Processamento de Linguagem Natural

Luís Filipe da Costa Cunha
lfilipecc1@gmail.com

José João Almeida
jj@di.uminho.pt

Expressões Regulares

“Regular expressions are extremely useful in extracting information from text such as code, log files, spreadsheets, or even documents.”

Expressões Regulares

```
import re

re.search(r'João',"o João não gosta de andar de comboio")

re.search(r'pytho.','adoro programar em python')

re.search(r'[0-9]{4}','2022 vai ser um bom ano!!')
```

Quantificação

```
import re

re.findall(r'colou?r+','Is this a color or colour?')     #['color', 'colour']


re.findall(r'[0-9]+','O semestre teve inicio no dia 1 de Fevereiro de 2020')    #['1', '2020']


re.findall(r'Java[a-zA-Z]*','Javascript is not Java')    #['Javascript', 'Java']


re.findall(r'\b[a-z]{1,6}\b',"ola! Apenas quero palavras pequenas" ) #['ola', 'quero']
```

Grouping

```
import re

re.search(r'alde(ão|ãe|õe)s','Os aldeãos fizeram uma festa na aldeia' )

re.findall(r'(Sra|Sr|Senhora|Senhor)','O Senhora Teresa encontrou a Sra. Maria no
shopping') ['Senhora', 'Sra']
```

Disjunction and Intervals

```
[AEIOU]
[12345678]
alun[oa]
[A-Z]
[a-z]
[0-9]
[a-zA-Z0-9]
[^aeiou]
```

Character Classes

- `\d Digit  ([0-9])`
- `\D not \d`
- `\w letter digit or underscore ([a-zA-Z0-9_]`
- `\W not \w`
- `\s whitespace`
- `\S not whitespace`

Anchors

`^`	`beginning of line`
`$`	`end of the line`
`\b`	`word boundary`

Capture Groups

```
re.findall(r'O ([A-Z][a-z]+) tem ([\w ]+)', "O Manuel tem uma mochila.")
#[('Manuel', 'uma mochila)]
```

```
re.findall(r'O ([A-Z][a-z]+) tem (:?[\w ]+)', "O Manuel tem uma mochila.")
#['Manuel']
```

Regex Functions

- match - Try to apply the pattern at the start of the string
- search - Scan through string looking for a match to the pattern
- findall - Return a list of all non-overlapping matches in the string
- sub - Replace occurrences of the regex pattern

Exercicios

Define regular expressions to match strings that:

1. have a 't'
2. have a 't' or a 'T'
3. have a letter (and how many)
4. have a digit
5. have a decimal number
6. have a length higher than 3 characters
7. have an 'M' but not an 'm'
8. have a character repeated twice

Exercicios

9. Have only one character repeated many times
10. put all words between {}

Processamento de Linguagem Natural

Luís Filipe da Costa Cunha
lfilipecc1@gmail.com

José João Almeida
jj@di.uminho.pt