Web Scraping + TF-IDF

José Carvalho
(a80424)

June 2020

Resumo

O projeto no qual este relatório se baseia tem como intuito realizar um Web Scraper para extrair informação de artigos de um website, aplicando após isto o algoritmo TF-IDF para ver quais são os artigos em que certas palavras têm mais peso. Este projeto foi realizado no constexto da Unidade Curricular de Scripting no Processamento de Linguagem Natural.

Conteúdo

1 Introdução

O projeto no qual este relatório se baseia tem como intuito realizar um Web Scraper para extrair informação de artigos de um website, aplicando após isto o algoritmo TF-IDF para ver quais são os artigos em que certas palavras têm mais peso. Este projeto foi realizado no constexto da Unidade Curricular de Scripting no Processamento de Linguagem Natural.

1.1 Web Scraper

Para realizar o Web Scraper, foi utilizada uma biblioteca de python denominada de Beautiful Soup. O website escolhido para realizar o Web Scraper foi o do jornal online ABola. A informação que foi recolhida dos artigos desse jornal foi o título, o texto e o link para a imagem principal dos artigos.

1.2 TF-IDF

Após a extração de toda a informação de uma porção considerável dos artigos do website, foi aplicado o algoritmo TF-IDF, com o propósito de perceber quais os artigos em que um conjunto de palavras, (selecionadas pelo utilizador) têm mais importância. Nesta aplicação do algoritmo foram atribuídos pesos distintos no aparecimento das palavras em secções diferentes do artigo.

1.3 Estrutura do Relatório

Este relatório é composto por uma introdução para dar ao leitor uma ideia básica do que irá ser discutido ao longo deste.

De seguida, é efetuada uma descrição informal do problema sobre o que este projeto se irá debruçar, numa tentativa de o resolver. Os requisitos que esta resolução necessita de cumprir estão explicitados a seguir, para ajudar a perceber quais os objetivos que terão de ter sido alcançados na conclusão do projeto.

Na terceira secção é exposta a conceção da resolução de forma sucinta e explícita, dando ênfase às etapas de obtenção de informação do website escolhido para esta poder ser utilizada no trabalho, à estruturação e interligação entre os conjuntos de páginas criadas, à elaboração de uma interface no terminal que seja de fácil utilização e à aplicação do algoritmo TF-IDF à informação obtida a partir dos artigos do website.

Na penúltima etapa é discutido o código escrito, com base nas elaboração das etapas referidas na secção anterior, e os testes que foram realizados para verificar a veracidade do código e do processo em si realizado.

Por fim é feita uma conclusão na qual é realizada uma análise crítica do trabalho realizado dando, indicações sobre o trabalho que poderá ser realizado futuramente, e sobre o que poderia ter sido realizado mas não foi conseguido.

2 Análise e Especificação

2.1 Descrição informal do problema

Foi apresentado aos alunos da Unidade Curricular de Scripting no Processamento de Linguagem Natural, uma lista de possíveis projetos para realizarem de forma individual. O projeto escolhido foi o projeto número 3, denominado de "Web scraping + TF-IDF". Este projeto exige que se extraia e armazene o conteúdo de um website à escolha do aluno, sendo que depois é aplicado a este conteúdo o algoritmo de TF-IDF, para auxiliar na pesquisa textual de artigos através de palavras chave.

2.2 Especificação dos Requisitos

Foram colocados no enuncido do trabalho alguns requisitos necessários para a sua boa resolução. Estes requisitos são:

- Conceber um Web Scraper para extrair a informação de artigos de um website escolhido pelo aluno e armazená-la localmente.
- Criar uma interface CLI ou Web que permita uma pesquisa textual nos artigos. Esta terá de devolver uma lista ordenada, por ordem decrescente, dos artigos em que as palavras pesquisadas tenham tido mais peso.
- Atribuir pesos diferentes consoante a secção do artigo. Ex: uma palavra no título de um artigo tem mais relevância do que a mesma palavra no texto de um artigo.

3 Conceção/Desenho da Resolução

3.1 Obtenção e estruturação da informação do website

3.1.1 Obtenção de informação

O primeiro passo que foi necessário tomar foi o de obter a informação do website. Para este passo foi utilizada a biblioteca Beautiful Soup do python. Esta biblioteca permite a extração de informação de ficheiros HTML, sendo por isso um instrumento fulcral na elaboração deste projeto.

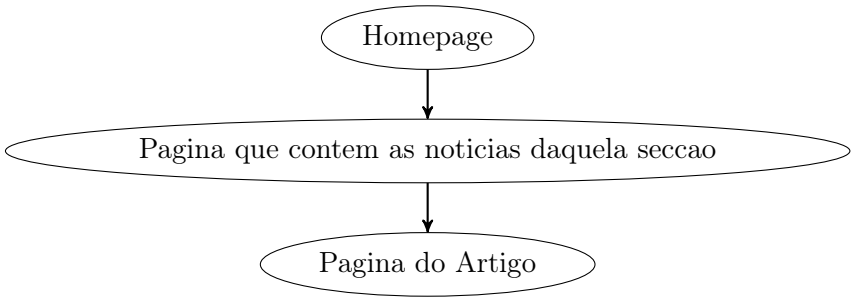
Existem várias funções dentro do ficheiro web_scraper.py, (ficheiro que realiza a web scraping) que são importantes nesta etapa do projeto. Algumas são:

- mainPage - Função que não só vai buscar informação e cria as páginas dos artigos que se encontram numa secção da página principal do website, como também preenche a página dessa secção com links para cada uma das páginas dos artigos criadas.
- getClubSoup - Função que vai buscar informação de todas as páginas de todos os clubes que foram encontradas no website.
- clubeCont - Função que faz o mesmo que a mainPage, mas para artigos que existem nas páginas de um clube.

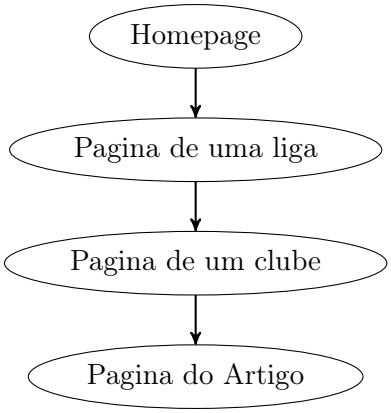
Após a extração da informação ter sido concluída é necessário perceber como armazená-la.

3.1.2 Estruturação

Após a obtenção da informação dos artigos, foi necessário estruturar as páginas constituídas inteiramente por links, as páginas armazenam a informação retirada dos artigos e principalmente as ligações entre estas, aquando da obtenção da informação do website, para o acesso à informação ser facilitado. A estruturação concebida varia, consoante o artigo esteja presente na página principal, ou esteja presente na página de um dos clubes. Caso a notícia esteja na página principal:



Caso a notícia exista na página de um clube:



De seguida, foi necessário dividir a informação que se iria extrair em duas diretorias: 'páginas' e 'artigos'. A informação estrutural, (links, nomes de secções, nomes de ligas,...) seria extraída para a diretoria 'paginas'. A informação contida dentro de cada artigo seria guardada dentro de um ficheiro

html localizado na diretoria 'artigos'. A principal razão para esta divisão foi o facto da pesquisa textual só ser relativa aos artigos, não querendo por conseguinte que as páginas que não contém informação relativa aos artigos do website não sejam pesquisadas.

Quando se corre pela primeira vez o ficheiro relativo ao web scraper este vai buscar a informação relativa aos artigos do website, por intermédio das funções já referidas no início desta subsecção. Este processo demora uma quantidade de tempo ainda relativamente grande, não só devido devido à informação que vai buscar, mas devido à sua estruturação, exemplificada nos grafos existentes nesta subsecção.

3.2 Elaboração da interface

A interface elaborada foi uma command line interface. Para correr esta interface basta executar o ficheiro menu.py. O utilizador é recebido com as seguintes alternativas:

```
1 - Consultar a informação do site do jornal ABola
2 - Aplicar o algoritmo do TFIDF
3 - Sair
```

Caso o utilizador escolha a opção número 1, existem duas situações distintas que podem acontecer:

- Caso esta seja a primeira vez que a aplicação é corrida, o utilizador terá que aguardar pela obtenção e estruturação referida na subsecção anterior.
- Caso esta não seja a primeira vez que a aplicação é corrida, a homepage é aberta e o utilizador pode consultar toda a informação armazenada.

Caso o utilizador escolha a opção número 2, é-lhe pedido que ele escolha quantas palavras quer procurar. De seguida, o utilizador digita esse numero de palavras no terminal. Quando o utilizador acaba de digitar a ultima palavra, o programa aplica a procura textual com TF-IDF aos artigos, com o resultado a aparecer por ordem decrescente num ficheiro, que abre de forma imediata.

Caso o utilizador escolha a opção número 3, a execução do ficheiro é terminada.

3.3 Aplicação do algoritmo TF-IDF

O ficheiro tfidf.py contém a classe TF-IDF que contém funcções muito importantes para a execução do algoritmo. Algumas destas são:

- addDocument - Função que pega num artigo que recebe e acrescenta-o a um array, em que cada elemento é um array de tamanho 2 composto pelo nome do artigo e por um dicionário em que a chave são as palavras existentes no artigo e o score respetivo. O array de arrays fica depois armazenado na variável 'documents' da classe.

- similarities - Função que recebe e efetua a pesquisa textual de uma palavra nao conjunto de arrays armazenados, retornando os resultados de forma descendente consoante a relevância das palavras nos artigos.

3.3.1 Atribuição dos pesos

Os títulos e os textos dos artigos foram separados, com intuito de atribuir pesos diferentes a cada um destes. O algoritmo de TF-IDF com pesos indica que a soma dos pesos das diferentes secções necessita de ser 1, logo visto que os artigos só possuem duas secções diferentes (título e texto), foi atribuído o peso de 0.7 aos títulos e de 0.3 ao texto.

4 Codificação e Testes

4.1 Alternativas, Decisões e Problemas de Implementação

Durante a realização do projeto existiram algumas problemas com os quais me deparei:

- havia algumas secções da página principal, nomeadamente a secção dos jogos, que não possuiam artigos mas sim jogos fornecidos pelo website.
- existiam pedaços de código que o Beautiful Soup não conseguia resolver, nomeadamente o bocado de código relativo as links que interligavam as páginas dos clubes de cada liga, com a página principal do website.
- A inexistência de conhecimento acerca do algoritmo de TF-IDF.

Estas foram algumas das soluções encontradas:

- Relativo à secção dos jogos, na página dos jogos se se clicar num dos links o utilizador é redirecionado para a página do jogo no website.
- Relativamente à interligação das páginas dos clubes com a página principal, só foi possível obter informação dos artigos dos clubes através dos paths deles. Cada clube tem um número respetivo, logo foi necessário recolher os números de todos os clubes. Alguns dos clubes não têm uma página própria logo não foi possível recolher notícias deles.
- Quanto ao algoritmo TF-IDF, ao fim de algum tempo foi possível implementá-lo através de artigos e pesquisas efetuadas.

4.2 Testes realizados e Resultados

4.2.1 Estrutura

A organização da página de um artigo é composta por um título, o link da imagem principal do artigo, e o texto do artigo. Eis como a página da artigo estaria estruturada:

100 golos e um feito único para Keisuke Honda

Atualmente no Botafogo, Keisuke Honda marcou o 100.º golo da carreira no empate com o Bangu (1-1). A cumprir a primeira temporada no futebol brasileiro, o japonês conseguiu também um feito único: tornou-se no único jogador a marcar golos nos seis continentes.

Formado no Nagoya Grampus, marcou os primeiros golos enquanto profissional no Japão (13 ao todos). Em 2007 muda-se então para o continente europeu, onde celebra tentos ao serviço de VVV-Venlo (26), CSKA Moscovo (27) e Milan (11).

Em 2013 estreia-se no continente norte-americano, e acaba por fazer 13 golos ao serviço do Pachuca (México). Segue-se uma curta passagem pelo Melbourne Victory, da Austrália, onde celebra 8 tentos.

Pelo meio, apontou dois golos no continente africano, ao serviço da seleção japonesa no Mundial-2010, na África do Sul. Os nipónicos foram eliminados nos oitavos de final pelo Paraguai (0-0, 3-5 gp),

```
mas Honda marcou nas vitórias com Camarões (1-0) e Dinamarca (3-1),
na fase de grupos.

É no jogo de estreia no futebol brasileiro que acaba por se estrear
a marcar na América do Sul e completa o périplo de continentes com
33 anos.
```

4.2.2 Testando o TFIDF:

Aplicando o algoritmo de TFIDF e pesquisando pelas palvras:

```
Porto
dois
João
```

Obtem-se o seguinte resultado:

```
GUARDA-REDES JOÃO LOPES ASSINA POR DOIS ANOS.html -> 0.04419642857142856

JAVI GARCÍA TEM DOIS PRETENDENTES.html -> 0.04

JUVENTUS PONDERA COLOCAR DOIS NOMES NA 'OPERAÇÃO JIMÉNEZ'.html ->
0.034374999999999996

HERNÂNI ESCAPA DE CIRURGIA, MAS PÁRA DOIS MESES.html -> 0.034374999999999996

JOÃO PEDRO RECUPERA LUGAR.html -> 0.012499999999999999

JOÃO MÁRIO COM PRETENDENTE ESPANHOL.html -> 0.011428571428571429

JOÃO VASCO NO ACADÉMICO VISEU.html -> 0.011428571428571429

PAULO FONSECA ATACA JOÃO PEDRO.html -> 0.011428571428571429

JAIME PINTO E JOÃO CARDOSO ANUNCIADOS COMO REFORÇOS.html -> 0.00982142857142857

OFERTA DO MARSELHA POR JOÃO MÁRIO NÃO AGRADA.html -> 0.00982142857142857
```

...

5 Conclusão

Na realização deste trabalho foram encontrados inicialmente alguns obstáculos, primariamente devido a não conhecer inicialmente a biblioteca TF-IDF e como trabalhar com ela. Porém, após algum estudo e alguma prática, foi possível aplicá-la no contexto escolhido de forma correta.

Existem alguns aspetos que poderiam ser melhorados na conceção deste projeto, entre os quais a interface, que poderia alterada de forma a ser mais facilmente utilizável. Uma interface web poderá vir a ser desenvolvida no futuro para ajudar com este problema. Apesar disto, considero que o trabalho corresponde ao que foi pedido, ao obter informação de artigos que fazem parte do website escolhido e ao eftuar pesquisa textual nos artigos de forma eficiente utilizando o algoritmo TF-IDF.