PDF Miner

서울보증 수재보험료 같은 경우에 PDF내용이 수동으로 작성한 내용이 아니고, 컴퓨터로 생성한 문서이므로 OCR할 필요없이 PDF의 텍스트를 추출하면 되고, 텍스트의 좌표정보를 구할 수 있으면 됩니다. 인터넷에서 찾아보니, Python으로 PDF를 처리하는 라이브러리는 대표적으로 다음 몇가지가 있습니다.

* PyPDF2
* ReportLab
* Pdfminer.six
* 기타

각 라이이브러리가 각자의 특징이나 장점이 있지만 현재 저희가 원하는 PDF 기능이 텍스트 추출 와 텍스트 좌표 정보 추출이므로, 이중에서 pdfminer.six라이브러리가 제일 적합합니다. 그래서 해당 라이브러리에서 다음과 같이 PDF추출 작업을 진행해봤습니다.

# 개발 준비

Python 버전: 3.11.2

설치된 라이브러리:

pdfminer.six, OpenCV

pdfminer.six는 실제로 PDF의 텍스트 내용 및 텍스트 좌표를 추출하는 라이브러리이고, OpenCV라이브러리는 추출된 정보를 간편하게 확인할수 있도록 추출된 내용을 그림으로 표현하는 도구입니다.

* pip list

Package Version

------------------ --------

cffi 1.15.1

charset-normalizer 3.1.0

cryptography 40.0.2

numpy 1.24.3

opencv-python 4.7.0.72

pdfminer.six 20221105

pip 23.1.2

pycparser 2.21

setuptools 65.5.0

# 소스코드

from pdfminer.pdfparser import PDFParser

from pdfminer.pdfdocument import PDFDocument

from pdfminer.pdfpage import PDFPage

from pdfminer.pdfpage import PDFTextExtractionNotAllowed

from pdfminer.pdfinterp import PDFResourceManager

from pdfminer.pdfinterp import PDFPageInterpreter

from pdfminer.layout import LAParams

from pdfminer.converter import PDFPageAggregator

import pdfminer

import numpy as np

import cv2

import json

import os

import argparse

parser = argparse.ArgumentParser(description='Parse PDF files', formatter\_class=argparse.ArgumentDefaultsHelpFormatter)

parser.add\_argument('-f',

                    '--filename',

                    help='pdf filename, without extension',

                    metavar='str',

                    type=str,

                    default='one')

args = parser.parse\_args()

filename = args.filename

pdf\_path = f'data/pdf\_sample/{filename}.pdf' # pdf 파일 경로 지정

try:

    fp = open(pdf\_path, 'rb') # pdf 파일 열기

except FileNotFoundError:

    print('File not found...')

    exit()

parser = PDFParser(fp) # pdf 파일과 관련된 parser생성

doc = PDFDocument(parser) # pdf 문서 구조를 저장하는 PDFDocument 겍체 생성

# pdf 내용 추출 가능 여부 판단

if not doc.is\_extractable:

    raise PDFTextExtractionNotAllowed

rsrcmgr = PDFResourceManager() # 공유 자원을 저장하는 PDFResourceManager객체 생성

laparams = LAParams() # Layout 파라미터 객체생성

device = PDFPageAggregator(rsrcmgr, laparams=laparams) # PDFPageAggregator객체 생성

interpreter = PDFPageInterpreter(rsrcmgr, device) # PDFPageInterpreter객채 생성

pdf\_data = {} # pdf 전체 데이터를 담을 dict

pdf\_data['version'] = '1.0'

pdf\_data['page\_list'] = []

for page in PDFPage.create\_pages(doc): # 페이지 별로 처리

    interpreter.process\_page(page)

    layout = device.get\_result() # layout 결과 가져오기

    print(f'#################### {layout.pageid}')

    page\_data = {} # 해당 페이지의 데이터

    page\_data['page\_id'] = layout.pageid # 페이지 번호

    page\_data['page\_width'] = layout.bbox[2] - layout.bbox[0]  # 페이지 너비

    page\_data['page\_height'] = layout.bbox[3] - layout.bbox[1]  # 페이지 높이

    page\_data['text\_list'] = [] # 텍스트 정보 리스트

    for obj in layout.\_objs:

        if isinstance(obj, pdfminer.layout.LTTextBoxHorizontal):

            data\_item = {}

            data\_item['bbox'] = {}

            data\_item['bbox']['x1'] = obj.bbox[0]

            data\_item['bbox']['y1'] = layout.bbox[3] - obj.bbox[3]

            data\_item['bbox']['x2'] = obj.bbox[2]

            data\_item['bbox']['y2'] = layout.bbox[3] - obj.bbox[1]

            data\_item['text'] = obj.get\_text()

            page\_data['text\_list'].append(data\_item)

    pdf\_data['page\_list'].append(page\_data)

json\_data = json.dumps(pdf\_data)

print(json\_data)

json\_path = f'data/json\_result/{filename}.json'

with open(json\_path, 'w', encoding='UTF8') as json\_file:

    json\_file.write(json\_data)

colors = {'blue': (255, 0, 0), 'green': (0, 255, 0), 'red': (0, 0, 255), 'yellow': (0, 255, 255),

          'magenta': (255, 0, 255), 'cyan': (255, 255, 0), 'white': (255, 255, 255), 'black': (0, 0, 0),

          'gray': (125, 125, 125), 'rand': np.random.randint(0, high=256, size=(3,)).tolist(),

          'dark\_gray': (50, 50, 50), 'light\_gray': (220, 220, 220)}

for index, page in enumerate(pdf\_data['page\_list']):

    page\_width = page['page\_width']

    page\_height = page['page\_height']

    image = np.zeros((int(page\_height), int(page\_width), 3), dtype='uint8')

    image[:] = colors['light\_gray']

    for text\_item in page['text\_list']:

        # bbox 출력

        cv2.rectangle(image, (int(text\_item['bbox']['x1']), int(text\_item['bbox']['y1'])), (int(

            text\_item['bbox']['x2']), int(text\_item['bbox']['y2'])), colors['red'], 1)

        # 텍스트 출력

        if text\_item.get('text') != None:

            for i, line in enumerate(text\_item.get('text').split('\n')): # OpenCV의 putText메소드에서 '\n'줄바꿈 처리가 잘 안돼서 수동으로 줄 단위로 잘라서 출력

                cv2.putText(image, line, (int(text\_item['bbox']['x1']), int(text\_item['bbox']['y1'] + 10 \* (i + 1))), cv2.FONT\_HERSHEY\_COMPLEX\_SMALL, 0.4, colors['black'], 1, cv2.LINE\_AA)

    page\_title = f"page {page['page\_id']}"

    print(f"data/pdf\_images/{filename}/page{page['page\_id']}.png")

    pdf\_images\_path = f"data/pdf\_images/{filename}"

    os.makedirs(pdf\_images\_path, exist\_ok=True)

    cv2.imwrite(f"{pdf\_images\_path}/page{page['page\_id']}.png", image)

위 소스는 pdfminer.six라이브러리를 이용해서 PDF파일의 전체 내용을 페이지별로 페이지 사이즈, 텍스트 내용, 텍스트 좌표 등 정보를 가져오는 소스입니다. 샘플 파일은 서울보증에서 제공한 수재보흠 문서입니다.

|  |  |  |
| --- | --- | --- |
| 파일명 | 수재보험료 종류 | 페이지 수 |
| data/pdf\_sample/one.pdf | Willis | 1 |
| data/pdf\_sample/two.pdf | Aon | 2 |
| data/pdf\_sample/three.pdf | Aon | 2 |
| data/pdf\_sample/four.pdf | Willis Hong Kong Limited | 71 |
| data/pdf\_sample/five.pdf | GuyCarpenter | 80 |

추출된 텍스트 와 좌표 정보의 json은 data/json\_result폴더에 저장되고, opencv로 그린 이미지는 pdf\_images폴더에 저장됩니다.

# 소스코드 실행 절차

1. github에서 소스 다운로드: git clone <https://github.com/ljyGmail/pdfminer_demo.git>
2. pdfminer\_demo경로에서 python가상 환경 만들기: python -m venv .venv



1. 가상환경 활성화: .\.venv\Scripts\activate



1. 사용된 라이브러리 설치 pip install -r .\requirements.txt



1. 프로그램 실행: python main.py -f {파일명(one/two/three/four/five)}



# CPU 사용

다음은 four.pdf를 실행할 때 나타나는 CPU사용율입니다.

A screenshot of a computer

Description automatically generated with medium confidence

# Pdfminer.six라이브러리 관련 링크

<https://pdfminersix.readthedocs.io/en/latest/tutorial/highlevel.html>

공식 사이트