DOU - Natural Language Processing

Paulo Cardoso

20/05/2019

Introdução

O objetivo deste documento é apresentar uma analise textual, tambem conhecida como processamento de linguagem natural. Como objeto da analise estão os dados abertos do diario oficial da união, obtidos no portal www.dados.gov.br.

Esta analise foi desenvolvida utilizando duas linguagens de programação, sendo a primeira Python que foi utilizado para a extração do dos dados dos arquivos XML e conversão para o formato tabular e a otra linguagem de programação utilizada foi R, que foi utilizado para realizar a mineração de texto e construção do wordcloud.

Apos realizar o download dos dados, e colocar os arquivos .zip no mesmo diretorio dos scripts a primeira parte da analise pode ser excecutada usando python. Para possibilitar a utilização dalinguagem Python nesse documento R Markdown foi utilizada a biblioteca reticulate.

```
require(reticulate)
```

```
## Loading required package: reticulate
```

1 - Descompactação, extração e transformação dos dados

Como primeiro passo, as bibliotecas Python são importadas:

```
# Loading Libraries
import os, zipfile, inspect, glob, re
import xml.etree.cElementTree as et
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Apos importar as bibliotecas a descompactação pode ser realizada, como pode ser observado abaixo:

```
# getting file name and directory
filename = inspect.getframeinfo(inspect.currentframe()).filename
dir_name = os.path.dirname(os.path.abspath(filename))

# Unzip
for item in os.listdir(dir_name):
    if item.endswith(".zip"):
        zip_ref = zipfile.ZipFile(item)
        zip_ref.extractall()
        print(item + " descompactado.")
```

```
## S03012019.zip descompactado.
## S01012019.zip descompactado.
## S02012019.zip descompactado.
```

Criando pandas Dataframe que recebera os dados extraidos dos XML.

```
# defining column names
dfcols = ['id', 'name', 'idOficio', 'pubName', 'artType', 'pubDate', 'artClass', 'artCategory', 'artSiz

# creating pandas dataframe
df_xml = pd.DataFrame(columns=dfcols)
```

Extraindo dados dos arquivos XML e populando do dataframe criado acima:

```
# extracting data from XMl and populating dataframe
for file in glob.iglob(os.path.join(dir_name, '*/*.xml')):
    tree = et.parse(file)
    for node in tree.getroot():
        id = node.attrib.get('id')
        name = node.attrib.get('name')
        idOficio = node.attrib.get('idOficio')
        pubName = node.attrib.get('pubName')
        artType = node.attrib.get('artType')
        pubDate = node.attrib.get('pubDate')
        artClass = node.attrib.get('artClass')
        artCategory = node.attrib.get('artCategory')
        artSize = node.attrib.get('artSize')
        artNotes = node.attrib.get('artNotes')
        numberPage = node.attrib.get('numberPage')
        pdfPage = node.attrib.get('pdfPage')
        editionNumber = node.attrib.get('editionNumber')
        highlightType = node.attrib.get('highlightType')
        highlightPriority = node.attrib.get('highlightPriority')
        highlight = node.attrib.get('highlight')
        idMateria = node.attrib.get('idMateria')
        Identifica = node.find('body/Identifica').text
        Data = node.find('body/Data').text
        Ementa = node.find('body/Ementa').text
        Titulo = node.find('body/Titulo').text
        SubTitulo = node.find('body/SubTitulo').text
        Texto = node.find('body/Texto').text
        df_xml = df_xml.append(pd.Series([id, name, idOficio, pubName, artType, pubDate, artClass, artC
```

Com o intuito de facilitar a segunda arte da analise a mineração de texto serão aplicadas algumas tecnicas de limpeza de dados textuais para facilitar a etapa seguinte do trabalho.

```
# copying to dataframe to new variable
df = df_xml.copy()

# removing spaces, numbers in date format and some special chatracters on the field Texto
for i in range(len(df)):
    df.Texto[i] = re.sub('<[^<]+?>', ' ', df.Texto[i])
    df.Texto[i] = re.sub('[-|0-9]',' ', df.Texto[i])
    df.Texto[i] = re.sub(r'[-./?!,":;()\']',' ', df.Texto[i])
```

Para conhecer um pouco sobre o dataset criado acima, podemos observar abaixo as 5 primeiras entradas e a seguir a descrição do dataset.

```
# presenting first 5 entries
df.head(5)
```

```
##         id  ...                                              Texto
```

```
## 0  7311944  ...   ATO Nº      DE    DE JANEIRO DE       Expede ...
## 1  6913734  ...    ANEXO II   a  QUADRO DEMONSTRATIVO DOS CARGO...
## 2  7159090  ...   PORTARIA Nº     DE    DE JANEIRO DE       Div...
## 3  6983174  ...     PORTARIA Nº    DE   DE JANEIRO DE       Dis...
## 4  7062206  ...                                    Unimed Norte...
##
## [5 rows x 23 columns]
```

```
# describing dataset
df.describe()
```

```
##              id  ...                                                Texto
## count     63504  ...                                                63504
## unique    63504  ...                                                63113
## top     7253784  ...   RETIFICAÇÃO  Subrogada pela UASG           UN...
## freq          1  ...                                                   10
##
## [4 rows x 23 columns]
```

Como encerramento da etapa de descompactação, extração e transformação de dados ocorre a persistencia do dataframe criado em disco para futura utilização.

```
# writing data on disc
df.to_csv('DOU1901.csv', sep = ',')
```

2 - Processamento de Linguagem Natural

Esta fase da analise apresenta tecnicas de mineração de texto, limpeza de dados textuais, tokenização e apresentação de wordclouds. Alem disso, foi desenvolvida utilizando a linguagem R com o suporte de bibliotecas relacionadas a text mining e datas.

Como primeiro passo, serão carregadas as bibliotecas utilizadas.

```
# loading libraries
library(tm)
```

```
## Loading required package: NLP
```

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(readr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

Os dados criados na primeira etapa e salvos em disco são carregados.

```
# data load
df <- read_csv("DOU1901.csv", col_types = cols(X1 = col_skip()))
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Warning: 1 parsing failure.
##  row       col               expected actual          file
## 2533 idMateria no trailing characters     -2 'DOU1901.csv'
```

2.1 - Limpeza e tratamento de Dados

Para que os dados do DOU estejam prontos para a analise é nessario que passem por diversas limpezas e transformações. Dentre os processos realizados estão: correção de formato de datas, criação de variaveis, agrupamento de variavais, remoção de acentuação e conversões de formato.

```
# substituindo / por - em pubdate, convertendo para data e criando campo de numero da semana paras egme
df$pubDate <- gsub('/', '-', df$pubDate)
df$pubDate <- dmy(df$pubDate)
df$week <- week(as.Date.character(df$pubDate))

# agrupando sessões
df$sessao <- df$pubName
df$sessao[df$sessao %in% c('DO1A','DO1','DO1E')] <- 1
df$sessao[df$sessao %in% c('DO2','DO2E')] <- 2
df$sessao[df$sessao %in%c('DO3','DO3E')] <- 3

# removendo acentos e caracteres especiais por meio de conversão utf-8 para ascii
df$Texto <- iconv(df$Texto,from="UTF-8",to="ASCII//TRANSLIT")

# removendo diversos espaços
df$Texto <- gsub("\\s+", " ", df$Texto)
```

2.2 - Wordcloud

Serão apresentados tres exemplos de wordclouds, sendo o primeiro contendo dados de todas as sessão do dia 31-01-2019 plotado inteiramente em preto já o segundo grafico é colorido.

Sendo que para apresentar o wordcloud são necessarias mais algumas limpezas e transformações de dados.

Mineração de Texto e Limpeza de dados

```
#df1 <- df[df$sessao == 3,] # filtro por sessão
#df1 <- df[df1$week == 3,] # filtro por semana
df1 <- df[df$pubDate == "2019-01-31",] # filtro apenas dia 2019-01-31
# creating corpus
corpus <- Corpus(VectorSource(df1$Texto))
```

```
# Convert all text to lower case
corpus <- tm_map(corpus, content_transformer(tolower))
```

```
## Warning in tm_map.SimpleCorpus(corpus, content_transformer(tolower)):
## transformation drops documents
```

```
# remover toda a pontuação
corpus <- tm_map(corpus, removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(corpus, removePunctuation): transformation
## drops documents
```

```
# Remove numbers
corpus <- tm_map(corpus, removeNumbers)
```

```
## Warning in tm_map.SimpleCorpus(corpus, removeNumbers): transformation drops
## documents
```

```
# remover Stopwords
corpus <- tm_map(corpus, removeWords, stopwords('pt'))
```

```
## Warning in tm_map.SimpleCorpus(corpus, removeWords, stopwords("pt")):
## transformation drops documents
```

```
corpus <- tm_map(corpus, removeWords, c('serao', 'meses', 'hora', 'caixa', 'cep', 'mail', 'WWW', 'uso',
```

```
## Warning in tm_map.SimpleCorpus(corpus, removeWords, c("serao", "meses", :
## transformation drops documents
```

```
# generalizando termos para raiz
#corpus <- tm_map(corpus, stemDocument)
```

```
# Replacing "/", "@" and "|" with space
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
corpus <- tm_map(corpus, toSpace, "/")
```

```
## Warning in tm_map.SimpleCorpus(corpus, toSpace, "/"): transformation drops
## documents
```

```
corpus <- tm_map(corpus, toSpace, "@")
```

```
## Warning in tm_map.SimpleCorpus(corpus, toSpace, "@"): transformation drops
## documents
```

```
corpus <- tm_map(corpus, toSpace, "\\|")
```

```
## Warning in tm_map.SimpleCorpus(corpus, toSpace, "\\|"): transformation
## drops documents
```

```
td_mtx <- TermDocumentMatrix(corpus, control = list(minWordLength = 6))
```

```
v <- sort(rowSums(as.matrix(td_mtx)), decreasing=TRUE) #ordena as palavras
```

```
fdf <- data.frame(word=names(v), freq=v[]) #organiza um novo banco
```

Wordcloud Monocolor (Preto)

```
wordcloud(fdf$word, fdf$freq, min.freq=333)
```

Wordcloud Colorido

```
pal1 <- brewer.pal(12,"Paired")
pal3 <- brewer.pal(12,"Set3")

wordcloud(fdf$word, fdf$freq, min.freq=333, random.order=FALSE, rot.per=.15,colors=pal1)
```


termos
itens
presencial
fonte
prefeitura
registro
tendo
maria
decreto
desta
reais
interno
demais
entra
contratante
atribuicoes
gestao
instituto
social
presidente
eletronico
jose
servico
deste
municipal
fundamento
sera
quadro
recorrida
ministerio
uasg
ato
precos
vigencia
nacional
siape
contratada
emenda
constitucional
classe
oficial
ata
mil
edital
federal
saude
extrato
dou
inciso
portaria
tcu
matricula
chefe
municipio
uniao
trabalho
servicos
relator
eireli
processo
gov
empresa
item
torna
brasil
vista
valor
site
pregao
legal
secao
objeto
contrato
prazo
total
cargo
resolve
padrao
pessoal
codigo
licitacao
termo
acordao
artigo
aditivo
centro
autos
contratado
publico
substituto
assinatura
geral
nivel
publica
sicon
sede
publicada
secretaria
aviso
rio
unico
fazenda
partir
aposentadoria
aquisicao
conceder
ocupante
recursos
diario
comissao
conforme
recorrente
oliveira
paulo
publicacao
contratacao
resultado
global
estado