

Universidade do Minho
Mestrado Integrado em Engenharia Informática
Scripting no Processamento de Linguagem Natural
Relatório do Trabalho Prático

João Cunha (A84775)
Luis Ramos (A83930)

27 de junho de 2021

Conteúdo

1	Introdução	3
1.1	Enquadramento e Contextualização	3
1.2	Problema e Objetivos	3
2	Conceção/desenho da Resolução	4
2.1	Extrator	4
2.2	Flask	5
3	Modo de Execução	7
3.1	Comandos	7
3.1.1	Script	7
3.1.2	Aplicação	7
4	Conclusão	8

Lista de Figuras

1	Extração da página e blocos de texto	4
2	Limpeza e obtenção do json	4
3	Escrita do ficheiro json	4
4	Validação do json	5
5	Interface gráfica do programa em flask	5
6	Opção de fazer download ao zip com os jsons	6

1 Introdução

1.1 Enquadramento e Contextualização

O presente relatório desenvolve-se no âmbito da Unidade Curricular de Scripting no Processamento de Linguagem Natural, do 4º ano do Mestrado Integrado em Engenharia Informática, tendo como principal objectivo o desenvolvimento de um extrator de bloco json em páginas web.

1.2 Problema e Objetivos

Os principais objetivos deste trabalho são os seguintes:

- Dado um link e uma palavra, extrair todos os blocos de json que contêm a palavra;
- Criar uma interface para ser mais fácil e intuitivo ao utilizador de recorrer ao extrator;
- Permitir ao utilizador fazer o download dos blocos de json extraídos.

Cumpridos todos os objetivos, foram ainda adicionados uns objetivos extras pelos próprios elementos do grupo.

2 Conceção/desenho da Resolução

2.1 Extrator

Começou-se por recorrer ao package **requests** de modo obter o html do link fornecido pelo utilizador.

Tendo o html, o objetivo passou por conseguir ir buscar ao html todos os blocos de texto que incluíssem a palavra dada pelo utilizador.

```
def getJson(link,name):  
    bool = True  
    pag = requests.get(link)  
    soup = BeautifulSoup(pag.text, 'html.parser')  
    matches = soup.find_all(string=re.compile(name))
```

Figura 1: Extração da página e blocos de texto

Para cada um desses blocos, foi necessário verificar se estes continham um json com a palavra incluída nele. Isso foi obtido através de alguma limpeza do bloco de texto e de um simples regex para verificar se era formato json.

```
for match in matches:  
    match = re.sub(r"}\n", "};\n", match)  
    match = re.sub(r"(?<=[^;])\n", " ", match)  
  
    list = re.findall(rf'(?:(const|var)\s+([\w]*)\s+?\s+=\s+(.*{name}.*)\s+;', match)
```

Figura 2: Limpeza e obtenção do json

Para cada bloco de json encontrado, criou-se o ficheiro com o nome correspondente ao dado ao json em que a informação seria o próprio json.

```
for (name,content) in list:  
    t = open('JsonsTemp/'+name+'.json', "w")  
    json_prettify = validateJSON(content)  
    t.write(json_prettify)  
    t.close()
```

Figura 3: Escrita do ficheiro json

De notar que ainda se realizou uma validação do json, em que, caso este esteja bem construído, então é colocado no ficheiro de forma indentada, caso contrário é apenas despejado o conteúdo para dentro do ficheiro.

```
def validateJSON(jsonData):  
    try:  
        json_object = json.loads(jsonData)  
        json_formatted_str = json.dumps(json_object, indent=2)  
    except ValueError as err:  
        return jsonData  
    return json_formatted_str
```

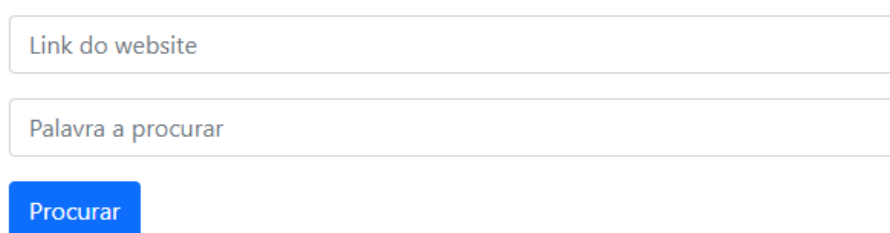
Figura 4: Validação do json

2.2 Flask

Após a conclusão do objetivo principal, que era o extrator de blocos json, o grupo decidiu adicionar ao projeto uma interface gráfica de forma a facilitar a utilização do programa, sendo esta feita em flask.

Foi feita um pequeno form para o utilizador introduzir o link do website pretendido e a palavra a procurar nos jsons.

Extrator Json



Link do website

Palavra a procurar

Procurar

Figura 5: Interface gráfica do programa em flask

Se forem encontrados jsons no website, é dada a opção de fazer o download com um zip contendo os jsons.

Extrator Json

Foram encontrados jsons!

<https://ge.globo.com/futebol/futebol-internacional/futebol-portugues/>

Braga

Procurar

Download

Figura 6: Opção de fazer download ao zip com os jsons

3 Modo de Execução

3.1 Comandos

3.1.1 Script

Para executar o script é necessário adicionar o link e a palavra como input, ou seja, o comando de execução terá o seguinte formato:

```
python script.py link palavra
```

Daremos agora um exemplo específico onde o link em que o utilizador deseja procurar é "https://www.flashscore.pt/" e a palavra que ele quer encontrar é "Portugal":

```
python script.py https://www.flashscore.pt/ Portugal
```

3.1.2 Aplicação

Para executar a aplicação em flask é apenas necessário correr o comando:

```
python app.py
```

4 Conclusão

Culminada a elaboração deste trabalho prático, importa referir que a execução do mesmo permitiu aos elementos do grupo expandir o conhecimento da linguagem python.

A execução deste trabalho prático permitiu uma melhor consolidação dos construtos teóricos e práticos através da utilização de packages de python, como por exemplo, requests, re e flask.

No ímpeto geral o desenvolvimento deste trabalho prático decorreu como planeado, alcançando todos os objectivos delineados.