ASIAN BANKING SCHOOL


Let's Talk Digital Series #15

# An Introduction to Optical Recognition Character (ORC)

# AN INTRODUCTION TO *OPTICAL RECOGNITION CHARACTER*

Some of our work involves manually reviewing and processing physical pages of documents, ranging from invoices to packing lists and insurance certificates, or to retrieve information from documents. It is a tedious process which can be automated by converting an image or a PDF (Portable Document Format) file into either a table or text. In this article, I will be introducing a few ways to retrieve text from PDFs and images.

For the first problem, I'll look at a summary of survey responses conducted by a news agency based in the US. Most PDF reports found online tend to be machine readable, so you can highlight parts of the report which contain the information you are after. It is usually easier to extract information from these documents as the Python code needed to perform this function is easy to write.

**Table 12. Nonmember Survey Response Rates by Region and Specialty, 2018**

| Specialty | Midwest (%) | Northeast (%) | South (%) | West (%) | Total (%) |
|---|---|---|---|---|---|
| Cancer | 30.2 | 39.4 | 40.3 | 29.7 | 35.4 |
| Diabetes & Endocrinology | 34.5 | 20.5 | 28.3 | 13.9 | 24.4 |
| Ear, Nose & Throat | 16.7 | 33.3 | 29.0 | 39.0 | 29.9 |
| Gastroenterology & GI Surgery | 24.3 | 37.8 | 24.6 | 12.8 | 24.7 |
| Geriatrics | 15.2 | 15.6 | 13.3 | 13.8 | 14.5 |
| Gynecology | 25.8 | 21.2 | 28.3 | 13.5 | 23.0 |
| Heart & Heart Surgery | 36.1 | 43.6 | 18.8 | 32.4 | 30.6 |
| Nephrology | 25.7 | 21.2 | 17.9 | 17.1 | 19.9 |
| Neurology & Neurosurgery | 20.6 | 28.1 | 17.5 | 25.7 | 22.2 |
| Ophthalmology | 28.1 | 23.7 | 21.7 | 24.4 | 24.0 |
| Orthopedics | 25.7 | 25.0 | 19.7 | 21.3 | 22.2 |
| Psychiatry | 29.2 | 15.4 | 17.1 | 16.7 | 18.6 |
| Pulmonology | 13.8 | 16.2 | 32.7 | 31.6 | 25.2 |
| Rehabilitation | 21.2 | 28.2 | 32.7 | 12.5 | 24.2 |
| Rheumatology | 13.3 | 18.9 | *26.9* | 7.7 | 17.7 |
| Urology | 23.5 | 38.7 | *25.4* | 10.5 | 24.1 |
| **Overall Response Rate** | **24.2%** | **26.3%** | **24.8%** | **20.3%** | **24.0%** |

https://www.usnews.com/static/documents/health/best-hospitals/BH_Methodology_2018-19.pdf

The image above shows an excerpt of the survey response rates by a hospital's departments. There are a few Python libraries available which can read text from PDF documents like PyPDF2, Textract, Apache Tika, pdfPlumber, and pdfMiner3. To read tables from PDF documents, I usually resort to a library called Tabula .

The raw result from Tabula is as follows:

```
[2]  1 import tabula
     2
     3 # Read pdf into list of DataFrame
     4 sample_list = tabula.read_pdf("BH_Methodology_2018-19.pdf", pages='45')
     5
     6 sample_list
```

```
Got stderr: Dec 09, 2020 3:53:42 AM org.apache.pdfbox.pdmodel.font.FileSystemFontProvider loadDiskCache
WARNING: New fonts found, font cache will be re-built
Dec 09, 2020 3:53:42 AM org.apache.pdfbox.pdmodel.font.FileSystemFontProvider <init>
WARNING: Building on-disk font cache, this may take a while
Dec 09, 2020 3:53:42 AM org.apache.pdfbox.pdmodel.font.FileSystemFontProvider <init>
WARNING: Finished building on-disk font cache, found 17 fonts
Dec 09, 2020 3:53:42 AM org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
WARNING: Using fallback font 'LiberationSans' for 'TimesNewRomanPSMT'
Dec 09, 2020 3:53:44 AM org.apache.pdfbox.pdmodel.font.PDTrueTypeFont <init>
WARNING: Using fallback font 'LiberationSans' for 'TimesNewRomanPSMT'

[                      Unnamed: 0 Unnamed: 1 Midwest  ...  South   West  Total
 0                            NaN  Specialty     (%)  ...    (%)    (%)    (%)
 1                         Cancer        NaN    30.2  ...   40.3   29.7   35.4
 2         Diabetes & Endocrinology        NaN    34.5  ...   28.3   13.9   24.4
 3               Ear, Nose & Throat        NaN    16.7  ...   29.0   39.0   29.9
 4    Gastroenterology & GI Surgery        NaN    24.3  ...   24.6   12.8   24.7
 5                       Geriatrics        NaN    15.2  ...   13.3   13.8   14.5
 6                       Gynecology        NaN    25.8  ...   28.3   13.5   23.0
 7            Heart & Heart Surgery        NaN    36.1  ...   18.8   32.4   30.6
 8                       Nephrology        NaN    25.7  ...   17.9   17.1   19.9
 9         Neurology & Neurosurgery        NaN    20.6  ...   17.5   25.7   22.2
 10                   Ophthalmology        NaN    28.1  ...   21.7   24.4   24.0
 11                     Orthopedics        NaN    25.7  ...   19.7   21.3   22.2
 12                     Psychiatry        NaN    29.2  ...   17.1   16.7   18.6
 13                     Pulmonology        NaN    13.8  ...   32.7   31.6   25.2
 14                  Rehabilitation        NaN    21.2  ...   32.7   12.5   24.2
 15                     Rheumatology        NaN    13.3  ...   26.9    7.7   17.7
 16                         Urology        NaN    23.5  ...   25.4   10.5   24.1
 17           Overall Response Rate        NaN   24.2%  ...  24.8%  20.3%  24.0%

 [18 rows x 7 columns]]
```

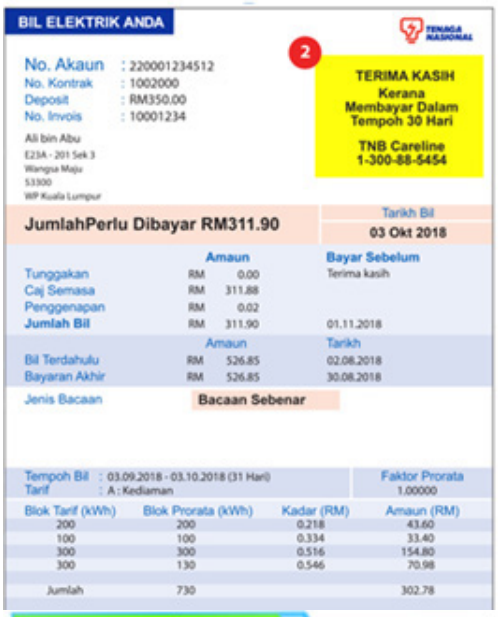
After cleaning the table, a snapshot of the result should look like the following:

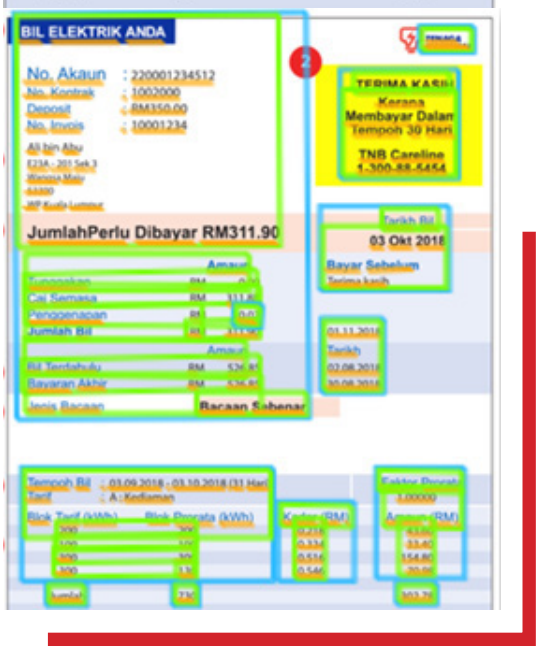
https://pypi.org/project/PyPDF2/
https://textract.readthedocs.io/en/stable/
https://github.com/chrismattmann/tika-python
https://pypi.org/project/pdfplumber/
https://pypi.org/project/pdfminer3/
https://pypi.org/project/tabula-py/

| | Specialty | Midwest (%) | Northeast (%) | South (%) | West (%) | Total |
|---|---|---|---|---|---|---|
| 1 | Cancer | 30.2 | 39.4 | 40.3 | 29.7 | 35.4 |
| 2 | Diabetes & Endocrinology | 34.5 | 20.5 | 28.3 | 13.9 | 24.4 |
| 3 | Ear, Nose & Throat | 16.7 | 33.3 | 29.0 | 39.0 | 29.9 |
| 4 | Gastroenterology & GI Surgery | 24.3 | 37.8 | 24.6 | 12.8 | 24.7 |
| 5 | Geriatrics | 15.2 | 15.6 | 13.3 | 13.8 | 14.5 |
| 6 | Gynecology | 25.8 | 21.2 | 28.3 | 13.5 | 23.0 |
| 7 | Heart & Heart Surgery | 36.1 | 43.6 | 18.8 | 32.4 | 30.6 |
| 8 | Nephrology | 25.7 | 21.2 | 17.9 | 17.1 | 19.9 |
| 9 | Neurology & Neurosurgery | 20.6 | 28.1 | 17.5 | 25.7 | 22.2 |
| 10 | Ophthalmology | 28.1 | 23.7 | 21.7 | 24.4 | 24.0 |
| 11 | Orthopedics | 25.7 | 25.0 | 19.7 | 21.3 | 22.2 |
| 12 | Psychiatry | 29.2 | 15.4 | 17.1 | 16.7 | 18.6 |
| 13 | Pulmonology | 13.8 | 16.2 | 32.7 | 31.6 | 25.2 |
| 14 | Rehabilitation | 21.2 | 28.2 | 32.7 | 12.5 | 24.2 |
| 15 | Rheumatology | 13.3 | 18.9 | 26.9 | 7.7 | 17.7 |
| 16 | Urology | 23.5 | 38.7 | 25.4 | 10.5 | 24.1 |
| 17 | Overall Response Rate | 24.2% | 26.3% | 24.8% | 20.3% | 24.0% |

Comparing the cleaned output of Tabula and the table from the original document, , it can be observed that the results are similar. Once the Tabula output has been transformed into a table, it is easy to slice and dice the data as you see fit.

Sometimes, you do get scans of documents like the sample electricity bill to the right. The problem with scans of documents, or images of documents, is that it is slightly more difficult to get the information you would like. The reason is it is not possible to highlight any text on the image. These problems require the use of Optical Character Recognition (OCR). For this demonstration, I will be using Google Vision API's OCR .




OCR algorithms typically work like this:

- The AI isolates the area which have black inkblots against areas which do not. For areas which do have text, the algorithm draws boxes around the text. These boxes are called bounding boxes.
- The OCR algorithm then looks at individual inkblots to make out words or characters, and whether they are organized as a table or free-form text. It converts these texts into computer understandable text contained within the image.

The image with the bounding boxes is shown on the right.

The raw output of Google Vision API's OCR is as follows:

```
...
description: "BIL ELEKTRIK ANDA\nTENAGA\nNASIONAL\nNo. Akaun :
220001234512\n: 1002000\n: RM350.00\nTERIMA KASIH\nNo. Kontrak\
nKerana\nDeposit\nNo. Invois\nMembayar Dalam\nTempoh 30 Hari\n:
10001234\nAli bin Abu\n3\nTNB Careline\n1-300-88-5454\nE23A - 201
Sek 3\nWangsa Maju\n53300\nWP Kuala Lumpur\nTarikh Bil\nJumlahPerlu
Dibayar RM311.90\n03 Okt 2018\nAmaun\nBayar Sebelum\nTunggakan\nCaj
Semasa\nPenggenapan\nRM\n0.00\nTerima kasih\nRM 311.88\n0,02\nRM\
nJumlah Bil\nRM\n311.90\n01.11.2018\nAmaun\n52685\nTarikh\n5\nBil
Terdahulu\nRM\n02.08.2018\nBayaran
```

https://cloud.google.com/vision/docs/ocr

```
Akhir\nRM\n526.85\n30.08.2018\nJenis Bacaan\nBacaan Sebenar\nTempoh
Bil : 03.09.2018 - 03.10.2018 (31 Hari)\nTarif\nFaktor Prorata\
n:A: Kediaman\n1,00000\nBlok Tarif (KWh)\n200\n100\n300\n300\nBlok
Prorata (kWh)\n200\n100\n300\n130\nKadar (RM)\n0.218\nAmaun (RM)\
n43.60\n33.40\n154.80\n70.98\n8\n0.334\n0.516\n0.546\nJumlah\n730\
n302.78\nTidak Kena\nST\nKena\nST\nKeterangan\nJumlah\nKegunaan
kWh\nKegunaan\nkWh\n600\n130\n730\nRM\n231.80\n70.98\n302.78\
nKegunaan Bulan Semasa\nService Tax (66)\nKWTBB (1.6%)\nRM\n231.80\
n70.98\n302.78\n4.26\n4.84\n10\nCaj Semasa\nRM\n311.88\nBacaan
Meter\nNo Meter\n\320\232egunaan\nUnit\nDahulu\nSemasa\n311201234\
n28470\n29200\n730\nkWh\nSubsidi 1.35 sen/kWh diblayai Kerajaan
Persekutuan RM 10.61\nService Tax (ST) 6% bagi penggunaan Domestik
melebihi 600 kWh\nBayaran melalui cek sah setelah penjelasan cek
oleh bank\n*220001234512000100012340000000000031190\nAli bin Abu\
nE2SA - 201 Sek 3\nWangsa Maju\n53300\nWP Kuala Lumpur\nRM\n311.90\
nAras 17, Wisma TNB, No. 19. Jalan Timur, 46200 Petaling Jaya,
Selangor.\nTENAGA NASIONAL BERHAD (200006), Nombor Datar ST W10-
1608-31022372\n"
bounding_poly {
...
```


The output can be converted into a table after some cleaning:

| | Blok Tarif (KWh) | Blok Prorata (kWh) | Kadar (RM) | Amaun (RM) |
|---|---|---|---|---|
| 1 | 200 | 200 | 0.218 | 43.60 |
| 2 | 100 | 100 | 0.334 | 33.40 |
| 3 | 300 | 300 | 0.516 | 154.80 |
| 4 | 300 | 130 | 0.546 | 70.98 |

After scanning or downloading files which contain the information you are looking for, this OCR algorithm can be applied to extract information from those files and feed this information to your systems by APIs. OCR works well if the document you are extracting information comes from files with a standard structure like ICs or any official forms, and the document should preferably be typed. For handwritten documents, the results of Google Vision API's OCR are likely to vary depending on the quality of the handwriting.

SCAN ME

The full code for both demonstrations above can be accessed via the QR code on the right, or via the link here.

https://cloud.google.com/vision/docs/handwriting
https://github.com/atlas-github/abs_digital/blob/master/Extracting_text_from_images_PDFs.ipynb

ASIAN BANKING SCHOOL

This article is part of the Digital Banking Learning Series, 'Let's Talk Digital', an initiative by the ABS Center for Digital Banking. It is written by industry practitioners and are aimed at educating the general public on the intricacies of digital applications in banking and other related industries, including the latest insights and trends of Digital Banking.

As the industry's preferred partner in learning and development, ABS offers relevant training programmes that covers a comprehensive list of banking areas that are designed and developed in-house by our Specialist Training Consultancy Team or in collaboration with strategic learning partners that includes some of the top business schools in the world. It also provides specialised consulting services and tailored learning solutions to meet the specific needs of its clients.

For more information, visit our website at **www.asianbankingschool.com** or email us at **digitalbanking@asianbankingschool.com**