pii_detect

November 20, 2021

[ ]:
```
import pandas as pd
import numpy as np
from presidio_analyzer import AnalyzerEngine
from presidio_anonymizer import AnonymizerEngine
from presidio_analyzer import PatternRecognizer
from presidio_image_redactor import ImageRedactorEngine
from presidio_image_redactor import ImageAnalyzerEngine
from PIL import Image
from presidio_analyzer import Pattern

from typing import List, Optional, Dict, Union, Iterator, Iterable
import collections
from dataclasses import dataclass
import pprint

from presidio_analyzer import AnalyzerEngine, RecognizerResult
from presidio_anonymizer import AnonymizerEngine
from presidio_anonymizer.entities import EngineResult

import os
import fnmatch
from pandas import read_excel
import docx2txt
from pdf2image import convert_from_path
```

[ ]:
```
@dataclass
class DictAnalyzerResult:
    """Hold the analyzer results per value or list of values."""
    key: str
    value: Union[str, List[str]]
    recognizer_results: Union[List[RecognizerResult],␣
 ↪List[List[RecognizerResult]]]


class BatchAnalyzerEngine(AnalyzerEngine):
    """
```

```
    Class inheriting from AnalyzerEngine and adds the funtionality to analyze␣
↪lists or dictionaries.
    """

    def analyze_list(self, list_of_texts: Iterable[str], **kwargs) ->␣
↪List[List[RecognizerResult]]:
        """
        Analyze an iterable of strings

        :param list_of_texts: An iterable containing strings to be analyzed.
        :param kwargs: Additional parameters for the `AnalyzerEngine.analyze`␣
↪method.
        """

        list_results = []
        for text in list_of_texts:
            results = self.analyze(text=text, **kwargs) if isinstance(text,␣
↪str) else []
            list_results.append(results)
        return list_results

    def analyze_dict(
     self, input_dict: Dict[str, Union[object, Iterable[object]]], **kwargs) ->␣
↪Iterator[DictAnalyzerResult]:
        """
        Analyze a dictionary of keys (strings) and values (either object or␣
↪Iterable[object]).
        Non-string values are returned as is.

                :param input_dict: The input dictionary for analysis
        :param kwargs: Additional keyword arguments for the `AnalyzerEngine.
↪analyze` method
        """

        for key, value in input_dict.items():
            if not value:
                results = []
            else:
                if isinstance(value, str):
                    results: List[RecognizerResult] = self.analyze(text=value,␣
↪**kwargs)
                elif isinstance(value, collections.abc.Iterable):
                    results: List[List[RecognizerResult]] = self.analyze_list(
                                list_of_texts=value,
                                **kwargs)
                else:
```

```
                    results = []
            yield DictAnalyzerResult(key=key, value=value,␣
↪recognizer_results=results)
```

[ ]:
```
location_list = pd.read_csv("us_cities_states_counties.csv", sep = '|').
↪reset_index(drop = True)
```

[ ]:
```
    batch_analyzer = BatchAnalyzerEngine()
    analyzer = AnalyzerEngine()
    image_analyzer = ImageAnalyzerEngine()
    redactor = ImageRedactorEngine()


    # Adding zip code in the entity list. Make sure zip code is turned into a␣
↪string for regex to work
    zip_pattern = Pattern(name="zip_pattern",regex= '(\\b\\d{5}(?:\\-\\d{4})?
↪\\b)', score = 0.5)#regular expression that selects zip codes
    zip_recognizer = PatternRecognizer(supported_entity="ZIPCODE", #name of new␣
↪entity
                                       patterns = [zip_pattern],
                                       context= ["zip","zipcode"])#including any␣
↪surrounding words that has zip or zipcode in it
    batch_analyzer.registry.add_recognizer(zip_recognizer)#adding new zip code␣
↪recognizer to the model
    analyzer.registry.add_recognizer(zip_recognizer)


    #Adding State
    state_recognizer = PatternRecognizer(supported_entity="STATE",#name of new␣
↪entity
                                         deny_list=list(location_list['State␣
↪short'].dropna().unique()),#only include unique states, drop null values
                                         context= ["state","address"])#including␣
↪any surrounding words that has state or address in it
    batch_analyzer.registry.add_recognizer(state_recognizer)#adding new state␣
↪recognizer to the model
    analyzer.registry.add_recognizer(state_recognizer)

    #Adding List of Cities
    city_recognizer = PatternRecognizer(supported_entity="CITY",#name of new␣
↪entity
                                        deny_list=list(location_list['City'].
↪dropna().unique()),#only include unique city, drop null values
                                        context= ["city","address"])#including␣
↪any surrounding words that has city or address in it
```

```
    batch_analyzer.registry.add_recognizer(city_recognizer)#adding new city␣
↪recognizer to the model
    analyzer.registry.add_recognizer(city_recognizer)

    #Adding Password
    password_pattern = Pattern(name="password_pattern",regex= '^(?=.*?[A-Z])(?=.
↪*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*-]).{8,}$', score = 0.5)#regular␣
↪expression that selects password
    password_recognizer = PatternRecognizer(supported_entity="PASSWORD", #name␣
↪of new entity
                        patterns = [password_pattern],
                        context= ["password"])##including any surrounding words␣
↪that has password in it
    batch_analyzer.registry.add_recognizer(password_recognizer) #adding␣
↪password recognizer to the model
    analyzer.registry.add_recognizer(password_recognizer)
```

[ ]:
```
    for filename in os.listdir('.'):
        if fnmatch.fnmatch(filename, 's_pii_*.txt'):

            df = pd.read_csv(filename, index_col = 0).reset_index(drop = True)
            df = df.astype(str)
            #df['zip'] = df['zip'].astype(str)
            #df['phone numbers'] = df['phone numbers'].astype(str)
            df_dict = df.to_dict(orient="list") #df being converted to a␣
↪dictionary
            analyzer_results = batch_analyzer.analyze_dict(df_dict,␣
↪language="en")
            analyzer_df = pd.DataFrame(analyzer_results) #converting into a␣
↪dataframe
            presidio_df = pd.DataFrame(list(analyzer_df['recognizer_results']),␣
↪analyzer_df['key']).reset_index()
            presidio_df.insert(0, 'filename', filename, True)


    for filename in os.listdir('.'):
        if fnmatch.fnmatch(filename, 's_pii_*.csv'):
            #from csv files
            csv = pd.read_csv(filename)
            csv = csv.astype(str).replace('nan',np.nan)
            df_dict_csv = csv.to_dict(orient="list") #df being converted to a␣
↪dictionary
            analyzer_results_csv = batch_analyzer.analyze_dict(df_dict_csv,␣
↪language="en")
            analyzer_df_csv = pd.DataFrame(analyzer_results_csv) #converting␣
↪into a dataframe
```

```
            presidio_df_csv = pd.
↪DataFrame(list(analyzer_df_csv['recognizer_results']),␣
↪analyzer_df_csv['key']).reset_index()
            presidio_df_csv.insert(0, 'filename', filename, True)


    for filename in os.listdir('.'):
        if fnmatch.fnmatch(filename, 's_pii_*.xlsx'):
            xlsx = pd.read_excel(filename, engine = 'openpyxl')
            xlsx = xlsx.astype(str).replace('nan',np.nan)
            df_dict_xlsx = xlsx.to_dict(orient="list") #df being converted to a␣
↪dictionary
            analyzer_results_xlsx = batch_analyzer.analyze_dict(df_dict_xlsx,␣
↪language="en")
            analyzer_df_xlsx = pd.DataFrame(analyzer_results_xlsx) #converting␣
↪into a dataframe
            presidio_df_xlsx = pd.
↪DataFrame(list(analyzer_df_xlsx['recognizer_results']),␣
↪analyzer_df_xlsx['key']).reset_index()
            presidio_df_xlsx.insert(0, 'filename', filename, True)


    for filename in os.listdir('.'):
        if fnmatch.fnmatch(filename, 's_pii_*.docx'):
            MY_TEXT = docx2txt.process(filename)
            with open("pii_docx_made.txt", "w") as text_file:
                print(MY_TEXT, file=text_file)

            docx = pd.read_csv("pii_docx_made.txt", sep = "\t")
            df_dict_docx = docx.to_dict(orient="list") #df being converted to a␣
↪dictionary
            analyzer_results_docx = batch_analyzer.analyze_dict(df_dict_docx,␣
↪language="en")
            analyzer_df_docx = pd.DataFrame(analyzer_results_docx) #converting␣
↪into a dataframe
            presidio_df_docx = pd.
↪DataFrame(list(analyzer_df_docx['recognizer_results']),␣
↪analyzer_df_docx['key']).reset_index()
            presidio_df_docx.insert(0, 'filename', filename, True)



    for filename in os.listdir('.'):
        if fnmatch.fnmatch(filename, 's_pii_*.pdf'):
        # Store Pdf with convert_from_path function
            images = convert_from_path(filename)
```

```
            for i in range(len(images)):  # Save pages as images in the pdf
                images[i].save(str('made_')+ filename + str(i) +'.jpg', 'JPEG')


    for filename in os.listdir('.'):
        if fnmatch.fnmatch(filename, 'made_s_pii_pdf*.jpg'):
            img_pdf = Image.open(filename)
            img_pdf = img_pdf.convert('RGBA')

            result_pdf = image_analyzer.analyze(image=img_pdf)
            #image_analyzer = ImagePiiVerifyEngine() #to see the result of the␣
↪analysis
            #result_pdf = image_analyzer.verify(image=img_pdf) #to see the␣
↪result of the analysis
            presidio_df_pdf = pd.DataFrame(result_pdf)
            presidio_df_pdf.insert(0, 'filename', filename, True)
            redacted_image_pdf = redactor.redact(image=img_pdf)#saving␣
↪redeacted image on the folder
            # save the redacted image
            redacted_image_pdf.save(str('redacted') + filename + ".png")


    for filename in os.listdir('.'):
        if fnmatch.fnmatch(filename, 's_pii_*.jpg'):
            img_jpg = Image.open(filename)

            result_jpg = image_analyzer.analyze(image=img_jpg)
            #image_analyzer = ImagePiiVerifyEngine() #to see the result of the␣
↪analysis
            #result_pdf = image_analyzer.verify(image=img_pdf) #to see the␣
↪result of the analysis
            presidio_df_jpg = pd.DataFrame(result_jpg)
            presidio_df_jpg.insert(0, 'filename', filename, True)
            redacted_image_jpg = redactor.redact(image=img_jpg)#saving␣
↪redeacted image on the folder
            # save the redacted image
            redacted_image_jpg.save(str('redacted') + filename + ".png")


    for filename in os.listdir('.'):
        if fnmatch.fnmatch(filename, 's_pii_*.png'):
            img_png = Image.open(filename)

            result_png = image_analyzer.analyze(image=img_png)
```

```
            #image_analyzer = ImagePiiVerifyEngine() #to see the result of the␣
↪analysis
            #result_pdf = image_analyzer.verify(image=img_pdf) #to see the␣
↪result of the analysis
            presidio_df_png = pd.DataFrame(result_png)
            presidio_df_png.insert(0, 'filename', filename, True)
            redacted_image_png = redactor.redact(image=img_png)#saving␣
↪redeacted image on the folder
            # save the redacted image
            redacted_image_jpg.save(str('redacted') + filename + ".png")
```

[ ]:
```
    frames = [presidio_df, presidio_df_csv, presidio_df_xlsx, presidio_df_docx,
            presidio_df_pdf, presidio_df_jpg, presidio_df_png]
    final = pd.concat(frames)
    final.to_csv("result_structured.csv")
```

[ ]:
```
final = pd.read_csv('result_structured.csv')
final
```

[ ]:
```
     Unnamed: 0       filename         key  \
0             0  s_pii_txt.txt          ID
1             1  s_pii_txt.txt  CREATED_BY
2             2  s_pii_txt.txt  CREATED_ON
3             3  s_pii_txt.txt  UPDATED_BY
4             4  s_pii_txt.txt  UPDATED_ON
..          …             …         …
193          21  s_pii_png.png         NaN
194          22  s_pii_png.png         NaN
195          23  s_pii_png.png         NaN
196          24  s_pii_png.png         NaN
197          25  s_pii_png.png         NaN

                                                     0  \
0                                                   []
1                                                   []
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
3                                                   []
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
..                                                  …
193    type: US_SSN, start: 965, end: 974, score: 0.05
194  type: US_BANK_NUMBER, start: 965, end: 974, sc…
195  type: US_DRIVER_LICENSE, start: 965, end: 974,…
196    type: AU_TFN, start: 965, end: 974, score: 0.01
197  type: LOCATION, start: 1144, end: 1148, score:…

                                                     1  \
0                                                   []
```

```
1                                                    []
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
3                                                    []
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
..                                                   …
193                                                 NaN
194                                                 NaN
195                                                 NaN
196                                                 NaN
197                                                 NaN

                                                     2  \
0                                                    []
1                                                    []
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
3                                                    []
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
..                                                   …
193                                                 NaN
194                                                 NaN
195                                                 NaN
196                                                 NaN
197                                                 NaN

                                                     3  \
0                                                    []
1                                                    []
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
3                                                    []
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
..                                                   …
193                                                 NaN
194                                                 NaN
195                                                 NaN
196                                                 NaN
197                                                 NaN

                                                     4  \
0                                                    []
1                                                    []
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
3                                                    []
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
..                                                   …
193                                                 NaN
194                                                 NaN
195                                                 NaN
```

```
196                                                  NaN
197                                                  NaN

                                                       5  \
0                                                      []
1                                                      []
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
3                                                      []
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
..                                                   …
193                                                  NaN
194                                                  NaN
195                                                  NaN
196                                                  NaN
197                                                  NaN

                                                       6  …  \
0                                                      []  …
1                                                      []  …
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]  …
3                                                      []  …
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]  …
..                                                   …  …
193                                                  NaN  …
194                                                  NaN  …
195                                                  NaN  …
196                                                  NaN  …
197                                                  NaN  …

                                                      90  \
0                                                      []
1                                                      []
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
3                                                      []
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
..                                                   …
193                                                  NaN
194                                                  NaN
195                                                  NaN
196                                                  NaN
197                                                  NaN

                                                      91  \
0                                                      []
1                                                      []
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
3                                                      []
```

```
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
..                                                   …
193                                                 NaN
194                                                 NaN
195                                                 NaN
196                                                 NaN
197                                                 NaN

                                                     92  \
0                                                    []
1                                                    []
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
3                                                    []
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
..                                                   …
193                                                 NaN
194                                                 NaN
195                                                 NaN
196                                                 NaN
197                                                 NaN

                                                     93  \
0                                                    []
1                                                    []
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
3                                                    []
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
..                                                   …
193                                                 NaN
194                                                 NaN
195                                                 NaN
196                                                 NaN
197                                                 NaN

                                                     94  \
0                                                    []
1                                                    []
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
3                                                    []
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
..                                                   …
193                                                 NaN
194                                                 NaN
195                                                 NaN
196                                                 NaN
197                                                 NaN
```

```
                                                    95  \
0                                                    []
1                                                    []
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
3                                                    []
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
..                                                   …
193                                                 NaN
194                                                 NaN
195                                                 NaN
196                                                 NaN
197                                                 NaN

                                                    96  \
0                                                    []
1                                                    []
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
3                                                    []
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
..                                                   …
193                                                 NaN
194                                                 NaN
195                                                 NaN
196                                                 NaN
197                                                 NaN

                                                    97  \
0                                                    []
1                                                    []
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
3                                                    []
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]
..                                                   …
193                                                 NaN
194                                                 NaN
195                                                 NaN
196                                                 NaN
197                                                 NaN

                                                    98   99
0                                                    []  NaN
1                                                    []  NaN
2    [type: DATE_TIME, start: 0, end: 16, score: 0.85]  NaN
3                                                    []  NaN
4    [type: DATE_TIME, start: 0, end: 16, score: 0.85]  NaN
..                                                   …  …
193                                                 NaN  NaN
```

```
194                                                    NaN  NaN
195                                                    NaN  NaN
196                                                    NaN  NaN
197                                                    NaN  NaN

[198 rows x 103 columns]
```

[ ]:
```
final['filename'].unique()# contains all of the files from where it analyzed␣
↪the data
```

[ ]:
```
array(['s_pii_txt.txt', 's_pii_csv.csv', 's_pii_excel.xlsx',
       's_pii_docx.docx', 'made_s_pii_pdf.pdf0.jpg', 's_pii_jpg.jpg',
       's_pii_png.png'], dtype=object)
```

[ ]: