# CHECKBOX DETCTION


## Abstract

The Portable Document Format (PDF) is the most used file format for online documents. The absence of effective means to extract text from these PDF files in a structured manner presents a significant challenge for developers. Our paper describes the construction and performance of a system that extracts text blocks from PDF-formatted documents and classifies them into logical units based on prior hand-made annotations that characterize specific sections. We also introduce new state-of-art check-box detection schema using RetinaNet deep-learning neural network.


## Problem Statement

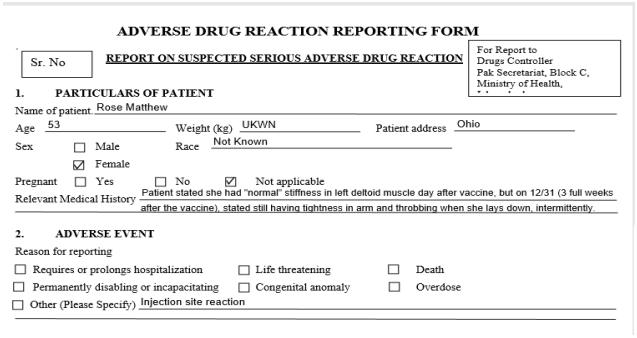
The purpose of the document analysis project is to create a web application, which will disaggregate multiple pdf format documents of a same type and reaggregate the information within in a manner that would make it identifiable, the retrievable and easily readable. Generally, data in the pdf documents is semi-structured. As opposed to well-structured data, which conforms to a schema or data model and can be queried using structured query language to answer questions, it does not adhere to any rigorous format.

## Solution

To receive structured data as described in problem statement, we propose the following steps:

| **Step 1:** Annotate single document of an each type | **Step 2:** Extract data from multiple documents in accordance with initial annotation | **Step 3:** Identify only information from checked boxes for extraction | **Step 4:** Create structured data in a csv format |
|---|---|---|---|

**Step 1** of the algorithm is to annotate a single document of a given type. See Figure 1 for examples of uploaded and annotated documents.



ADVERSE DRUG REACTION REPORTING FORM

Sr. No

REPORT ON SUSPECTED SERIOUS ADVERSE DRUG REACTION

For Report to
Drugs Controller
Pak Secretariat, Block C,
Ministry of Health,

1. PARTICULARS OF PATIENT

Name of patient: Rose Matthew

Age: 53 Weight (kg): UKWN Patient address: Ohio

Sex: ☐ Male ☑ Female Race: Not Known

Pregnant: ☐ Yes ☐ No ☑ Not applicable

Relevant Medical History: Patient stated she had "normal" stiffness in left deltoid muscle day after vaccine, but on 12/31 (3 full weeks after the vaccine), stated still having tightness in arm and throbbing when she lays down, intermittently.

2. ADVERSE EVENT

Reason for reporting

☐ Requires or prolongs hospitalization ☐ Life threatening ☐ Death

☐ Permanently disabling or incapacitating ☐ Congenital anomaly ☐ Overdose

☐ Other (Please Specify) Injection site reaction

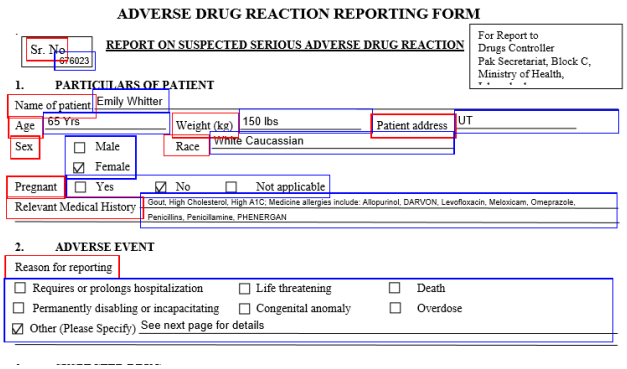



Figure 1: An example of an unannotated document (upper left) and annotated document (lower right).

The document should be annotated so that the key and value pairs would be identifiable, I.e. Name: Smith, Age: 58 years, where key is the question, and value is the corresponding answer. In picture 1 it is apparent that we have chosen to annotate keys with red boxes, while the values with blue boxes.

**Step 2.** Using a python library fitz, we can detect the red and blue boxes and extract their corresponding the coordinates. We can also detect the text within the boxes and extract them into any python data structure. Thus, keeping the coordinates of the boxes, we can receive the information from the boxes from any document of the same type. As you can see in Figure 3, the data from three analogous documents for three different patients was received having coordinates of annotation boxes of a single document.

| | Sr. No | Name of patient. | Age | Weight (kg) | Patient address | Sex | Race | Pregnant | Relevant Medical History | Reason for reporting | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | | Rose Matth... | 53 | UKWN | Ohio | MaleFemal... | NotKnown | YesNo Not... | Patients t... | Requireso... | ... |
| 1 | [676023] | [Emily Whitter] | [65 Yrs] | [150 lbs] | [UT] | [Male, Female] | [White Caucassian] | [Yes No Not applicable] | [Gout, High Cholesterol, High A1C; Medicineal... | [Requires or prolongs hospitalization Life thr... | ... |
| 2 | [676158] | [Hayden Blank] | [59] | [U] | [New York City] | [Male, Female] | [African American] | [Yes No Not applicable] | [Season Allergies (Pollen)., Other Medication ... | [Requires or prolongs hospitalization Life thr... | ... |

Figure 3: Structured data received from three documents of the same type.

However, to retrieve only information from a checked box is yet another challenge. In figure 3 we can observe for section “Sex” both options – male and female. Information from the annotated boxes is retrieved fully irrespective of the checked box in front of only one of the options.

**Step 3.** To detect which box is checked, we suggest using RetinaNet deep learning network. We propose a completely new solution to the problem of checkbox detection: instead of detecting the box itself, we trained the network in a manner, that it could identify the bounding box of the text near the checked box (see Figure 4). This facilitates the work, because it reduces the number of steps in the code. It also gives a possibly higher accuracy, since we don’t have to rely on engineering solutions to find out whether the check box is on the left of the text or on the right.

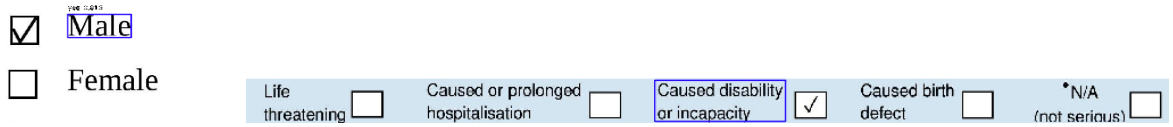



Figure 4. Detection of the text near the checked box using RetinaNet deep learning network.

We have also considered cases where there is an additional information not only near the checkboxes, or the words selected are circled instead (Figure 5). For instance, in the Figure 5, “56 years” is the output of our program.

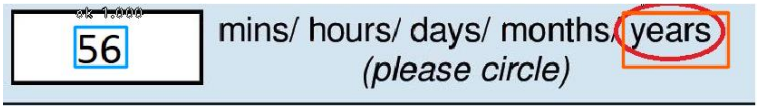



Figure 5. Detection of circled words and additional words.

After coordinate detection of the bounding box of the text, we proceed to the **Step 4**. The image is cropped according to coordinates are image is passed to tesseract OCR. OCR transferred text is then inserted to the corresponding space in the data structure shown in Figure 3.

**Data Preprocessing for RetinaNet Neural Network**

The data for the network was collected from various documents containing different type of checkboxes (Figures 4, 5). Checkmarks inside the boxes also vary, being X-shaped or V-shaped. Some noises were added to the pictures like dots of different colors to avoid overfitting. Negative examples included both – figures with all unchecked boxes and figures without any checkboxes on them.

For positive data a code was written using pytesseract python library to give as an output an hocr file. Hocr is an open standard of data representation for formatted text obtained from, optical character recognition, which among many other things contains the bounding box coordinates of the words. For each figure a bounding box coordinates of the words were found and were given as labels of the data. Three different type of classes were created: one for words near the checkboxes for cases represented in picture 4, and two additional classes for cases presented in Figure 5: one for the text input without checkboxes, and the other for circled words. We named the classes yes, ok and ellipse. You can see an example of preprocessed data in Figure 6.

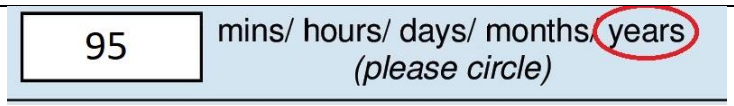
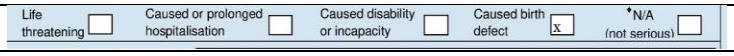
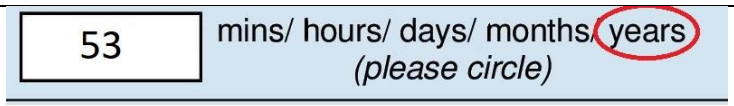
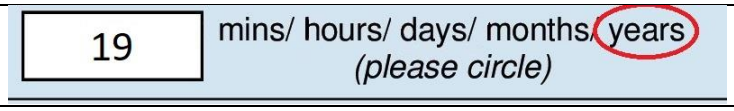
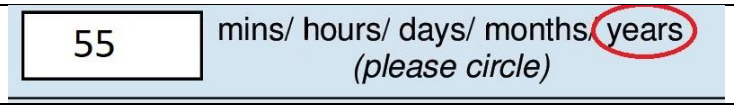
| Checkbox image | $X_{min}$ | $Y_{min}$ | $X_{max}$ | $Y_{max}$ | Class name |
|---|---|---|---|---|---|
|  | 505 | 18 | 670 | 55 | tick |
|  | 80 | 25 | 135 | 65 | notation |
| | 660 | 10 | 77 | 65 | circle |
|  | 330 | 15 | 480 | 50 | tick |
|  | 1340 | 15 | 1580 | 50 | tick |
|  | 1385 | 5 | 1620 | 95 | tick |
|  | 80 | 25 | 135 | 65 | notation |
| | 660 | 10 | 770 | 65 | circle |
|  | 700 | 10 | 890 | 95 | tick |
|  | 85 | 25 | 140 | 65 | notation |
| | 660 | 10 | 770 | 65 | circle |
|  | 70 | 25 | 125 | 65 | notation |
| | 660 | 10 | 770 | 65 | circle |
|  | 840 | 25 | 1190 | 70 | tick |
|  | 1100 | 20 | 1225 | 55 | tick |

Table 6. Preprocessed data for RetinaNet deep learning neural network.

Each of the data lines in Table 6 contains of path to image where it is located, x,y coordinates of upper left and lower right corners of the bounding boxes of the label data (cells 2 through 5), and their respective classes (cell 6).

The data was checked for avoiding possible mistakes and making it work for RetinaNet[1] through keras-retinanet/keras_retinanet/bin/debug.py script designed for debugging images by RetinaNet model developers. To make text detectable by the neural network anchor parameters of anchor boxes were set to ratios = np.array([0.01,0.03,0.09,0.27, 1], keras.backend.floatx()), scales = np.array([$2^0$, $2^{(1.0 / 3.0)}$, $2^{(2.0 / 3.0)}$, 3.5] in keras-retinanet/keras_retinanet/utils/anchors.py[1] python script. All the coordinates were verified through debug.py

**RetinaNet Training**

One of peculiarities of RetinaNet[2] model is the usage of Focal Loss ($FL(p_t) = -(1 - p_t)^{\gamma} * \log(p_t)$) function during the classification. This function increases the overall contribution of positive examples in the training process. The Focal Loss is designed to address the one-stage object

[1] https://github.com/fizyr/keras-retinanet
[2] https://arxiv.org/pdf/1708.02002.pdf

detection scenario in which there is an extreme imbalance between foreground and background classes during training (e.g., 1:1000).

Training data contained 1205 samples. The parameters of training are as follows:

batch_size= 1, steps= 1249, epochs= 50. We used pretrained COCO weights for RetinaNet[1].

We used random-transform and no-resize parameters while training1. Random transform usage allows not to use min_rotation, max_rotation, min_translation, max_translation, min_shear, max_shear, min_scaling, max_scaling, flip_x_chance, flip_y_chance transformation parameters on initial pictures. No-resize is designed to feed the network with pictures of original sizes, and not set one size for all of them

## Conclusion

RetinaNet model was used for detecting/predicting checkboxes and their corresponding values. RetinaNet outputs the probabilities of checkboxes (i.e. the probability of existence of a checkbox), and we consider boxes as checked if the probabilities are higher than 0.5. It has been shown that our model detects checkboxes with high accuracy, approximately 90% on a given test data.

05 March 2020