PEL218 - Atividade 1

Processamento de Linguagem Narual (Prof. Guilherme Wachs)

Claudio Aparecido Borges Junior (RA 120122-7)

Atividade

Escolher qualquer corpus (conjunto de documentos) ou livro de até 100 MB em português e extrair as seguintes informações:

1. Quantidade de palavras distintas;
2. Histograma das palavras;
3. Histograma de prefixos de tamanho (1,2,3,4 e 5) • Histograma de sufixos de tamanho (1,2,3,4 e 5)

Você poderá utilizar qualquer linguagem de programação (inclusive shell).

Resolução

Foi escolhida a linguagem *Python* versão 3.8.5. O conjunto de dados é o "Brazilian Portuguese Literature Corpus" com 3,7 milhões de palavras provindos da literatura Brasileira publicado entre 1840 e 1908. Tal conjunto de dados pode ser encontrado em: https://www.kaggle.com/rtatman/brazilian-portuguese-literature-corpus.

A resolução e a análise encontram-se abaixo.

In [3]:
```
import os
import re
from collections import Counter
from fnmatch import fnmatch
import matplotlib.pyplot as plt
```

Abaixo são definidas algumas variáveis. A codificação utilizada pelo corpus é a ISO-8859-1 (link), e os arquivos de interesse são terminados em *.txt*. O conteúdo do corpus é dividido em múltiplos arquivos, em diversas subpastas, portanto é necessário percorrer toda a árvore a partir da raiz.

In [4]:
```
root='./'
pattern='*.txt'
encode='ISO8859_1'

def list_all_files() -> list:
    # List all files that matches a pattern an return the file names
    # as a list of str
    corpora = []
    for path, subdirs, files in os.walk(root):
        for name in files:
            if fnmatch(name, pattern):
                corpora.append(os.path.join(path, name))
    return corpora

def read_file(path: str) -> str:
    # Read a single file and return its content as str
    with open(path, 'r', encoding=encode) as file:
        data = file.read()
    return data
```

Assim, pode-se ler todos os arquivos armazenando-os em memória já que o tamanho total é relativamente pequeno para um computador moderno

In [5]:
```
contents = [read_file(file) for file in list_all_files()]
corpus = ' '.join(contents)
```

Um pequeno ajuste no corpus é realizado para deixar todas as palavras com o mesmo tipo de caixa, desse modo pode-se identificar palavras idênticas sem a interferência da caixa

In [6]:
```
corpus_adj = corpus.lower()
```

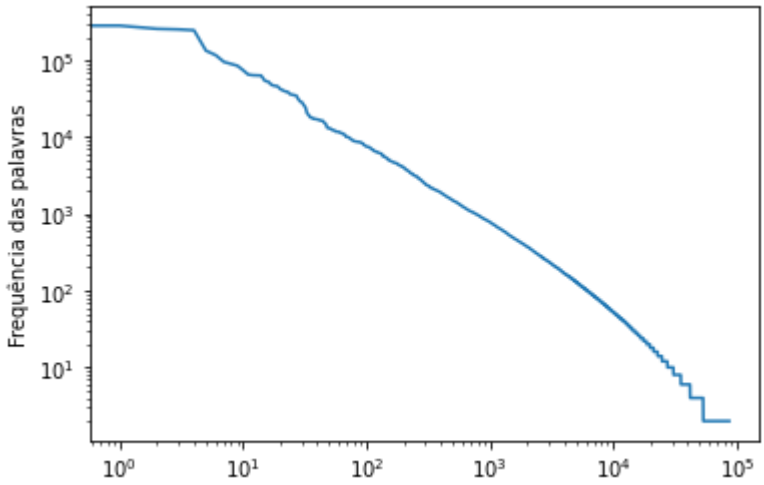
Um método simples de detectar palavras é a busca por letras acentuadas ou não em uma sequencia igual ou superior a 1 elemento. Tais palavras são então contabilizadas em um dicionário onde a chave é a própria palavra e o valor é o número de vezes que ela aparece

In [7]:
```
words = re.findall(r'[a-záàâãéèêíïóôõöúçñ]+', corpus_adj)
words_freq = Counter(words)
print('Palavras distintas identificadas: ' + str(len(words)))
```

```
Palavras distintas identificadas: 7617596
```

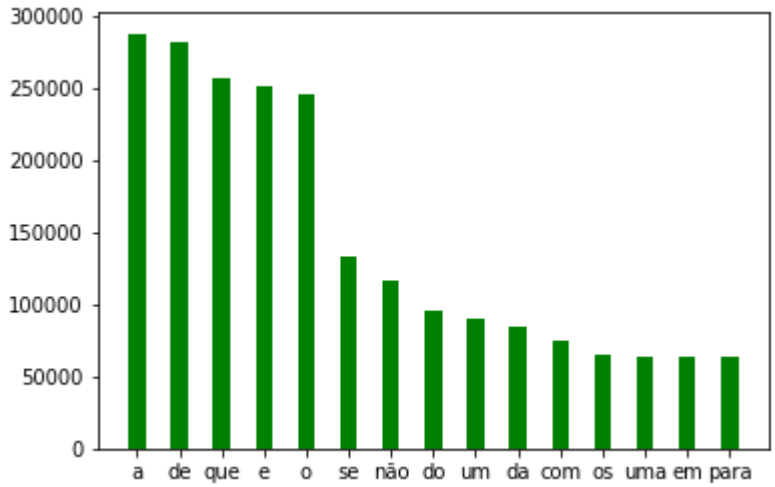
A primeira análise é a de frequência de palavras e seu decaimento. É possível reparar que que a frequência das palavras diminui de forma logarítmica. Tal efeito é semelhante a *Lei de Zipf (link), com ressalvas para as extremidades. Essa lei estabelece que a primeira palavra mais comum é usada exatamente duas vezes mais que a segunda, e a segunda, duas vezes mais que a terceira, e assim por diante. Esse efeito é verdadeiro para outras linguagens além do português

In [8]:
```
freq = sorted(list(words_freq.values()), reverse=True)
plt.plot(freq)
plt.ylabel('Frequência das palavras')
plt.yscale('log')
plt.xscale('log')
plt.show()
```



As palavras mais frequentes são artigos, preposições e conjunções, sendo que, nesse corpus, o artigo feminino possui maior frequência do que o artigo masculino. Porém, ambos parecem estar no mesmo nível de utilização, assim como as palavras um e uma, indicando que o gênero não possui relevância na frequência de utilização.

In [9]:
```
top = words_freq.most_common(15)
top_keys = [elm[0] for elm in top]
top_vals = [elm[1] for elm in top]
plt.bar(top_keys, top_vals, 0.4, color='g')
plt.show()
```



Os prefixos e sufixos são analisados sendo eles determinados em palavras de tamanho superior ao conjunto de elementos necessários, sendo assim, uma palavra de tamanho 4, somente possui sufixo ou prefixo de tamanho 1, 2 e 3.

In [10]:
```
def prefix_counter(words: list, n: int) -> Counter:
    prefix = [word[:n] for word in words if len(word) > n]
    return Counter(prefix)

def sufix_counter(words: list, n: int) -> Counter:
    sufix = [word[-n:] for word in words if len(word) > n]
    return Counter(sufix)
```

In [11]:
```
parts = [1, 2, 3, 4, 5,]
prefixes = {part: prefix_counter(words, part) for part in parts}
sufixes = {part: sufix_counter(words, part) for part in parts}
```

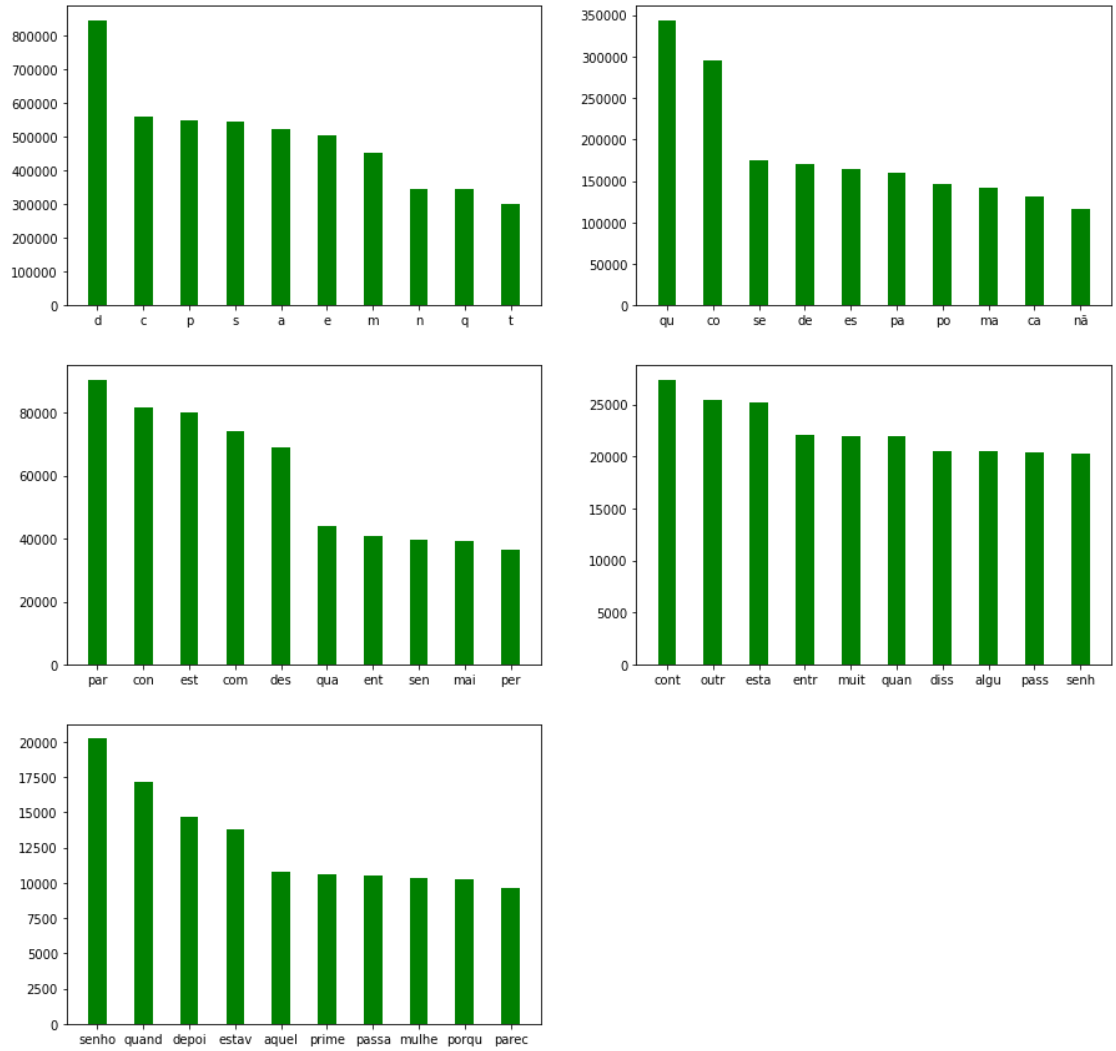
Abaixo são apresentados 5 gráficos de prefixos com tamanhos 1, 2, 3, 4 e 5 sucessivamente. O prefixo de temanho "d" é o mais utilizado, seguido pelo "c", "p" e "d".

O prefixo de tamanho 2 mais utilizado é o "qu" (queda, querida e queijo), seguido pelo "co" (começo, coronel e covarde) e depois o "se" (semana, sentimento e sentinela).

O sufixo de tamanho 4 mais utilizado é o "cont", que faz parte de diveras palavras como: "contra", continuou e contrário. O segundo sufixo mais utilizado é o "outr", que é prefixo de outro e suas diversas formas gramaticais.

O prefixo de tamanho 5 mais utilizado é "senh", de senhor, senhora, senhoril entre outros. Mesmo ele sendo um substantivo, essa palabra é muito utilizado para substituir o pronome "você". O segundo sufixo mais utilizadoé "quand", de "quando", sendo utilizado como advérbio, conjunção e pronome relativo.

In [18]:
```
fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(15,15))
plt.delaxes(axs[2,1])
for i in range(len(parts)):
    top = prefixes[parts[i]].most_common(10)
    top_keys = [elm[0] for elm in top]
    top_vals = [elm[1] for elm in top]
    axs[int(i / 2), int(i % 2)].bar(top_keys, top_vals, 0.4, color='g'
plt.show()
```

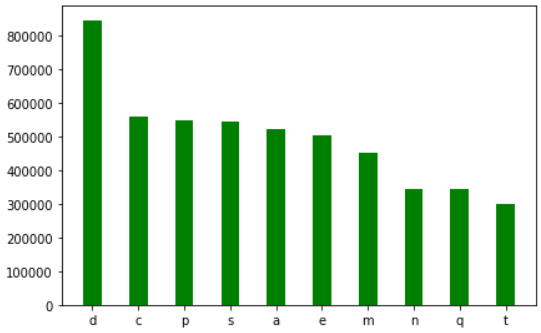


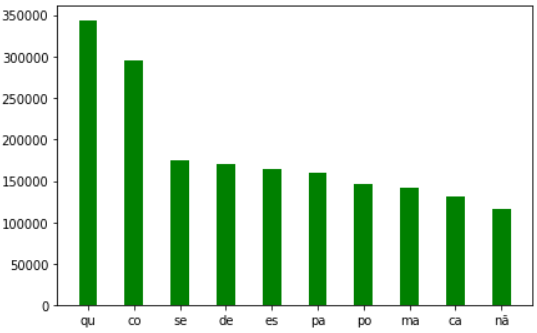
Abaixo são apresentados 5 gráficos de sufixos com tamanhos 1, 2, 3, 4 e 5 sucessivamente. Os artigos feminino e masculino "a" e "o" são os sufixos de tamanho 1 mais comuns, indicando sua utilização como artigos e gênero de palavras.

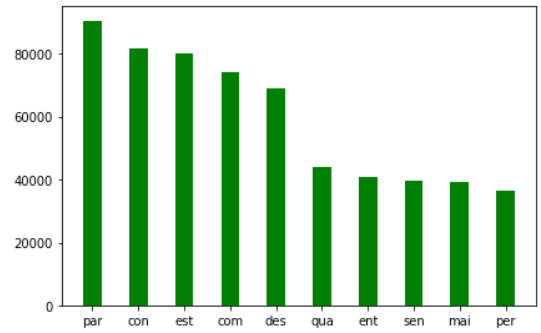
O sufixo de tamanho 4 mais utilizado é o "ando", presente em palabras como, cantando, pulando, andando, sendo essas uma conjugação verbal do português.

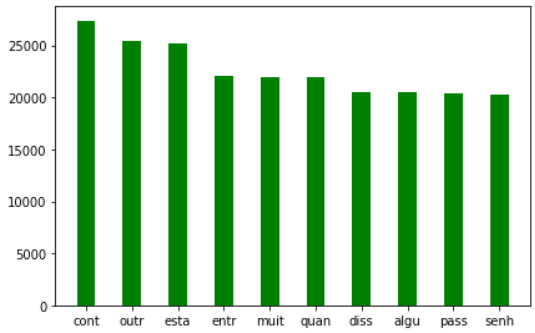
O sufixo de tamanho 5 mais utilizado é o "mente", de popularmente, visivelmente e aleatoriamente.

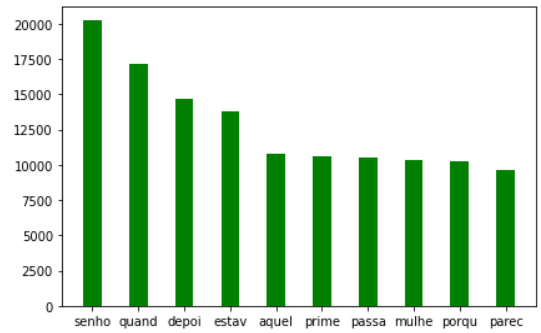
In [19]:
```
fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(15,15))
plt.delaxes(axs[2,1])
for i in range(len(parts)):
    top = prefixes[parts[i]].most_common(10)
    top_keys = [elm[0] for elm in top]
    top_vals = [elm[1] for elm in top]
    axs[int(i / 2), int(i % 2)].bar(top_keys, top_vals, 0.4, color='g'
plt.show()
```


800000
700000
600000
500000
400000
300000
200000
100000
0
d
c
p
s
a
e
m
n
q
t


350000
300000
250000
200000
150000
100000
50000
0
qu
co
se
de
es
pa
po
ma
ca
nã


80000
60000
40000
20000
0
par
con
est
com
des
qua
ent
sen
mai
per


25000
20000
15000
10000
5000
0
cont
outr
esta
entr
muit
quan
diss
algu
pass
senh


20000
17500
15000
12500
10000
7500
5000
2500
0
senho
quand
depoi
estav
aquel
prime
passa
mulhe
porqu
parec