```
!pip install python-levenshtein
```

```
Collecting python-levenshtein
  Downloading https://files.pythonhosted.org/packages/42/a9/d1785c85ebf9b7dfacd08938d
     |████████████████████████████████| 51kB 1.6MB/s
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-packages (
Building wheels for collected packages: python-levenshtein
  Building wheel for python-levenshtein (setup.py) ... done
  Created wheel for python-levenshtein: filename=python_Levenshtein-0.12.0-cp36-cp36m
  Stored in directory: /root/.cache/pip/wheels/de/c2/93/660fd5f7559049268ad2dc6d81c4e
Successfully built python-levenshtein
Installing collected packages: python-levenshtein
Successfully installed python-levenshtein-0.12.0
```

```
import os
import re
import Levenshtein
```

```
'''OCR text taken of PAN card document, Here any kind of text image can be used and any ki
For tutorial sake,let's assume this OCR text is perfect/without any error '''
good_ocr_text = '''आयकर विभाग
भारत सरकार
GOVT OF INDIA
INCOME TAX DEPARTMENT
स्थायी लेखा संख्या काई
Permanent Account Number Card
ABCXY1234Z
Name
ABCD XYZ
father's Name
PQRST UVW
Applcart Sigrahre
Date of Birth
01/01/1975
uSianature'''

'''
Methods to extract certain entities from OCR text using regular expressions
'''
def getPanName(text):
    pattern='(?:Name)\s*([a-z]*\s*[a-z]*\s*[a-z]*)$'
    value = re.search(pattern, text, re.I|re.MULTILINE)
    if value:
        return value.group(1)
def getPanNumber(text):

    pattern= '(?:Number Card)\s*([a-z]{5}\d{4}[a-z]{1})$'
    value = re.search(pattern, text, re.I|re.MULTILINE)
    if value:
        return value.group(1)
```

```
print('Name:',getPanName(good_ocr_text))
print('PAN No:',getPanNumber(good_ocr_text))
```

```
Name: ABCD XYZ
PAN No: ABCXY1234Z
```

```
'''
Now,let's assume due to some reason(image quality changed,OCR engine changed etc.) OCR tex
'''
bad_ocr_text = '''आयकर विभाग
भारत सरकार
GOVT OF INDIA
INCOME TAX DEPARTMENT
स्थायी लेखा संख्या काई
Permanent Account Numper Card
ABCXY1234Z
Nane
ABCD XYZ
father's Nane
PQRST UVW
Applcart Sigrahre
Date of Birth
01/01/1975
uSianature'''


'''
Running same regular expressions will not give desired results
'''
print('Name:',getPanName(bad_ocr_text))
print('PAN No:',getPanNumber(bad_ocr_text))
```

```
Name: None
PAN No: None
```

```
'''
Create list of words used in regular expressions
'''
good_words = ['Name','Number','Card']


'''
Method to correct bad ocr words:
If bad OCR word matches(75% match) with any of good word then that bad OCR word should be
'''
def ocr_corrector(text,good_words):
  corrected_text = ''
  for sent in text.split("\n"):
    #print(sent)
    new_sent = []
    for word in sent.split(" "):
      if not word.lower() in good_words:
        for gword in good_words:
          if Levenshtein.ratio(word,gword) >= 0.75:
            word = gword.strip()
```

```
        new_sent.append(word)
      corrected_text += (" ".join(new_sent)+"\n")
    return corrected_text


bad_ocr_text_corrected = ocr_corrector(bad_ocr_text,good_words)


'''
Bad OCR words are corrected.
Numper --> Number
Nane --> Name
'''
print(bad_ocr_text_corrected)
```

```
आयकर विभाग
भारत सरकार
GOVT OF INDIA
INCOME TAX DEPARTMENT
स्थायी लेखा संख्या कार्ड
Permanent Account Number Card
ABCXY1234Z
Name
ABCD XYZ
father's Name
PQRST UVW
Applcart Sigrahre
Date of Birth
01/01/1975
uSianature



'''
This corrected OCR will give desired results
'''
print('Name:',getPanName(bad_ocr_text_corrected))
print('PAN No:',getPanNumber(bad_ocr_text_corrected))
```

```
Name: ABCD XYZ
PAN No: ABCXY1234Z
```