Interagindo com Arquivos de Texto

Flávio Codeço Coelho
FGV-EMAp
(Dated: September 8, 2019)

CONTENTS

I. ABRINDO ARQUIVOS DE TEXTO

Neste curso de mineração de textos usaremos como material principal de trabalho, os verbetes do Dicionário Histórico e Biográfico do Brasil – DHBB. Estes verbetes stão disponíveis para Download público.

Neste capítulo vamos aprender a interagir com os verbetes no disco e extrair informações simples a partir dos mesmos.

Vamos começar importando alguma bibliotecas que nos serão úteis nesta tarefa:

```
In [1]: import os
        import glob
```

Assumindo que os dados do DHBB já foram baixados para um diretório local, podemos começar inspecionando o diretório e listando o seu conteúdo.

```
In [4]: caminho = "../dhbb/text/*.text"
        arquivos  = glob.glob(caminho)
        len(arquivos)
```

```
[4]: 7687
```

Temos 7687 verbetes neste diretório. Vamos agora ver como abrir um destes verbetes e inspecionar o seu conteúdo:

```
In [5]: arquivos[0]
```

```
[5]: '../dhbb/text/1956.text'
```

```
In [6]: with open(arquivos[0],'r') as f:
            verbete = f.read()
            print(verbete)
```

```
---
title: FERRAZ, Gabriel Lopes
natureza: biográfico
```

```
sexo: m
cargos:
 - dep. fed. RJ 1960
---

ńGabriel Lopes Ferrazż nasceu no Rio de Janeiro, então Distrito Federal,
no dia 20 de dezembro de 1907, filho de Álvaro Lopes Ferraz e de Maria
Amélia dos Santos Ferraz.

Médico, elegeu-se deputado estadual à Assembléia Legislativa do Estado
do Rio de Janeiro na legenda do Partido Social Progressista (PSP) em
outubro de 1954. Assumiu o mandato no início do ano seguinte,
exercendo-o até janeiro de 1959. No pleito de outubro de 1958, disputou
uma vaga, pelo PSP, na Câmara dos Deputados na bancada do Rio de
Janeiro, mas obteve apenas uma suplência. Exerceu o mandato de julho a
dezembro de 1960.

Com a extinção dos partidos políticos pelo Ato Institucional nž 2
(27/10/1965) e a posterior instauração do bipartidarismo, filiou-se à
Aliança Renovadora Nacional (Arena), partido de apoio ao regime militar
instaurado no país em março de 1964. Por esta legenda, candidatou-se a
uma cadeira na Assembléia Legislativa do Rio de Janeiro no pleito de
outubro de 1966, não sendo bem-sucedido.

Em novembro de 1970, concorreu à prefeitura de Nilópolis, perdendo
novamente. Contudo, no pleito de novembro de 1972 elegeu-se
vice-prefeito de Nilópolis, sempre pela Arena. Desincompatibilizando-se
do cargo executivo, candidatou-se a uma cadeira na Assembléia
Legislativa, pela legenda da Arena, no pleito de novembro de 1974, não
logrando êxito.
```

A variável **verbete** que criamos na célula anterior é uma variável do tipo `string`, que é a otipo usado pelo Python para representar um bloco de texto. Podemos manipular o texto dentro de uma `string` de diversas maneiras:

```
In [7]: print(verbete.split('---')[1])
```

```
title: FERRAZ, Gabriel Lopes
natureza: biográfico
sexo: m
cargos:
 - dep. fed. RJ 1960
```

Tipos de dados em Python, também conhecidos como objetos, possuem métodos. O método `split` do tipo `string` segmenta uma string nas posicões em que ocorram uma sequência específica de caracteres, retornando um outro tipo de dado, denominado `lista`.

```
In [8]: type(verbete.split('---'))
```

```
[8]: list
```

Listas são sequencias de objetos de quaisquer tipos que também apresenta seu conjunto de métodos. Para descobrir os métodos de qualquer objeto, basta colocar um ponto após o nome da variável e pressionar a tecla <tab>. Listas são delimitadas por colchetes: [] (lista vazia).

```
In [9]: l = verbete.split('---')
        l
```

```
[9]: ['',
      '\ntitle: FERRAZ, Gabriel Lopes\nnatureza: biográfico\nsexo: m\ncargos: \n -
     dep. fed. RJ 1960\n',
      '\n\nńGabriel Lopes Ferrazż nasceu no Rio de Janeiro, então Distrito
     Federal,\nno dia 20 de dezembro de 1907, filho de Álvaro Lopes Ferraz e de
     Maria\nAmélia dos Santos Ferraz.\n\nMédico, elegeu-se deputado estadual à
     Assembléia Legislativa do Estado\ndo Rio de Janeiro na legenda do Partido Social
     Progressista (PSP) em\noutubro de 1954. Assumiu o mandato no início do ano
     seguinte,\nexercendo-o até janeiro de 1959. No pleito de outubro de 1958,
     disputou\numa vaga, pelo PSP, na Câmara dos Deputados na bancada do Rio
     de\nJaneiro, mas obteve apenas uma suplência. Exerceu o mandato de julho
     a\ndezembro de 1960.\n\nCom a extinção dos partidos políticos pelo Ato
     Institucional nž 2\n(27/10/1965) e a posterior instauração do bipartidarismo,
     filiou-se à\nAliança Renovadora Nacional (Arena), partido de apoio ao regime
     militar\ninstaurado no país em março de 1964. Por esta legenda, candidatou-se
     a\numa cadeira na Assembléia Legislativa do Rio de Janeiro no pleito de\noutubro
     de 1966, não sendo bem-sucedido.\n\nEm novembro de 1970, concorreu à prefeitura
     de Nilópolis, perdendo\nnovamente. Contudo, no pleito de novembro de 1972
     elegeu-se\nvice-prefeito de Nilópolis, sempre pela Arena. Desincompatibilizando-
     se\ndo cargo executivo, candidatou-se a uma cadeira na Assembléia\nLegislativa,
     pela legenda da Arena, no pleito de novembro de 1974, não\nlogrando
     êxito.\n\n\n']
```

Note que nas strings acima existem várias ocorrências da sequencia de caracteres '\n'. Esta sequência identifica quebra de linhas. Podemos então utilizá-la para dividir o cabeçalho do verbete em uma lista de linhas:

```
In [10]: cabeçalho = verbete.split('---')[1]
         cabeçalho.split('\n')
```

```
[10]: ['',
       'title: FERRAZ, Gabriel Lopes',
       'natureza: biográfico',
       'sexo: m',
       'cargos: ',
       ' - dep. fed. RJ 1960',
       '']
```

Elementos de uma lista podem ser acessado por sua posição na sequência, por exemplo:

```
In [11]: print(l[2])
```

```
ńGabriel Lopes Ferrazż nasceu no Rio de Janeiro, então Distrito Federal,
no dia 20 de dezembro de 1907, filho de Álvaro Lopes Ferraz e de Maria
Amélia dos Santos Ferraz.

Médico, elegeu-se deputado estadual à Assembléia Legislativa do Estado
do Rio de Janeiro na legenda do Partido Social Progressista (PSP) em
```

```
outubro de 1954. Assumiu o mandato no início do ano seguinte,
exercendo-o até janeiro de 1959. No pleito de outubro de 1958, disputou
uma vaga, pelo PSP, na Câmara dos Deputados na bancada do Rio de
Janeiro, mas obteve apenas uma suplência. Exerceu o mandato de julho a
dezembro de 1960.

Com a extinção dos partidos políticos pelo Ato Institucional nž 2
(27/10/1965) e a posterior instauração do bipartidarismo, filiou-se à
Aliança Renovadora Nacional (Arena), partido de apoio ao regime militar
instaurado no país em março de 1964. Por esta legenda, candidatou-se a
uma cadeira na Assembléia Legislativa do Rio de Janeiro no pleito de
outubro de 1966, não sendo bem-sucedido.

Em novembro de 1970, concorreu à prefeitura de Nilópolis, perdendo
novamente. Contudo, no pleito de novembro de 1972 elegeu-se
vice-prefeito de Nilópolis, sempre pela Arena. Desincompatibilizando-se
do cargo executivo, candidatou-se a uma cadeira na Assembléia
Legislativa, pela legenda da Arena, no pleito de novembro de 1974, não
logrando êxito.
```

```
In [12]: campos = {l.split(':')[0].strip() :l.split(':')[1].strip() for l in
         cabeçalho.split('\n')[:4] if l}
         campos
```

[12]: `{'title': 'FERRAZ, Gabriel Lopes', 'natureza': 'biográfico', 'sexo': 'm'}`

Na célula acima contruímos um novo tipo de variável chamada *Dicionário*, é basicamente um conjunto de pares, delimitado por `{}`. Estes pares são chamados pares `chave: valor`.

I.1. Abrindo um grande número de documentos texto

Como vimos acima existem 7687 verbetes à nossa disposição no disco, mas não podemos abrir todos ao mesmo tempo pois, em primeiro lugar podem não caber na memória, em segundo lugar raramente precisaremos inpecioná-los todos ao mesmo tempo. O mais comum é analisá-los em sequência. Vamos inspecionar os primeiros 10:

```
In [13]: for a in arquivos[:10]:
             with open (a, 'r') as f:
                 verbete = f.readlines()
             print('Verbete: ', a.split('.text')[0].split('/')[-1])
             print(verbete[1])
```

```
Verbete:  1956
title: FERRAZ, Gabriel Lopes

Verbete:  10978
title: DESCONSI, Orlando

Verbete:  2687
title: LACERDA, Jorge

Verbete:  3429
```

```
title: MELO, Geraldo Medeiros de

Verbete:  2839
title: LIMA, Alceu Amoroso

Verbete:  2088
title: FONTES, Tomás

Verbete:  11055
title: FROSSARD, Denise

Verbete:  11650
title: MURAD, Jamil

Verbete:  12159
title: SÁ, Liliam

Verbete:  4940
title: SEIXAS, Luís Siqueira
```

```
In [14]: arquivos[1]
```

```
[14]: '../dhbb/text/10978.text'
```

Acima utilizamos uma estrutura de repetição, denominada "laço for" para abrir sequencialmente os arquivos. É importante notar que a cada volta do laço, o arquivo texo é atribuído à mesma variável, o que significa que nunca há mais do que apenas um verbete na memória. Desta forma poderíamos potencialmente analisar todos os milhares de verbetes ocupando apenas uma quantidade pequena e constante de memória. Outro detalhe do código acima é que, para facilitar a extração do título do verbete, Fizemos a leitura do arquivo com o método `readlines` que retorna o verbete já divido em uma lista de linhas ao invés de uma `string`.

I.2. Outros recursos do DHBB

O arquivo do DHBB disponível no Github oferece outros recursos textuais para nos auxiliar em nossa pesquisa, como por exemplos dicionários com identificadores de "Entidades" presentes nos verbetes, como pessoas, organizações, eventos, etc.

```
In [26]: with  open("../dhbb/dic/pessoa-individuo.txt", 'r') as f:
             pessoas = f.readlines()
         pessoas[:10]
```

```
[26]: ['Aarão Rebelo\n',
       'Aarão Steinbruch\n',
       'Abalcazar Garcia\n',
       'Abdias Do Nascimento\n',
       'Abdon Goncalves Nanhay\n',
       'Abdon Gonçalves\n',
       'Abdon Sena\n',
       'Abdon de Mello\n',
       'Abdur R. Khan\n',
       'Abel Avila dos Santos\n']
```

In [28]:
```
with  open("../dhbb/dic/pessoa-papel.txt", 'r') as f:
    profissão = f.readlines()
profissão[:10]
```

[28]:
```
['Advogado\n',
 'Advogado Geral da União\n',
 'Agente de investimento\n',
 'Agente de segurança judiciária\n',
 'Alfaiate\n',
 'Analista administrativo\n',
 'Analista de comércio exterior\n',
 'Antiquário\n',
 'Arcebispo\n',
 'Armador\n']
```

In [29]:
```
with  open("../dhbb/dic/evento.txt", 'r') as f:
    evento = f.readlines()
evento[:10]
```

[29]:
```
['A Rusga\n',
 'ATENTADO DA TONELEIROS\n',
 'ATENTADO DO RIOCENTRO\n',
 'Aclamação de Amador Bueno\n',
 'Balaiada\n',
 'Batalha da Maria Antônia\n',
 'Batalha da Venda Grande\n',
 'Batalha das Toninhas\n',
 'Batalha de Santa Luzia\n',
 'COMÍCIO DAS REFORMAS\n']
```

In [30]:
```
with  open("../dhbb/dic/organizacao.txt", 'r') as f:
    organização = f.readlines()
organização[:10]
```

[30]:
```
['Abrigo Lar dos Velhos Vicentini\n',
 'Academia Alagoana de Letras\n',
 'Academia Brasileira de Ciências\n',
 'Academia Brasileira de Ciências Econômicas e Administrativas\n',
 'Academia Brasileira de Ciências Sociais e Políticas\n',
 'Academia Brasileira de Direito Empresarial\n',
 'Academia Brasileira de Letras\n',
 'Academia Brasileira de Música\n',
 'Academia Brasiliense de Letras\n',
 'Academia Cultural de Curitiba\n']
```

In [32]:
```
with  open("../dhbb/dic/formulacao-politica.txt", 'r') as f:
    politica = f.readlines()
politica[:10]
```

[32]:
```
['anteprojeto Constitucional\n',
 'anteprojeto da Carta Magna\n',
 'anteprojeto da Comissão Provisória\n',
 'anteprojeto da Comissão Provisória de Estudos Constitucionais\n',
```

```
 'anteprojeto da Comissão de Sistematização\n',
 'anteprojeto da Consolidação das Leis do Trabalho\n',
 'anteprojeto da Constituição\n',
 'anteprojeto da Lei Orgânica da Magistratura\n',
 'anteprojeto da Lei de Acidentes no Trabalho\n',
 'anteprojeto da Lei de Direitos Autorais\n']
```

II. EXTRAINDO INFORMAÇÃO ESTRUTURADA

Agora que sabemos como abrir arquivos de texto e ler o seu conteúdo, podemos experimentar a extração de imformações específicas dos verbetes e organizá-la em uma tabela. Para isso vamos lançar mão de uma biblioteca chamada **Pandas** para organizar em uma estrutura tabular, chamada `DataFrame` os dados que vamos extrair.

```
In [15]: import pandas as pd
         pd.set_option("display.latex.repr", True)
```

Nós vimos acima que os verbetes contém uma seção inicial delimitada pelos caracteres `---` vamos utilizar esta característica do texto para guiar nossa extração de informação. Como você pode perceber, já começamos a reutilizar código que escrevemos anteriormente. Para facilitar o reuso e reduzir a necessidade de escrever múltiplas vezes o mesmo código vamos aprender a organizá-lo melhor. Vamos começar definindo uma função.

```
In [16]: def tabula_verbete(n=None):
             """
             Carrega todos os verbetes disponíveis, ou os primeiros n.
             n:  número de verbetes a tabular
             """
             if n is None:
                 n = len(arquivos)
             linhas = []
             for a in arquivos[:n]:
                 with open (a, 'r') as f:
                     verbete = f.read()
                 cabeçalho = verbete.split('---')[1]
                 campos = {l.split(':')[0].strip() :l.split(':')[1].strip() for l in
         cabeçalho.split('\n')[:4] if l}
                 campos['arquivo'] = os.path.split(a)[1]
         #         campos['cargos'] = cabeçalho.split('cargos:')[1]
         #         campos['corpo'] = verbete.split('---')[2]
                 linhas.append(campos)
             tabela = pd.DataFrame(data = linhas, columns=['arquivo','title', 'natureza',
         'sexo'])
             return tabela
```

A função acima inclui a maior parte do código que escrevemos anteriormente, só que encapsulado em uma função que nos permite executar a extração e tabulação do cabeçalho para o numero de verbetes que desejarmos. Podemos ver abaixo que na verdade é muito rápido processar todos os verbetes.

```
In [17]: help(tabula_verbete)
```

```
Help on function tabula_verbete in module __main__:

tabula_verbete(n=None)
    Carrega todos os verbetes disponíveis, ou os primeiros n.
    n:  número de verbetes a tabular
```

```
In [18]: tab = tabula_verbete()
         tab.head()
```

[18]:

	arquivo	title	natureza	sexo
0	1956.text	FERRAZ, Gabriel Lopes	biográfico	m
1	10978.text	DESCONSI, Orlando	biográfico	m
2	2687.text	LACERDA, Jorge	biográfico	m
3	3429.text	MELO, Geraldo Medeiros de	biográfico	m
4	2839.text	LIMA, Alceu Amoroso	biográfico	m

Podemos visualizar uma descrição básica da tabela resultante

```
In [19]: tab.describe()
```

[19]:

	arquivo	title	natureza	sexo
count	7687	7687	7687	6722
unique	7687	7596	2	2
top	1638.text	ALBUQUERQUE, Carlos	biográfico	m
freq	1	3	6724	6517

Por exemplo fica fácil ver que no DHBB predominam biografias de personagens do sexo masculino.

```
In [20]: print(tab.sexo.value_counts())
```

```
m    6517
f     205
Name: sexo, dtype: int64
```

Percebemos também que a natureza predominante dos verbetes é biográfica e que só existem duas naturezas, mas qua a outra?

```
In [21]: print(tab.natureza.value_counts())
```

```
biográfico    6724
temático       963
Name: natureza, dtype: int64
```

III. EXPORTANDO PARA BANCOS DE DADOS

Depois de realizarmos a nossa análise e tabular os resultados, podemos exportar a tabela em vários formatos. Em primeiro lugar, caso queiramos abri nossa trabalho em uma planilha, devemos salvar no formato CSV, ou "comma-separated-values". Este formato pode ser aberto imediatamente em uma planilha.

```
In [22]: tab.to_csv("minha_tabela.csv", sep='|')
```

Acima usamos o caracter "|" como separador para evitar confusões com as virgulas existentes no texto. ## Exportando para um banco de dados relacional Para exportar para um banco relacional, precisamos de uma biblioteca adicional, o SQLAlchemy. Estabiblioteca nos permite interagir com a maioria dos banco relacionais. Aqui vamos usar o banco SQLite.

```
In [32]: from sqlalchemy import create_engine
```

```
In [34]: engine = create_engine('sqlite:///minha_tabela.sqlite', echo=False)
         tab.to_sql('resultados', con=engine, if_exists='append')
```

Uma vez inserido no banco relacional, podemos fazer consultas aos dados usando a linguagem SQL. Abaixo obtemos o resultado da consulta em uma lista.

```
In [39]: engine.execute("select * from resultados where natureza='temático'").
 ↪fetchall()[:10]
```

[39]:
```
[(354, '10989.text', 'Destacamento de Operações e Informações  Centro de
Operações e Defesa Interna (DOI-CODI)', 'temático', None),
 (1027, '11595.text', 'Agência Brasileira de Inteligência (Abin)', 'temático',
None),
 (1028, '11596.text', 'Associação Brasileira de Emissoras de Rádio e Televisão
(ABERT)', 'temático', None),
 (1029, '11597.text', 'Associação Nacional de Jornais (ANJ)', 'temático', None),
 (1030, '11598.text', 'Associação Nacional dos Membros do Ministério Público
(CONAMP)', 'temático', None),
 (1031, '11599.text', 'CAROS AMIGOS', 'temático', None),
 (1034, '11600.text', 'CARTA CAPITAL', 'temático', None),
 (1035, '11601.text', 'Central dos Trabalhadores e das Trabalhadoras do Brasil
(CTB)', 'temático', None),
 (1036, '11602.text', 'Central Geral dos Trabalhadores do Brasil (CGTB)',
'temático', None),
 (1037, '11603.text', 'Conselho de Comunicação Social (CCS)', 'temático', None)]
```

Se quisermos os resultado na forma de um `Dataframe`, podemos usar o **Pandas**.

In [40]:
```
pd.read_sql_query("select * from resultados where natureza='temático'",
        con=engine).head()
```

[40]:

	index	arquivo	title	natureza	sexo
0	354	10989.text	Destacamento de Operações e Informações Cent...	temático	None
1	1027	11595.text	Agência Brasileira de Inteligência (Abin)	temático	None
2	1028	11596.text	Associação Brasileira de Emissoras de Rádio e ...	temático	None
3	1029	11597.text	Associação Nacional de Jornais (ANJ)	temático	None
4	1030	11598.text	Associação Nacional dos Membros do Ministério ...	temático	None

IV. EXERCÍCIOS

1. Construa uma função para buscar apenas verbetes de personagens que tenham ocupado o cargo de deputado federal. Tabule os resultados incluindo o número de mandatos.
2. Construa uma função para buscar o primeiro verbete temático e apresente o seu conteúdo.
3. Encontre todos os verbetes que contenham “Academia Brasileira de Letras”. Que porcentagem destes correspondem a membros da dita academia?
4. Construa uma linha do tempo que represente a cobertura histórica do DHBB.

In [ ]: