesses are just a few examples of different business organizations that rely on data entry services in order to run smoothly. However, data entry is not an infallible process, and there are many issues that can cause setbacks, frustration, and further problems for any business that utilizes it. One of the most common data entry problems occur during the actual data input process is error in data entry. A seemingly insignificant mistype can cause short and long term problems, leading to inaccurate records, misinformation, and disorganization. This is particularly common in instances of manual, human-based data entry. Unfortunately, even the best data entry clerk can make mistakes, which in turn can cause a lot of problems for a business. Even the best, most comprehensive data entry program can produce problems for a business. Incorrect formatting is a common issue, and can result in the right data being entered into the wrong fields. A business that deals with a large network of people may need to have multiple means of contacting their clients, thus they use a program that has several fields for addresses and phone numbers. In industries like Banking and Trading where correct data is very essential, incorrect manual data entry can have a huge impact and can harm the business. Making mistakes is an inherent part of human nature. However, in the corporate world, payroll mistakes can have some serious consequences on the line of business. The wrong numbers can result incorrect payment, leading to inefficiencies within the organization. Such mistakes might be very costly and time-consuming to rectify. Furthermore, it might at times infringe on government legislations, putting the organization at risk.

Automated Data Extraction is the more efficient, modern and preferred way of extracting data from scanned documents. Automated data entry solutions do a great job of reading scanned documents and images and then transferring that data into a different format such as excel sheet or csv. There are numerous benefits of automating data extraction process. It is faster, easier and more efficient, provides an error-free extraction with Real-time data tracking. It saves time, money and efforts and makes the process customizable which means that if, at any stage, you need to make a change in the process you can do it through automation. One of the most important qualities of information in digital form is that by its nature, it is not fixed in the way that texts are printed on paper. Digitization is the process of converting a printed item, image captured using a scanner or digital camera into a digital format and electronically storing it on a computer. It converts media into electronic forms through scanning, sampling or re-keying by various technologies. By embracing digitalization, banks can provide enhanced customer services. This provides convenience to customers and helps in saving time. Digitalization reduces human error and thus builds customer loyalty. Today, people have round-the-clock access to banks due to online banking. Managing large amounts of cash has also become easier. However, in order to leverage the last benefit of customization, the software needs to be trained and the software your business is using should have the feature of customization. If a structured document is any type of module in which the positions of the data to be extracted are precise and known in advance, an unstructured document is instead a document in which there are, however, very precise data, but their position and the their layout is not known a priori and can vary greatly between the document and the document of the same typology. Digitization in data extraction should focus on three main components: Optical Character Recognition (OCR), Natural Language Processing (NLP) and Extraction using Name entity recognition (NER).

2. Proposed Approach

In order to learn about image data extraction, document scanning and their data extraction, we need to understand what makes it so difficult to extract data from scanned documents and images. There are several reasons that make data extraction from scanned images difficult and some of them are:

- Scanned documents and images do not contain any text which can just be 'selected' with a cursor
- Extracting tables from scanned documents is tricky! Tables are basically just 'blocks of texts' and a software is needed to identify table rows and cells
- It becomes even more difficult when the data tables are spanned across multiple images and pages of the document, or when the tabular data is not in a simple row-column format (but rather nested i.e. when we have a table within a table)
- Sometimes the images are not clear i.e. the OCR software knows there is data but can't accurately read it

To accomplish this task, good Optical Character Recognition (OCR) is needed. The proposed approach is a combination of multi-voting for OCR, open NLP for Parts-Of-Speech tagging and Extraction using pattern recognition, advanced Zonal OCR, SpaCy and rule engine.

1) Multi-Voting

OCR (optical character recognition) is the use of technology to distinguish printed or handwritten text characters inside digital images of physical documents, such as a scanned paper document. The basic process of OCR involves examining the text of a document and translating the characters into code that can be used for data processing. OCR is sometimes also referred to as text recognition. Before the development of OCR programs, paper documents needed to be converted into digital copies by hand. Therefore, the main advantages of OCR technology are saved time, decreased errors and minimized effort. OCR programs can vary in their techniques, but typically involve targeting one character, word or block of text at a time. Characters are then identified using one of two algorithm:

1) Pattern recognition- OCR programs are fed examples of text in various fonts and formats which are then used to compare, and recognize, characters in the scanned document.
2) Feature detection- OCR programs apply rules regarding the features of a specific letter or number to recognize characters in the scanned document. Features could include the number of angled lines, crossed lines or curves in a character for comparison. For example, the capital letter "A" may be stored as two diagonal lines that meet with a horizontal line across the middle.

Multi-voting is a smart choice when you need to narrow down a list. That is the strength of this kind of decision making – to take a large list and pare it down to the options on the list that are the most popular among the group. The proposed approach makes using of Multi-voting mechanism where according to the document it decides which OCR technique to use for better results. We have chosen Tesseract and Omnipage as the best OCR engine options because they provide better accuracy, pre-processing and efficiency.

Tesseract is an OCR engine with support for unicode and the ability to recognize more than 100 languages out of the box. It can be trained to recognize other languages. It is available for Linux, Windows and Mac OS X. Tesseract up to and including version 2 could only accept TIFF images of simple one-column text as inputs. These early versions did not include layout analysis, and so inputting multi-columned text, images, or equations produced garbled output. Since version 3.00 Tesseract has supported output text formatting, hOCR positional information and page-layout analysis. Support for a number of new image formats was added using the Leptonica library. Tesseract can detect whether text is mono-spaced or proportionally spaced. Tesseract is probably the first OCR engine able to handle white-on-black text so trivially. At this stage, outlines are gathered together, purely by nesting, into Blobs. Blobs are organized into text lines, and the lines and regions are analyzed for fixed pitch or proportional text. Text lines are broken into words differently according to the kind of character spacing. Fixed pitch text is chopped immediately by character cells. Proportional text is broken into words using definite spaces and fuzzy spaces. Recognition then proceeds as a two-pass process. In the first pass, an attempt is made to recognize each word in turn. Each word that is satisfactory is passed to an adaptive classifier as training data. The adaptive classifier then gets a chance to more accurately recognize text lower down the page. Since the adaptive classifier may have learned something useful too late to make a contribution near the top of the page, a second pass is run over the page, in which words that were not recognized well enough are recognized again. A final phase resolves fuzzy spaces, and checks alternative hypotheses for the x-height to locate small-cap text.

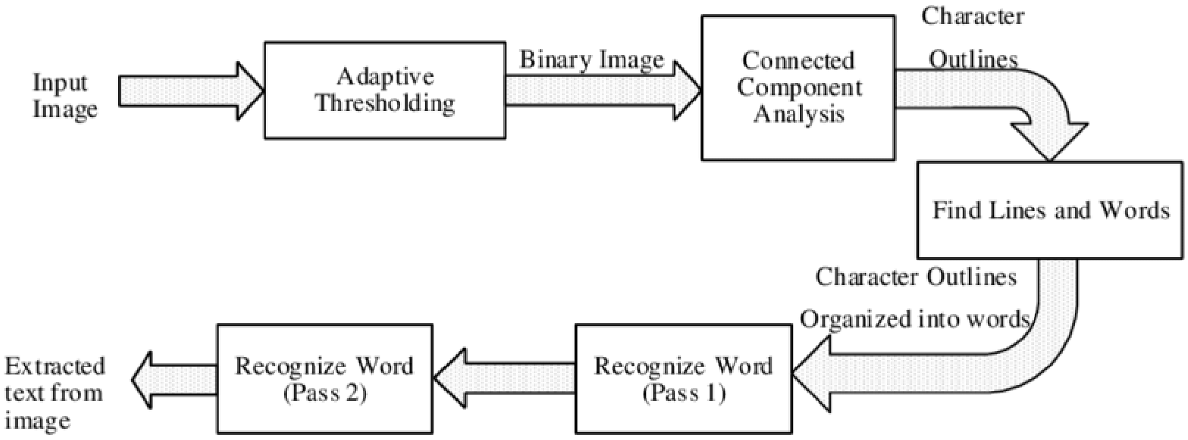


Figure 1: Architecture of Tesseract OCR

Accuracy of a OCR system depends on the quality of input document. Sometimes the output from OCR systems is often quite “noisy”. Post processing is done on the text to correct the noise. The average time taken to recognize 20 words is 350ms and that of 100 words is 500ms. The accuracy of the OCR system also depends on the camera used to capture the raw image of the document. Various factors affecting the quality are: Focus of the camera, resolution of the picture, amount of noise present etc. Tesseract engine achieved an average accuracy of 93%.

OmniPage uses optical character recognition (OCR) technology to transform text from scanned pages or image files into editable text for use in your favorite computer applications. In addition to text recognition, OmniPage can retain the following elements and attributes of a document through the OCR process. Graphics (photos, logos) Form elements (checkboxes, radio buttons, text fields) Text formatting (character and paragraph) Page formatting (column structures, table formats, headings, placing of graphics) Documents in OmniPage A document in OmniPage consists of one image for each document page. After you perform OCR, the document will also contain recognized text, displayed in the Text Editor, possibly along with graphics, tables and form elements.

2) Apache OpenNLP

OpenNLP supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, language detection and coreference resolution. Following are the notable features of OpenNLP-

- Sentence detection − Open While processing a natural language, deciding the beginning and end of the sentences is one of the problems to be addressed. This process is known as **S**entence **B**oundary **D**isambiguation (SBD) or simply sentence breaking.
- Named Entity Recognition (NER) − Open NLP supports NER, using which you can extract names of locations, people and things even while processing queries. To perform various NER tasks, OpenNLP uses different predefined models namely, en-nerdate.bn, en-ner-location.bin, en-ner-organization.bin, en-ner-person.bin, and en-ner-time.bin. All these files are predefined models which are trained to detect the respective entities in a given raw text. The opennlp.tools.namefind package contains the classes and interfaces that are used to perform the NER task.
- Tokenization − To tokenize the given sentences into simpler fragments, the OpenNLP library provides three different classes −SimpleTokenizer that tokenizes the given raw text using character classes. WhitespaceTokenizer that uses whitespaces to tokenize the given text. TokenizerME that converts raw text into separate tokens. It uses Maximum Entropy to make its decisions.
- Summarize − Using the summarize feature, you can summarize Paragraphs, articles, documents or their collection in NLP.
- Searching − In OpenNLP, a given search string or its synonyms can be identified in given text, even though the given word is altered or misspelled.
- Tagging (POS) − Tagging in NLP is used to divide the text into various grammatical elements for further analysis.
- Information grouping − This option in NLP groups the textual information in the content of the document, just like Parts of speech.
- Natural Language Generation − It is used for generating information from a database and automating the information reports such as weather analysis or medical reports.
- Speech recognition − Though it is difficult to analyze human speech, NLP has some builtin features for this requirement.

3) Extraction

In the proposed approach, data from unstructured documents are extracted with the help of Zonal OCR, spaCy and a rule engine.

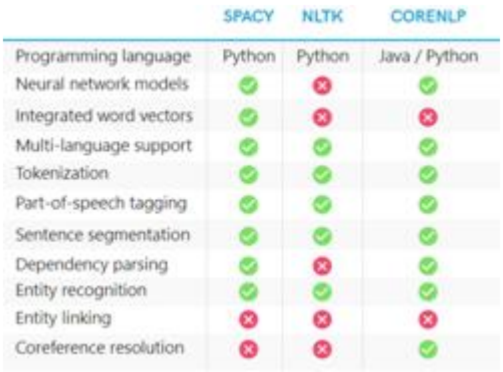
Zonal OCR: Zonal Optical Character Recognition (OCR), also sometimes referred to as Template OCR, is a technology used to extract text located at a specific location inside a scanned document. In this article we’ll explain how Zonal OCR works and how it can be used to automate data-entry workflows. Most of today’s document and PDF scanning offer out of the box Optical Character Recognition (OCR) capabilities which convert your scanned images (JPG, PNG, or TIFF files) into searchable and editable PDF documents. In some cases, a simple OCR system is however not enough and you need to level up your game. For example if you are not interested in the whole text of a document, but rather want to pull certain text elements which are located at specific positions. This is when a technology called “Zonal OCR” (also referred to as Template OCR) comes into play. Zonal OCR basically allows to extract only important data fields from a scanned document and then store the extracted values in a structured database. One popular use case for Zonal OCR is to convert PDF to Excel or Automated Invoice Processing. OCR is used to convert scanned documents into searchable and editable documents. But having the whole text of the document accessible is only the first step. Zonal OCR goes one step further. Instead of only converting your scanned images into text, a Zonal OCR software system can be trained to understand the structure and hierarchy of you document. By defining “zones”, it is possible to teach a zone based OCR system to distinguish certain data fields from each other. The following cases cannot be handled by a simple Zonal OCR system:

- Extracting compound data fields (e.g. First + Last Name, Postal Address)
- Repeating data fields (e.g. Multiple product numbers)
- Table data
- Data fields with variable positions (e.g. Invoice totals)

For the above reasons, the approach uses SpaCy and a rule engine to overcome the cases that cannot be handled by Zonal OCR.

SpaCy:
spaCy is the best way to prepare text for deep learning. It interoperates seamlessly with TensorFlow, PyTorch, scikit-learn, Gensim and the rest of Python's awesome AI ecosystem. With spaCy, you can easily construct linguistically sophisticated statistical models for a variety of NLP problems. It provides Named entity recognition, supports 49+ languages, 16 statistical models for 9 languages, pre-trained word vectors, POS tagging and labeled dependency parsing. It is an efficient binary serializer and provides a robust, rigorously evaluated accuracy.

Following figure explains the difference between functionalities offered by spaCy, NLTK and CoreNLP



	SPACY	NLTK	CORENLP
Programming language	Python	Python	Java / Python
Neural network models	✓	✗	✓
Integrated word vectors	✓	✗	✗
Multi-language support	✓	✓	✓
Tokenization	✓	✓	✓
Part-of-speech tagging	✓	✓	✓
Sentence segmentation	✓	✓	✓
Dependency parsing	✓	✗	✓
Entity recognition	✓	✓	✓
Entity linking	✗	✗	✗
Coreference resolution	✗	✗	✓

Figure 2: Comparison of the functionalities offered by spaCy, NLTK and CoreNLP.

Simple copying-pasting is not possible as there is no text data to select from. And even if the document was OCRed properly, copy-paste is a manual process and when businesses deal with huge chunks of data, automation is the key. The most classic example of unstructured document in which it is very easy to come across on a daily is represented by *bills*: although we know a priori that each invoice is the *business name* of the supplier, the date, the *number* progressive, the *taxable*, the *VAT* and the *total*, we cannot know in advance where these data are located.

The approach that is used to solve this problem is rather than starting from a spatial definition, part by a logical definition of the data. In practice, the data to read are defined, and then identified by a series of specific attributes, such as, for example, key words next to them, formatting type awaited, relative position, presence or absence of graphical elements, the criteria of cross-validation check, and so on. In practice, the software instructs you to "think" like humans do: in fact, when we look on a bill given the TOTAL DOCUMENT we are naturally inclined to look at the bottom right of the sheet, maybe we focus on a box particularly evident or marked and try as "test" the words "TOTAL DOCUMENT" O "INVOICE AMOUNT" or "TOT. INVOICE". In the same way it acts a system for processing of unstructured documents: this is based on our information, on the basis of the rules properly reset, which must then be defined in a precise and exhaustive. The basis of these features is the use of optical character recognition (*OCR*) of the entire document together with a robust algorithm of layout analysis: the combined use of these two tools makes it possible to identify blocks of text, vertical lines, horizontal and text elements with their confidences, with the possibility of verifying whether or not the logical conditions imposed on the research data on the page. To make it even more accurate processing of unstructured documents is also possible to combine the two strategies described above: if the system is able to associate the document to be treated to a template known, is treated as a structured document, otherwise it is treated as a document unstructured and processed equally.

3. Conclusion

There are many existing approaches that may provide good OCR quality, or good data extraction, however the proposed approach is better in terms of efficiency, reliability and accuracy. It handles OCR quality by multi-voting, noun recognition using NLP and key-label recognition using spaCy. It detects all the text segments having some possibility to be part of the output template, and selects from the set of candidate text segments, those that are useful to produce the extraction output.

References

[1] Cowie, J., Lehnert, W.: Information Extraction. Communications of the ACM, Vol. 39, No. 1 (1996) 80-91

[2] Freitag, D.: Machine Learning for Information Extraction in Informal Domains. Ph.d. thesis, Computer Science Department, Carnegie Mellon University, (1998)

[3] https://spacy.io/
[4] A Machine Learning Approach to Information Extraction in Lecture Notes in Computer Science 3406:539 547
[5] https://opensource.google.com/projects/tesseract
[6] https://opennlp.apache.org/