@Odemir Depieri Jr

Guia rápido

Trabalhando com PDFs em Python

PyPDF2

Minerar/Manipular PDF

PyPDF2

PyPDF2 é um **framework** para Python para **dividir, mesclar, recortar e transformar páginas em seus PDFs**. Ele fornece diversas funções para adicionar dados, opções de visualização e senhas aos PDFs.

Podemos minerar/manipular diversas formas os PDFs:
- ✓ Extrair dados;
- ✓ Dividir Documentos;
- ✓ Rotacionar documentos;
- ✓ Inserir marcas d'agua;
- ✓ E outros.

Python-Module PyPDF2...

Dados não estruturados

Os **pdfs** fazem parte dos **dados não estruturados**, o que é diferente seu tratamento quanto aos dados estruturados (Tabelas, Planilhas e etc).

Minerar esse tipo de informação é um **pouco diferente** quanto a outros tipos de dados.

Vou explorar apenas algumas funções e situações nesse tutorial.

Mas caso esteja trabalhando com esse tipo de informação é importante se aprofundar no Framework e explorar suas funções.

@Odemir Depieri Jr

Vamos instalar o framework

```
pip install pypdf2
```

```
Collecting pypdf2
  Downloading PyPDF2-1.26.0.tar.gz (77 kB)
     |████████████████████████████████| 77 kB 3.1 MB/s
Building wheels for collected packages: pypdf2
  Building wheel for pypdf2 (setup.py) ... done
  Created wheel for pypdf2: filename=PyPDF2-1.26.0-py3-none-any.whl size=61101 sha256=c41
  Stored in directory: /root/.cache/pip/wheels/80/1a/24/648467ade3a77ed20f35cfd2badd32134
Successfully built pypdf2
Installing collected packages: pypdf2
Successfully installed pypdf2-1.26.0
```

Vamos iniciar importando a função para ler os PDS

```
[10] # Importando a Função para ler o PDF
     from PyPDF2 import PdfFileReader
```

Lendo um PDF

```
[11] # Lendo o Arquivo PDF
     Leitor = PdfFileReader( open('Guia sobre WordCloud.pdf', 'rb') )
```

Para esse tutorial pode se utilizar qualquer PDF.

Identificando a quantidade de Paginas no PDF

```
[12] # Identificando o Numero de Paginas do PDF
     Numeros_Paginas = Leitor.getNumPages()
     print('Numeros de Paginas no arquivo: ', Numeros_Paginas )
```

```
Numeros de Paginas no arquivo:  7
```

Identificando as informações da construção do PDF

```
[13] # Identificando as informações da construção do PDF
     Informações = Leitor.getDocumentInfo()
     Informações
```

```
{'/Author': 'POSITIVO-7336NT',
 '/CreationDate': "D:20210523171110-03'00'",
 '/Creator': 'Microsoft® PowerPoint® 2013',
 '/ModDate': "D:20210523171110-03'00'",
 '/Producer': 'Microsoft® PowerPoint® 2013',
 '/Title': 'Apresentação do PowerPoint'}
```

Normalmente em arquivos PDFs são **registrado diversas informações** sobre sua criação.
Assim com essa função podemos verificar todas as essa infos.

Podemos extrair individualmente essas informações

```
# Pegando apenas o Ator
Ator = Informações.author
print( Ator )

# Pegando apenas o Criado
Criador = Informações.creator
print( Criador )

# Pegando apenas o Produtor
Produtor = Informações.producer
print( Produtor )

# Pegando apenas o Sujeito
Sujeito = Informações.subject
print( Sujeito )

# Pegando apenas o Titulo
Titulo = Informações.title
print( Titulo )
```

```
POSITIVO-7336NT
Microsoft® PowerPoint® 2013
Microsoft® PowerPoint® 2013
None
Apresentação do PowerPoint
```

Lendo uma pagina e extraindo o conteúdo da Pagina

```
[14] # Acessando a pagina numero 1 do PDF
     Acessando_Pagina = Leitor.getPage(1)

     # Extrair o Texto do PDF
     Extraindo_Textos = Acessando_Pagina.extractText()
     print( Extraindo_Textos )
```

```
@Odemir Depieri Jr
Guia sobre
wordcloud
Nuvem de Palavras [ Word
Cloud
]
Nuvem de palavras (
word
cloud
) é um
gráfico
digital que mostra
o grau de
frequência das palavras
em um texto. Quanto mais a
palavra é utilizada, mais chamativa é a representação dessa
palavra no gráfico.
Exemplos de uma Nuvem de Palavras
```

Com apenas algumas linhas de comando é possível extrair todo o conteúdo do PDF

Selecionando outra pagina e extraindo o conteúdo

```
# Acessando a pagina numero 2 do PDF
Acessando_Pagina = Leitor.getPage(2)

# Extrair o Texto do PDF
Extraindo_Textos = Acessando_Pagina.extractText()
print( Extraindo_Textos )
```

```
@Odemir Depieri Jr
Definindo as palavras
Vamos importar as bibliotecas externas que precisamos
Lorem
Ipsum é simplesmente uma simulação de texto da indústria tipográfica e
de impressos, e vem sendo utilizado desde o século XVI, quando um impressor
desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro
de modelos de tipos.
Lorem
Ipsum sobreviveu não só a cinco séculos, como
também ao salto para a editoração eletrônica, permanecendo essencialmente
inalterado. Se popularizou na década de 60, quando a
Letraset
lançou
decalques contendo passagens de
Lorem
Ipsum, e mais recentemente quando
passou a ser integrado a softwares de editoração eletrônica como
Aldus
PageMaker Ao
```

Varrendo o PDF em um loop e extraindo todo o conteúdo

```
# Variavel para armazenar o texto extraido do PDF
Texto_Extraido = ''

# Loop para varrar o PDf
for Paginas in range(Numeros_Paginas):

  # Acessando a Pagina do PDF de acordo com o Loop
  Acessando_Pagina = Leitor.getPage(Paginas)

  # Extraindo as informações
  Extraindo_Textos = Acessando_Pagina.extractText()

  # Concatenando o texto do PDF
  Texto_Extraido =  Texto_Extraido + Extraindo_Textos

print( Texto_Extraido )
```

```
palavra no gráfico.
Exemplos de uma Nuvem de Palavras
@Odemir Depieri Jr
Definindo as palavras
Vamos importar as bibliotecas externas que precisamos
Lorem
Ipsum é simplesmente uma simulação de texto da indústria tipográfica e
de impressos, e vem sendo utilizado desde o século XVI, quando um impressor
desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro
de modelos de tipos.
Lorem
Ipsum sobreviveu não só a cinco séculos, como
também ao salto para a editoração eletrônica, permanecendo essencialmente
inalterado. Se popularizou na década de 60, quando a
Letraset
lançou
decalques contendo passagens de
Lorem
```

Gerando uma base analítica com as informações do PDF

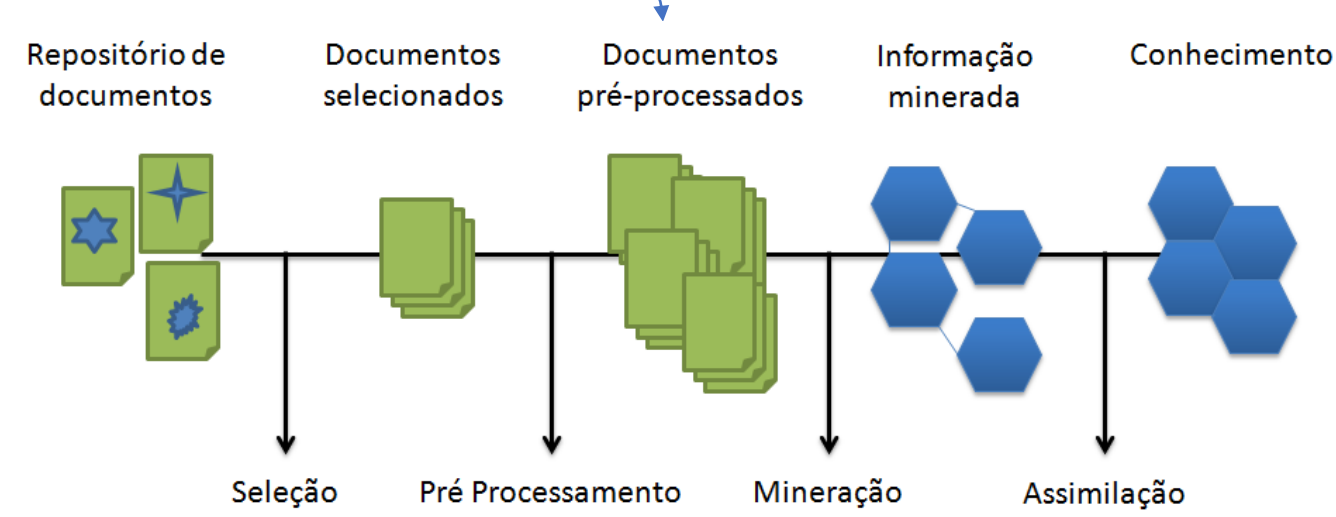
```
[33] # Gerando uma base analitica com todas as informações do PDF

     # Vamos contar quantas palavras há em nosso PDF
     Total_Palavras = len( Texto_Extraido.split(' ') )

     # Criando um Dicionario com os dados organizado
     Dicionario = [{
         'Ator':Ator,
         'Criador':Criador,
         'Produtor':Produtor,
         'Sujeito':Sujeito,
         'Titulo':Titulo,
         'Numero de Paginas': Numeros_Paginas,
         'Conteudo do PDf': Texto_Extraido,
         'Total de Palavras': Total_Palavras
     }]

     # Importando uma função do Pandas para ordenar os dados
     from pandas import DataFrame

     # Transformando o Dicionario em um DataFrame
     Base = DataFrame( Dicionario )
     Base
```

	Ator	Criador	Produtor	Sujeito	Titulo	Numero de Paginas	Conteudo do PDf	Total de Palavras
0	POSITIVO-7336NT	Microsoft® PowerPoint® 2013	Microsoft® PowerPoint® 2013	None	Apresentação do PowerPoint	7	Guia rápido \nsobre criar \nNuvem de \nPalavra...	387

Com esse script é possível **extrair diversas informações** dos PDFs e gerar uma **base analítica** com as informações.
Essa técnica é ótima para **consolidar as informações** dos PDFs.

Sugiro melhorar esse script de acordo com a sua necessidade.

A coluna com o conteúdo do PDF é perfeita para ser usada para técnicas de **Text Mining.**



Rotações nas paginas

```
[36] # Rotação com paginas do PDF
     # Aqueles problemas da pagina estar invertida ou fora da posição

     # Rotacionado 90 graus para a Direita
     Leitor.getPage(0).rotateClockwise(90)

     # Rotacionado 90 graus para a Esquerda
     Leitor.getPage(0).rotateCounterClockwise(90)

     print('')
```

Criando um novo PDF a partir de outros PDFs

```
[40] # Função para criar/mesclar um PDF
     from PyPDF2 import PdfFileWriter

     # Atribuindo a função para criar o PDF
     Funcao_Criar = PdfFileWriter()

     # Abrindo 3 Arquvios de  PDF
     Arquivo_1 = PdfFileReader(open('Guia sobre Seaborn.pdf', 'rb'))
     Arquivo_2 = PdfFileReader(open('Guia sobre Serie Temporal.pdf', 'rb'))
     Arquivo_3 = PdfFileReader(open('Guia sobre WordCloud.pdf', 'rb'))

     # Selecionado diferentes paginas dos PDFs
     PDF_01 = Arquivo_1.getPage(1)
     PDF_02 = Arquivo_1.getPage(3)
     PDF_03 = Arquivo_1.getPage(2)

     # Adicionando essas paginas no novo PDf
     Funcao_Criar.addPage(PDF_01)
     Funcao_Criar.addPage(PDF_02)
     Funcao_Criar.addPage(PDF_03)

     # Escrendo o novo PDF e exportando
     with open('Novo_PDF.pdf', 'wb') as fh:
             Funcao_Criar.write(fh)
```

Incluindo Criptografia

```
[49] # Atribuindo a função para criar o PDF
     Funcao_Criar = PdfFileWriter()

     # Abrindo 3 Arquvios de  PDF
     Arquivo_1 = PdfFileReader(open('Guia sobre Seaborn.pdf', 'rb'))
     Arquivo_2 = PdfFileReader(open('Guia sobre Serie Temporal.pdf', 'rb'))

     # Selecionado diferentes paginas dos PDFs
     PDF_01 = Arquivo_1.getPage(1)
     PDF_02 = Arquivo_1.getPage(1)


     # Adicionando essas paginas no novo PDf
     Funcao_Criar.addPage(PDF_01)
     Funcao_Criar.addPage(PDF_02)

     # Criando criptografia para o PDF
     Funcao_Criar.encrypt(
       user_pwd='Escolha a Senha',
       owner_pwd=None
     )

     # Escrendo o novo PDF e exportando
     with open('Novo_PDF - Com Senha.pdf', 'wb') as fh:
             Funcao_Criar.write(fh)
```

Incluindo Marca D'agua

```
[51] # Atribuindo a função para criar o PDF
     Funcao_Criar = PdfFileWriter()

     # Abrindo 3 Arquvios de  PDF
     Arquivo_1 = PdfFileReader(open('Guia sobre Seaborn.pdf', 'rb'))
     Arquivo_2 = PdfFileReader(open('Guia sobre Serie Temporal.pdf', 'rb'))

     # Selecionado diferentes paginas dos PDFs
     PDF_01 = Arquivo_1.getPage(1)
     PDF_01.mergePage(PDF_03)

     PDF_02 = Arquivo_1.getPage(1)
     PDF_02.mergePage(PDF_03)

     # Adicionando essas paginas no novo PDf
     Funcao_Criar.addPage(PDF_01)
     Funcao_Criar.addPage(PDF_02)

     # Escrendo o novo PDF e exportando
     with open('PDF com Marca Dagua.pdf', 'wb') as fh:
             Funcao_Criar.write(fh)
```

Final

Esse guia rápido é para ter conhecimentos prévios sobre como utilizar a biblioteca **PyPDF2**
Caso queira mais informações, acesse a documentação oficial do framework.

Guia da documentação caso queira mais detalhes
https://pythonhosted.org/PyPDF2/index.html

PyPDF2

Odemir Depieri Jr

Intelligence Analyst Sr
Tech Lead
Specialization AI