**Code for Text Extraction using NER**

Copy Paste All the Code and execute one by one as the following steps.

Make sure you perform this on Open CV. Otherwise it will give some errors in normal compiler. (Prefer Google Collab).

1. For Internshala

def internshala():

    """Code for Name"""

    name=""

    for i in ocr\_result:

      name+=i

      if i=="\n":

        break

    name = name.strip()

    """Code for Date"""

    date\_ = ""

    for i in ocr\_result:

      date\_+=i

    words = date\_.split()

    for word in words:

        if len(word) == 10 and word[4] == "-" and word[7] == "-":

            # print(f"Date of Certification: {word}")

            # new\_list2 = list(word)

            break

    """Code for Week"""

    res = []

    for i in str(words):

      if i.isdigit():

          res.append(i)

    """Code for Course Name"""

    string = str(words)

    start = "training', 'on', "

    end = ", 'The"

    start\_index = string.index(start) + len(start)

    end\_index = string.index(end)

    cropped\_string = string[start\_index:end\_index].strip()

    course = ""

    for i in cropped\_string:

      course+=i

    special\_char = ["'",","]

    x= course.replace("'",'')

    normal\_string = x

    for i in special\_char:

      normal\_string = normal\_string.replace(i,"")

    """Attributes"""

    fname = list(name)

    certificate = list(word)

    week = f"{res[0]}-weeks"

    course = normal\_string

    l1=['Name','Date of Certification','Course Name']

    l2=[name, word, course]

    d1=zip(l1,l2)

    d2 = dict(d1)

    file\_path = 'certificate.xlsx'

    try:

        df\_existing = pd.read\_excel(file\_path)

    except FileNotFoundError:

        data\_dict = d2

        df\_new = pd.DataFrame(data\_dict, index=[0])

        df\_new.to\_excel(file\_path, index=False)

    else:

        data\_dict = d2

        df\_new = pd.DataFrame(data\_dict, index=[0])

        df\_updated = pd.concat([df\_existing, df\_new], ignore\_index=True)

        df\_updated.to\_excel(file\_path, index=False)

1. For TechEdu

def techedu():

  name=""

  string = ocr\_result

  start = "TO\n"

  end = "\nFOR"

  start\_index = string.index(start) + len(start)

  end\_index = string.index(end)

  cropped\_string = string[start\_index:end\_index].strip()

  name=""

  for i in cropped\_string:

    name+=i

  date\_ = ""

  for i in ocr\_result:

    date\_+=i

    if i=="\n":

      break

  new\_date = str(date\_.split())

  special\_char = ["'",",","[","]"]

  x= new\_date.replace("'",'')

  normal\_string = x

  for i in special\_char:

    normal\_string = normal\_string.replace(i,"")

  string = ocr\_result

  start = "\nIN"

  end = " BY"

  start\_index = string.index(start) + len(start)

  end\_index = string.index(end)

  cropped\_string = string[start\_index:end\_index].strip()

  course=""

  for i in cropped\_string:

    course+=i

  fname = list(name)

  certificate = list(course)

  date = normal\_string

  l1=['Name','Date of Certification','Course Name']

  l2=[name, date, course]

  d1=zip(l1,l2)

  d2 = dict(d1)

  import pandas as pd

  file\_path = 'certificate.xlsx'

  try:

      df\_existing = pd.read\_excel(file\_path)

  except FileNotFoundError:

      data\_dict = d2

      df\_new = pd.DataFrame(data\_dict, index=[0])

      df\_new.to\_excel(file\_path, index=False)

  else:

      data\_dict = d2

      df\_new = pd.DataFrame(data\_dict, index=[0])

      df\_updated = pd.concat([df\_existing, df\_new], ignore\_index=True)

      df\_updated.to\_excel(file\_path, index=False)

1. For Coursera

def coursera():

  name=""

  string = ocr\_result

  start = "2022\n"

  end = "\nhas"

  start\_index = string.index(start) + len(start)

  end\_index = string.index(end)

  cropped\_string = string[start\_index:end\_index].strip()

  name=""

  for i in cropped\_string:

    name+=i

  date\_ = ""

  for i in ocr\_result:

    date\_+=i

    if i=="\n":

      break

  new\_date = str(date\_.split())

  special\_char = ["'",",","[","]"]

  x= new\_date.replace("'",'')

  normal\_string = x

  for i in special\_char:

    normal\_string = normal\_string.replace(i,"")

  string = ocr\_result

  start = "completed\n"

  end = "\nan"

  start\_index = string.index(start) + len(start)

  end\_index = string.index(end)

  cropped\_string = string[start\_index:end\_index].strip()

  course=""

  for i in cropped\_string:

    course+=i

  fname = list(name)

  certificate = list(course)

  l1=['Name','Date of Certification','Course Name']

  l2=[name, normal\_string, course]

  d1=zip(l1,l2)

  d2 = dict(d1)

  import pandas as pd

  file\_path = 'certificate.xlsx'

  try:

      df\_existing = pd.read\_excel(file\_path)

  except FileNotFoundError:

      data\_dict = d2

      df\_new = pd.DataFrame(data\_dict, index=[0])

      df\_new.to\_excel(file\_path, index=False)

  else:

      data\_dict = d2

      df\_new = pd.DataFrame(data\_dict, index=[0])

      df\_updated = pd.concat([df\_existing, df\_new], ignore\_index=True)

      df\_updated.to\_excel(file\_path, index=False)

1. Install pdfplumber

# It is used to extract the text from the pdf.

!pip install pdfplumber

5. Import all the libraries of the pdfplumber

import pdfplumber

with pdfplumber.open('19.pdf') as pdf:

            for page in pdf.pages:

              ocr\_result = page.extract\_text()

6. Import nltk libraries

# nltk is an NLP tool use to extract the words from the given pdf

import nltk

nltk.download('popular')

#to tokenize words

from nltk.tokenize import word\_tokenize

print(word\_tokenize(ocr\_result))

word\_tokenized = word\_tokenize(ocr\_result)

7.Tokenize all the words which we extracted

#to tokenize sentences

from nltk.tokenize import sent\_tokenize

print(sent\_tokenize(ocr\_result))

import re

import pandas as pd

8.Final result will be saved in the excel sheet

# The below code will distinguish and print the output of required certificate

# The Date format is different for the certificates so we have to use different format

# We have used If and else technique for different certificates

from datetime import datetime

from dateutil import parser

txt = ocr\_result

name1='all the best for future endeavours'

name2='IS PROUDLY PRESENTED'

name3='online non'

x = re.search(f"{name1}",txt)

y = re.search(f"{name2}",txt)

z = re.search(f"{name3}",txt)

if x:

  print("Internshala")

  internshala()

if y:

  print("Techedu")

  techedu()

if z:

  print("coursera")

  coursera()