Processamento de Linguagem Natural

Aula 2: frequência de termos

Objetivo: contar a frequência de termos em um texto para evidenciar seu conteúdo

|  |
| --- |
| **Exercício 1** |
| *Objetivo: usar a urllib para baixar um texto* |
| O código abaixo faz o *download* do texto de “Os Lusíadas” e imprime um pedaço dele na tela:  import urllib  url = "https://www.gutenberg.org/cache/epub/3333/pg3333.txt" # Os Lusíadas (Camões)  texto = urllib.request.urlopen(url).read()  print(texto[1000:2000])  Quando o código é executado, o texto é impresso na tela corretamente? Em que situações há problemas? Por que esses problemas acontecem? |

|  |
| --- |
| **Exercício 2** |
| *Objetivo: carregar um texto com encoding UTF-8* |
| Compare o código abaixo com o do exercício 2. Qual é a diferença do código? Qual é sua saída?  import urllib  url = "https://www.gutenberg.org/cache/epub/3333/pg3333.txt" # Os Lusíadas (Camões)  texto = urllib.request.urlopen(url).read().decode()  print(texto[1000:2000]) |

|  |
| --- |
| **Exercício 3** |
| *Objetivo: dividir um texto usando re.split()* |
| Uma das maneiras de identificar palavras em uma string é separando a string quando encontramos espaços. Para tal, podemos usar a construção:  import re  s = re.split("\s+", texto)  print(s)  Execute esta construção. Analisando os resultados gerados, verifique se ela permite encontrar todas as palavras de “Os Lusíadas” e, simultaneamente, excluir elementos que não são palavras. |

|  |
| --- |
| **Exercício 4** |
| *Objetivo: entender como funciona a divisão de textos usando re.findall()* |
| Uma outra maneira de dividir uma string é encontrando todos os elementos que batem com uma expressão regular. Podemos, por exemplo, usar a seguinte construção:  import re  s = re.findall("[^\s]+", texto)  print(s)  Execute esta construção. Compare os resultados gerados com os presentes no exercício 3. Quais elementos estão incorretamente divididos? |

|  |
| --- |
| **Exercício 5** |
| *Objetivo: dividir um texto usando re.findall()* |
| Encontre uma expressão regular capaz de dividir um texto escrito em português em suas palavras isoladas, excluindo as pontuações. A expressão deve funcionar bem, ao menos, para o texto abaixo:  texto = "Postule-se, têmpera doze átomo, diga-me o que há. 95? Doze!" |

|  |
| --- |
| **Exercício 6** |
| *Objetivo: contar palavras e entender a diferença entre total de palavras e total de palavras únicas* |
| (a) Após executar-mos a chamada s = re.findall(...) com a expressão regular que você encontrou no exercício 5, qual é a relação entre len(s) e o número de palavras do texto?  (b) Qual característica do texto é calculada por len(set(s))? Faça o teste desta propriedade na *string* de teste do Exercício 5. Existe algum bug adicional que gostaríamos de resolver?  (c) Calcule a *densidade de vocabulário*, isto é, a razão entre o número de palavras únicas e o número total de palavras de “Os Lusíadas”. |

|  |
| --- |
| **Exercício 7** |
| *Objetivo: contar a frequência de termos em um texto* |
| Já sabemos qual é o total de palavras em um texto. Também, sabemos quantas palavras diferentes existem em um texto. Agora, contaremos quantas ocorrências de cada palavra única existem no texto.  Das duas propostas de algoritmos abaixo, qual deve executar mais rapidamente? Por que?  def contar1(s):      ja\_contei = []      contagem = []      for w in s:          idx = ja\_contei.index(w)          if idx >= 0:              contagem[idx] += 1          else:              ja\_contei.append(w)              contagem.append(1)      return ja\_contei, contagem  def contar2(s):      contagem = dict()      for w in s:          if w in contagem.keys():              contagem[w] += 1          else:              contagem[w] = 1      return contagem |

|  |
| --- |
| **Exercício 8** |
| *Objetivo: encontrar a frequência de palavras em “Os Lusíadas”* |
| (a) Quais são as 20 palavras mais frequentes em “Os Lusíadas”? Qual é a frequência delas?  (dica: use o código abaixo para ordenar palavras de acordo com suas contagens):  tf = []  for i in contagem.keys():      tf.append( (contagem[i],i) )  tf = sorted(tf, reverse=True)  print(tf[0:20])  (b) Essas palavras têm significado em relação ao sentido do texto? |

|  |
| --- |
| **Exercício 9** |
| *Objetivo: encontrar a frequência de palavras em “Os Lusíadas”* |
| Encontre na Internet uma lista de *stopwords* em português. Remova as *stopwords* de sua lista de palavras e calcule a frequência de palavras novamente. Já é possível inferir algo sobre o conteúdo do texto a partir dessas palavras? |