Automatic Information Extraction from Electronic Documents using Machine Learning

Nishanthan Kamaleson, Dominique Chu, and Fernando E. B. Otero

School of Computing, University of Kent, Canterbury, UK
{N.Kamaleson, D.F.Chu, F.E.B.Otero}@kent.ac.uk

Abstract. The digital processing of electronic documents is widely exploited across many domains to improve the efficiency of information extraction. However, paper documents are still largely being used in practice. In order to process such documents, a manual procedure is used to inspect them and extract the values of interest. As this task is monotonous and time consuming, it is prone to introduce human errors during the process. In this paper, we present an efficient and robust system that automates the aforementioned task by using a combination of machine learning techniques: optical character recognition, object detection and image processing techniques. This not only speeds up the process but also improves the accuracy of extracted information compared to a manual procedure.

Keywords: OCR · Layout Analysis · Image Detection · Information Extraction

1 Introduction

In the last few years, the applications of machine learning (ML) have displayed a strong positive impact in various industrial sectors. With the help of ML, most of the complex and tedious tasks have been transformed into smartly automated tasks. For example, ML-enabled chatbots have started assisting the customers with product related questions, henceforth, the sales cycle became more efficient. Since the interventions of ML, a paradigm shift has started towards digitalisation. As a result, many sectors decided to move beyond paper documents and adopt digital document management systems. Nevertheless, use of paper documents is still widespread within some sectors such as financial and legal. The key information (KI) from such documents need to be extracted and stored electronically to adhere to the paradigm of digitalisation. This process has been done manually by humans until the introduction of optical character recognition (OCR), which makes it possible to convert text within an image into machine readable text.

As OCR systems have evolved in recent years, the recognition of handwritten and printed text from scanned images has significantly improved. However, extracting the values of interest from varying complex structures still is a challenging problem. Previously, there have been many methods introduced to tackle this problem in various contexts [3, 4, 6–8]. However, they are either very generic or too specific and complex to solve the problem of extracting values from more than 2000 different types of electronic documents, where the number of types could possibly increase over time.

In this paper, we present a novel system for KI extraction, named as *Doctract*, which is effective and robust in handling complex documents of various structures. *Doctract* is composed of multiple sophisticated tools and techniques including state-of-the-art open-source OCR engine *Tesseract*[1] and object detection model *YOLOv4*.[2] It utilises the features of documents, such as text, layout, position and visual cues, to obtain a semantic representation that enables the efficient and precise extraction of KI. For a certain document type, these features should be defined for every KI of interest using a document template file. Later, when processing a document, Doctract system will refer to the document template file and extract KI accordingly.

The rest of the paper is organised as follows. Section 2 describes the problem of extracting key information. Section 3 describes prior works related to key information extraction from electronic documents. Section 4 details the proposed system and how it uses machine learning techniques to solve the problem. The evaluation of our approach is discussed in section 5, followed by conclusion in section 6.

2 Problem: Extracting Key Information

The problem of extracting key information that we are addressing in this paper can be exemplified as follows. An anonymous company collects different documents that enclose KI of their clients from external sources with their consent. In their current work flow, such KI are manually extracted and uploaded into their systems for record keeping. This process consumes on average between 5 to 10 minutes per document with up to 15% likelihood of human error (85% accuracy). Each of these external sources could have more than one type (structure) of document and there are about 2000 different external sources, therefore a KI can be located at different sections of the document and equivalent KI can be identified by different labels (e.g., member number, account number, member ID).

Our goal is to automate their KI extraction process to improve the processing time and eliminate the need for human resources while making sure the likelihood of error is below 15%. As the processing of documents can be parallelised,

[1] https://opensource.google/projects/tesseract
[2] https://github.com/AlexeyAB/darknet

our focus is only on extracting the precise KI as more computational resources could be allocated to speed up the overall process. In this paper, the document corpus was created from 6 different external sources where 1 of them has 3 different document types whilst the rest has only two. Overall, we obtained 50 document instances with at least 3 samples for each of the document type. Although we are working with scanned documents, the proposed system is capable of handling any electronic documents.

3 Related Work

The extraction of KI from scanned images of complex structured documents has been explored and studied extensively by many researchers [4,5,7].

Ishitani [5] proposed a document analysis method which enables automated extraction of KI and their logical relationship from scanned documents using a set of pre-stored models, which defines the type of each document. The KI extraction pipeline of this particular work is very similar to our proposed solution, where their models are referred in our context as document templates. However, their models assume that the documents have fixed structure for a given type. In our documents, KI is not only spanned across multiple pages but also their geometric positions are likely to change within a same type of document.

Peanho et al. [7] proposed another similar solution where their document model is a set of relations between text segments which describes positional and geometric relations. Although their document model considers significant positional displacement of KI within a same type of document, it assumes the document model is always known a priori. When the document model is unknown, their system applies all the models to the corresponding document and selects the appropriate model. This approach is not efficient in our case as we are dealing with more than 2000 types of document layouts.

Commercial products such as Google's Cloud Vision[3] and Amazon's Textract[4] have good performance in carrying out OCR tasks. However, the flexibility of their layout analysis is limited. For example, Textract can identify text, forms and tables but there is not an option to selectively extract only the values of a particular section of the text or table from documents. Additionally, they do not provide a facility to extract the same information from documents with different layout or structure. In this work, we overcome these limitations by introducing document templates, which allows a user to define the relative location of a KI, together with an automatic document detection to select the specific template to extract the information.

[3] https://cloud.google.com/vision
[4] https://aws.amazon.com/textract/

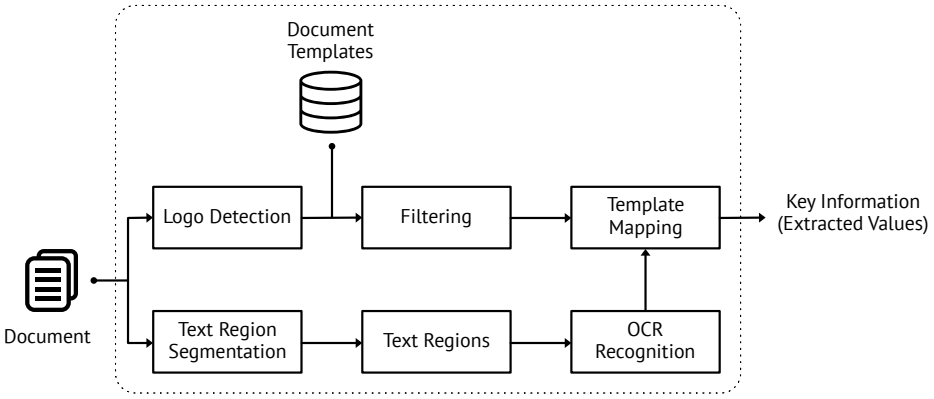

Fig. 1: Pipeline of key information extraction process

4 Proposed System

This section describes the architecture of the proposed Doctract system and the underlying tools and techniques used to achieve our goal, including the state-of-the-art object detection architecture, image processing techniques used in computer vision and an optical character recognition (OCR) tool.

4.1 Overview

The Doctract system performs the KI extraction process over three phases. During the first phase, it detects logos within the given electronic document, henceforth, identifies the respective external source. In the second phase, it initially performs text segmentation and then constructs a dictionary using the OCR outputs and bounding box coordinates of those respective text segments. Finally, it selects all document templates matching the identified logo and maps each of them against the constructed dictionary. Thereafter, it picks the appropriate template based on confidence scores, which measures the relevance of a document template to the document. Finally, returns the KI that was extracted using the best template. Figure 1 shows the pipeline of the Doctract System including all of these phases.

4.2 Object Detection

Image classification [2] is the process of categorising and labeling groups of pixels within an image based on a set of rules. Image localisation [9] is a regression problem where the output is x and y coordinates around the object of interest to draw bounding boxes. Object detection is a complex problem that combines the concepts of both image localisation and classification to return bounding boxes around all objects of interest and assign a class to them.

When classifying an electronic document into a certain type, we would normally consider using image classification techniques. However, we have access to only a small number of documents and their structures vary significantly even within the same document type. Therefore, we decided to focus on image objects that uniquely identifies the type of a given document. Based on our investigation, we learned that the logos in these documents are a good candidate to identify the external source as they do not change often.

However, these logo objects are too small comparing to the size a document page. Unlike image classification, the likelihood of preserving the classification signal is very high in object detection as it generates features from images at more fine-grained level. In our particular scenario, therefore, we could achieve better performance with object detection over the image classification even when we are not interested in the exact location or counts of the object.

As we have small number training image samples, it is necessary for us to augment our existing training dataset so that our machine learning model will be able to learn from a wider array of situations. The authors of the YOLOv4 [1], a widely used object detector, built in a set of techniques called Bag-of-Freebies, which includes some popular techniques, such as data augmentation, random cropping and shadowing, to improve the accuracy and robustness of the model during training and post-processing. Therefore, we decided to use YOLOv4 as our object detector, which in our case eliminated the necessity to augment the training data further.

4.3 Optical Character Recognition

Optical Character Recognition (OCR) can be quite sensitive to the quality (resolution) of the document. The rudimentary document quality standards which we recommend to pay attention to when scanning the documents are the dot per inch (dpi), brightness and contrast level. The widely used rule of thumb for dpi and brightness levels are between 300 to 600 dpi and 50% brightness level, respectively. Keeping the dpi below 300 may produce unclear and incomprehensible results whilst keeping the dpi above 600 will lead to only larger output files and no further improvements in the quality of the results. The brightness levels that are too high or low could negatively impact the OCR quality. However, when it comes to contrast level, only the lower settings would result in poor OCR accuracy. Therefore, keeping the contrast level above 50% is recommended. We can improve the accuracy of an OCR process either by maintaining the list of document quality standards we mentioned before at the acquisition stage and by applying a combination of image processing techniques as appropriate at the preprocessing stage. We will discuss the commonly used image processing techniques and how we leveraged them to achieve our goals in the next section.

There are currently plenty of commercial and open-source OCR tools available in the market. Cloud services from Google (Google Vision API), Microsoft (Azure Computer Vision API)[5], Amazon (Textract API) are providing their own OCR models as subscription services. Tesseract[6], GOCR[7] and CuneiForm[8] are widely used open-source OCR Tools. Among the open-source OCR Tools, the performance of the Tesseract OCR engine is comparable to the commercial tools. Therefore, we decided to select Tesseract as the OCR tool of our choice to perform KI extraction in the Doctract system.

4.4 Image Processing

The main objective of the preprocessing stage in the OCR process is to improve the readability of the document for the OCR system to recognise a character/word from the background. Most commonly used image preprocessing techniques are binarization, skew correction, noise reduction. However, in this specific phase, we apply the following techniques in the given order:

1. Binarisation
2. Noise removal
3. Horizontal and vertical line detection
4. Add vertical spaces before and after horizontal lines
5. Remove horizontal and vertical lines

First, we converted the RGB image into a grayscale image and then we applied adaptive thresholding to binarize the image. Any noises introduced during the scanning process were reduced using the morphological operations such as opening and closing. Most of our document instances included tables where the paddings between cells were too small to perform precise segmentation. Thus, we added spaces before and after the detected horizontal lines which allowed us to easily separate the cells from each other during the text segmentation phase. Moreover, we decided to remove all the lines as we are only interested in the textual content.

Later, we inverted the image and applied the morphological transformation operation called dilation using a rectangle kernel. As a result of this process, we will obtain white rectangle blocks which could possibly represent the location of a sentence, a phrase or a word. We finally extract the bounds of these white rectangle blocks. The knowledge of these bounds are not only used to distinguish the text segments but also used during the application of document templates to compare the relative positions of key text segments against each other.

[5] https://azure.microsoft.com/en-gb/services/cognitive-services/computer-vision/
[6] https://github.com/tesseract-ocr/tesseract
[7] http://jocr.sourceforge.net/
[8] https://en.wikipedia.org/wiki/CuneiForm_(software)

```
<Templates>
  <Company name="ABC Ltd.">
    <Document id="1" name="Form">
      <Term name="Start Date">
        <DataType>Date</DataType>
        <Content type="key-value">
          <Key>Commencement Date</Key>
          <Context>
            <Neighbour>
              <Detail>
                <Text>Job Title</Text>
                <Location>Above</Location>
              </Detail>
            </Neighbour>
          </Context>
        </Content>
      </Term>
    </Document>
  </Company>
</Templates>
```

Listing 1.1: Sample Document Template

4.5 Document Templates

The templates for documents were introduced with the intention of locating KI within a given document. We realised that during an extraction process, we have to deal with different types of values such as currency, dates, names and IDs. In addition to this, our values could be found either as a key-value pair (e.g. "Start Date: 01/02/2019") or located within a cell of a table. Finally, we decided to include the contextual information of neighbours as constraints to increase the confidence. Document templates are specified in XML – the XML code snippet shown in Listing 1.1 is an example for an application form of fictitious "ABC Ltd." company.

In this application form, we are aiming to extract the corresponding value for the term "start date". It is also given that the value's data type is "Date". As this value is expected to appear as a key-value pair in the document, we should look for that particular key in the dictionary which we constructed during the image processing phase. Later, the algorithm verifies whether the "Job Title" is above this particular key. If the value satisfies the neighbour constraint, Doctract looks for the values immediately on the right or bottom of the given key and verifies whether it is a type of "Date" or not; if so, returns the value.

We have defined a metric which allows us to evaluate the performance of our Doctract system in extracting these values. Let T be the number of times a value is repeated, the confidence score C for the i^{th} term (where $i < n$ and n is the total number of terms) is defined as below:

$$C_i = \frac{T_i^2}{\sum_{j=0}^{n} T_j^2} \tag{1}$$

For example, assume we are trying to extract a value for the term "name" from an application form. The Doctract system extracted the values "David" and "Tom" from the given document three times and one time, respectively. As "David" repeated more times, we are more confident that "David" is likely to be the right value. If we substitute these values into Equation 1, we get a 90% confidence score that "David" is the correct value.

4.6 Algorithm

The Doctract algorithm illustrated in Algorithm 1, begins with identifying the logo located within the document. If the logo detection is successful, every page in the document will be preprocessed and the relevant document template will be pulled from the templates XML file. Thereafter, the algorithm loops through each of these preprocessed pages and detects the elements such as words/phrases/sentences. Using the coordinates of the bounding boxes of these elements, the original page is cropped and passed to the Tesseract OCR tool for extracting the text within those bounding boxes.

Once the text is extracted, a dictionary will be constructed where the bounding boxes and the extracted texts are keys and values, respectively. Finally, the template is applied on the constructed dictionary to extract the values of interest based on the relative locations defined in the templates. The results from all the pages are aggregated and returned. If multiple values that satisfies the specification of a given term have appeared within the document then the most frequent one is selected.

5 Evaluation

In this section, we present the evaluation results of the proposed system. The performance was measured comparing the values extracted by the system against the actual values in the documents. As these documents hold sensitive information, we keep the context as well as the values from these documents anonymous. We also discuss details of the runtime environment of the proposed system and the steps carried out to perform the experiment. Our dataset contained 50 documents from 6 different providers and each provider had 2

Algorithm 1 Doctract algorithm

```
Require: Scanned document d
l := DetectLogo(d)
if l in supportedDocs then
    globalResults := {}
    prepDoc := preprocess(doc)
    t := findTemplate(l)
    for page in prepDoc do
        bounds := extractBound(page)
        d := {}
        for b in bounds do
            roi := prepDoc.crop(b)
            text := tesseract.read(roi)
            dictionary.add(roi, text)
        end for
        pageResults := applyTemplate(d, t)
        globalResults.agg(pageResults)
    end for
    return globalResults
else
    return {}
end if
```

document types except provider "D" which had 3 different document types. In total, we created 13 different document templates.

All experiments were carried out on a system of Intel(R) Core(TM) i9-9980HK CPU @ 2.40GHz (16 CPUs) with 32 GB RAM and NVIDIA GeForce GTX 1650 4GB Graphics card.

5.1 Logo Detection Model Accuracy

In Section 4, we discussed in detail why the object detection approach is a better fit than image classification in our context. In order to train any supervised machine learning models, we need labelled data. For object detection task, images with corresponding bounding box coordinates and classes are the labelled data. As we have obtained only a few scanned documents, we created our own training dataset using these documents as well as images of logos retrieved from the internet.

All of these documents were converted into images and then the logos were annotated with the help of an open-source image annotation tool, known as LabelImg[9]. In our training dataset we labelled six different logos, are for each different external source, and created around 100 samples for each of these

[9] https://github.com/tzutalin/labelImg

Table 1: Accuracy of Extracted Key Information From the Scanned Documents. In this experiment, Doctract confidence score is calculated using Equation 1 for each extracted value and then averaged across all the documents. Accuracy refers to whether the value extracted by Doctract is same as the value present in the document. Highlighted value refers to the case when Doctract extracted an incorrect value from one of the "D01" documents due to poor scanned quality of the document.

Logo	Doc Type	Docs Count	No. of Extracted Values per Doc	Logo Detection Accuracy	Average Confidence	Average Accuracy
A	A01	3	5	100%	100%	100%
	A02	3	5	100%	100%	100%
B	B01	7	5	100%	100%	100%
	B02	3	5	100%	100%	100%
C	C01	3	5	100%	100%	100%
	C02	10	5	100%	100%	100%
D	D01	3	5	100%	**100%**	**93.3%**
	D02	3	3	100%	100%	100%
	D03	3	5	100%	100%	100%
E	E01	3	5	100%	100%	100%
	E02	3	5	100%	100%	100%
F	F01	3	5	100%	100%	100%
	F02	3	5	100%	100%	100%

logos. *YOLOv4* achieved the mean average precision of 90% in 3000 iterations on our logo dataset.

5.2 Performance of Doctract System

We evaluated the performance of the Doctract system by comparing the results against the manually extracted values. In this context, these values can be a type of currency, ID or date. Table 1 shows the results of evaluation against 6 different logos (each logo represents an external source) and 11 different types of documents. Our results show that for all the documents the Doctract system accurately extracted the expected values, apart from *D01*. In the case of document type *D01*, the Doctract system is 100% confident that the values it extracted were the right ones. However, during the manual inspection, it was revealed that one of the extracted values was incorrect due to an erroneous reading by the OCR tool.

Table 2: Performance of Doctract System Across Different Phases

Document Structure	Logo Detection (secs/page)	Preprocessing and OCR (secs/page)	Application of Template (secs/page)
with Lines	2.13	17.27	0.006
without Lines	2.09	12.10	0.006

Table 2 shows the time taken to process a page within the three phases of the extraction process. From the table, we can see that the presence of lines in a page does not impact the time taken to detect the logo or the application of a template as they do not deal with the lines. On the other hand, the time of the second phase is increased by a few seconds as it is the place where the lines in a page are identified and removed so that an OCR system can read the values with more clarity.

Our corpus mostly includes documents of pages 6 to 40. The proposed Doctract system was able to extract the key information from the smallest one within 2 minutes whilst it took 10 minutes to process the largest one. In general, it takes about 5 to 10 minutes for an experienced person to complete this task. As we can parallelise the processing at document level, the Doctract system can certainly improve the efficiency of the existing workflow.

6 Conclusion

Our main objective in this work was to improve the existing work flow of manually extracting key information from electronic documents. Henceforth, we have proposed a system, called Doctract, as a solution to aid the existing work flow. Our system was able to recognise different types of documents from various external sources and extracted KI with the support of document templates. Our evaluation shows that Doctract can extract information with high accuracy (99.4%) across 50 real-world documents.

Our future work concentrates on developing a graphical user interface to assist users with generating document templates with ease. In addition to this, we are also exploring in the direction of leveraging the power ML to automate the generation of templates. Currently, adding a new provider requires the collection of logo samples and re-training the logo detection model, which can be a time consuming task. Therefore, we are exploring a more efficient way to detect the type of document to make the system more scalable and perform experiments with a larger set of documents.

References

1. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
2. Druzhkov, P., Kustikova, V.: A survey of deep learning methods and software tools for image classification and object detection. Pattern Recognition and Image Analysis **26**(1), 9–15 (2016)
3. Hirano, T., Okano, Y., Okada, Y., Yoda, F.: Text and layout information extraction from document files of various formats based on the analysis of page description language. In: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). vol. 1, pp. 262–266. IEEE (2007)
4. Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., Jawahar, C.: Icdar2019 competition on scanned receipt ocr and information extraction. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1516–1520. IEEE (2019)
5. Ishitani, Y.: Model based information extraction and its application to document images. In: Proceedings of the Workshop on Document Layout Interpretation and its Applications (2001)
6. Meier, R., Urbschat, H., Wanschura, T., Hausmann, J.: Methods for automatic structured extraction of data in ocr documents having tabular data (Feb 2 2016), uS Patent 9,251,413
7. Peanho, C.A., Stagni, H., da Silva, F.S.C.: Semantic information extraction from images of complex documents. Applied Intelligence **37**(4), 543–557 (2012)
8. Takasu, A., Aihara, K.: Quality enhancement in information extraction from scanned documents. In: Proceedings of the 2006 ACM symposium on Document engineering. pp. 122–124 (2006)
9. Vaillant, R., Monrocq, C., Le Cun, Y.: Original approach for the localisation of objects in images. IEE Proceedings-Vision, Image and Signal Processing **141**(4), 245–250 (1994)