Processamento de Linguagem Natural

Luís Filipe da Costa Cunha
lfilipecc1@gmail.com

José João Almeida
jj@di.uminho.pt

Scripting no Processamento de Linguagem Natural

Utilização de linguagens de scripting para o desenvolvimento e integração de ferramentas de PLN.

Plano

- Introdução ao Python
- Unix Filters
- Expressões Regulares
- Precision Recall F1-score
- Corpora
- Gramática do Português
- Terminologia Dicionários e enciclopédias
- Word Embeddings entre outros

Aulas

Sistema Operativo:

- Unix
- Python > 3.6

GitHub:

- https://github.com/lfcc1/plneb
- Bibliografia
- Aulas
- Data
- Slides
- Tpcs

Avaliação

- 2 ou 3 trabalhos práticos
- 1 teste escrito
- TPCs

Introdução ao Python

```
#!/usr/bin/env python3

print("Hello World")

editor = input("What is your favorite text editor? ")

a = [3, 6, 7, 2, 5, 9, 4, 0, 10]

for i in a:
  if i < 5:
    print(str(i) + " is lower than 5")
  elif i > 5:
    print(str(i) + " is higher than 5")
  else:
    print("FIVE!")

# comment
```

- Listas
- Dicionários
- Ficheiros
- Exercícios

Listas

```
my_list = [1,2,3,4,"Filipe", 1.232]

my_list.append(5)
my_list.append([6,7])
my_list.extend([6,7,"Alexandra"])

res = [1,2,3] + [4,5,6]
my_list[4] = 'João'
my_list.insert(4,"Pedro")
my_list[::-1]
```

```
[1, 2, 3, 4, 'Filipe', 1.232, 5]
[1, 2, 3, 4, 'Filipe', 1.232, 5, [6, 7]]
[1, 2, 3, 4, 'Filipe', 1.232, 5, [6, 7], 6, 7, 'Alexandra']
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 4, 'João', 1.232, 5, [6, 7], 6, 7, 'Alexandra']
[1, 2, 3, 4, 'Pedro', 'João', 1.232, 5, [6, 7], 6, 7, 'Alexandra']
['Alexandra', 7, 6, [6, 7], 5, 1.232, 'João', 'Pedro', 4, 3, 2, 1]
```

Listas

```
my_list = list(range(10)) + ['exemplo','exemplo','exemplo','exemplo']
my_list.pop(3) #index
my_list.pop(-3)
my_list.remove(5)
my_list.remove('exemplo') #primeira ocorrencia
my_list = list(filter(lambda el : el != 'exemplo' ,my_list))
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 'exemplo', 'exemplo', 'exemplo', 'exemplo']
[0, 1, 2, 4, 5, 6, 7, 8, 9, 'exemplo', 'exemplo', 'exemplo', 'exemplo']
[0, 1, 2, 4, 5, 6, 7, 8, 9, 'exemplo', 'exemplo', 'exemplo']
[0, 1, 2, 4, 6, 7, 8, 9, 'exemplo', 'exemplo', 'exemplo']
[0, 1, 2, 4, 6, 7, 8, 9, 'exemplo', 'exemplo']
[0, 1, 2, 4, 6, 7, 8, 9]
```

Dicionários

```
dict = {}
colors = {'fcp': 'blue', 'slb': 'red', 'scp': 'green'}
len(colors)
'slb' in colors
colors['aca'] # KeyError: 'aca'
colors.get('fcp', 'Não existe')
colors.get('aca', 'Não existe')

colors.keys()
colors.values()
colors.items()
```

```
3
True
blue
Não existe
['fcp', 'slb', 'scp']
['blue', 'red', 'green']
[('fcp', 'blue'), ('slb', 'red'), ('scp', 'green')]
```

Listas por compreensão

```
my_list = list(range(1,6))
my_list = [ x * 10 for x in my_list]

for i in range(len(my_list)):
    print(my_list[i])

for index, elem in enumerate(my_list):
    print(index, ":", elem, end=' ')
```

```
10
20
30
40
50
0 : 10 1 : 20 2 : 30 3 : 40 4 : 50
```

Ordenação

```
b = {"a":{"x":2, "y":8}, "b": {"x":3, "y":7}, "c":{"x":2, "y":9}}

print(sorted(b.items(), key=lambda el:el[1]["x"], reverse=True))

print(sorted(b.items(), key=lambda el: (el[1]['x'], -el[1]['y'])))
```

```
[('b', {'x': 3, 'y': 7}), ('a', {'x': 2, 'y': 8}), ('c', {'x': 2, 'y': 9})]
[('c', {'x': 2, 'y': 9}), ('a', {'x': 2, 'y': 8}), ('b', {'x': 3, 'y': 7})]
```

Ficheiros

```
file = open('data/exemplo', "r")
lines = file.readlines()

file1 = open('data/exemplo', "r")
string = file1.read()

file2 = open('data/exemplo', "r")
line = file2.readline(10)
```

```
['Ola! Bem vindo a Scripting no Processamento de Linguagem Natural \n', 'Este
ficheiro é apenas um exemplo.\n', 'Boa sorte!']
---
Ola! Bem vindo a Scripting no Processamento de Linguagem Natural
Este ficheiro é apenas um exemplo.
Boa sorte!
---
Ola! Bem v
```

Exercícios

1. Programa que pergunta ao utilizador o nome e imprime em maiúsculas.
2. Função que recebe array de números e imprime números pares.
3. Função que recebe nome de ficheiro e imprime linhas do ficheiro em ordem inversa.
4. Função que recebe nome de ficheiro e imprime número de ocorrências das 10 palavras mais frequentes no ficheiro.
5. Função que recebe um texto como argumento e o ”limpa”: separa palavras e pontuação com espaços, converte para minúsculas, remove acentuação de caracteres, etc.

Processamento de Linguagem Natural

Luís Filipe da Costa Cunha
lfilipecc1@gmail.com

José João Almeida
jj@di.uminho.pt