SPLN TP3: RSS Search
Aplicação Web para análise por TF-IDF

António Chaves A75870
Alexandre Teixeira A73547
Miguel Guimarães A66822

30 de Junho de 2019

1 Introdução

RSS é uma forma standard de distribuição de conteúdo de um distribuidor online, isto é, uma representação abstrata de conjuntos de noticias, artigos ou até peças.

Neste problema utilizaremos uma ferramenta de **scrap**, capaz de filtrar noticias, pertencentes a **RSS** feeds. Além disso, foi criado um script com base no algoritmo **TF-IDF**, abreviação do inglês term frequency–inverse document frequency, que significa frequência do termo–inverso da frequência nos documentos, capaz de classificar cada palavra em cada "noticia". De forma a representar a informação, implementamos uma **Aplicação Web** que representa os dados obtidos pelos comandos anteriores.

2 Manual de Utilização

Para a correta utilização da nossa aplicação recomendamos a instalação do **Scrapy**, assim como, a instalação de **NPM**.

- **NPM:** $ apt install npm
- **Scrapy:** $ pip3 install scrapy

Para iniciar o projeto é necessário instalar os pacotes com o comando **$ npm install**, e de seguida executá-lo com **$ npm run serve**.

Após a correta instalação é recomendado a utilização da ferramenta de criação de spiders, implementada na **aplicação web**, de forma a criar um modelo de **scrap**, com base no **XPATH** da página web a analisar.

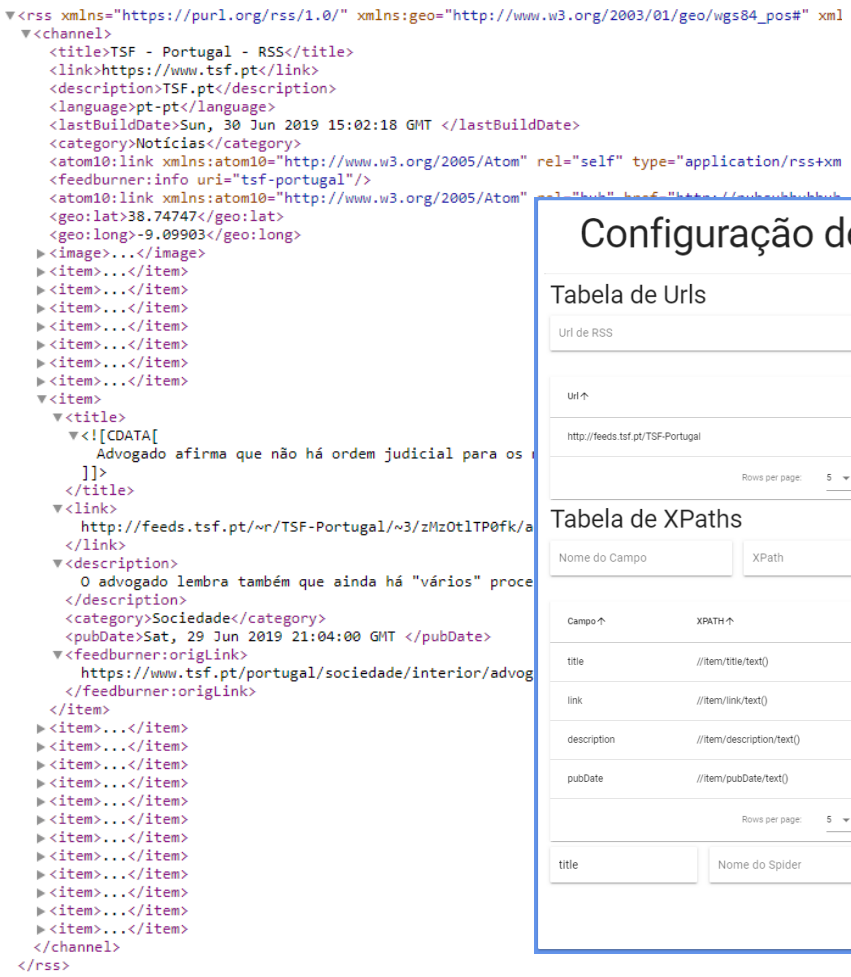
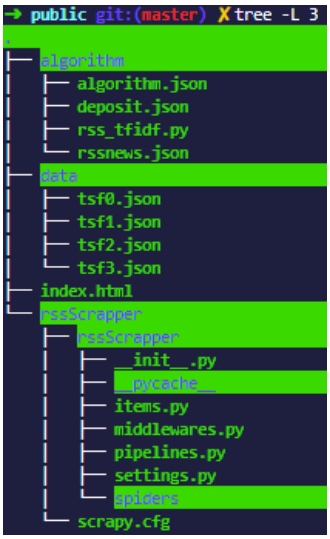


Figura 1: Elementos da página

Figura 2: Configuração de Spider

Após o download do novo **spider**, é recomendados copia-lo para a pasta dos **spiders**, em */RSS/public/rssScrapper/rssScrapper*.

Como podemos atentar, existem 3 sub-pastas, presentes na pasta *public*. Estas contêm os dois scripts python, o scrapy e o algoritmos de análise, assim como a pasta que guarda os dados.



Para tal são necessários os seguintes commandos:

1. **$ scrapy crawl SPIDER_NAME**
   Executar o comando na pasta *rssScrapper*

2. **$ python3 rss_tfidf.py**
   Executar o comando na pasta *algorithm*

Estes comandos premitem atualizar o conteúdo a ser mostrado na aplicação web, sendo este, devidamente processado e calculado com o algoritmo **TF-IDF**. De seguida podemos verificar o resultado obtido na aplicação web através do comando de desenvolvimento

- **$ npm run serve**
  Executado na raiz do projeto

3 Demonstração da Aplicação Web

3.1 Página Inicial

Na página inicial, é possível verificar algumas estatísticas iniciais, assim como uma breve discrição do algoritmo **TF-IDF**. Além disso é apresentada uma lista dos objetos retirados pelo **Scrapy**.

Figura 3: Pagina Inicial da Aplicação Web

De seguida, é possível clicar numa das linhas da tabela, e desta forma são utilizados os dados do algoritmo **TF-IDF**, para apresentar dois gráficos distintos.

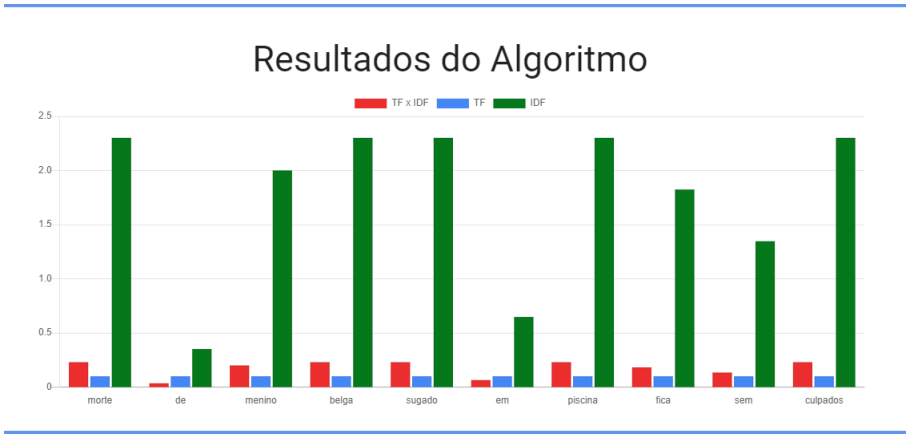


Figura 4: Gráfico relativo aos valores TF-IDF, TF e IDF

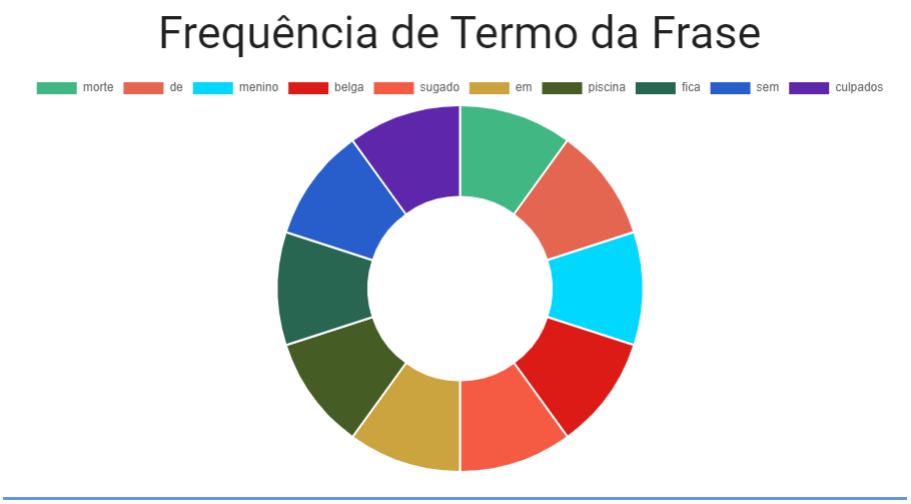


Figura 5: Gráfico relativo aos valores TF

Por fim, endereçando o problema inicial, procura com base no algoritmo TF-IDF, para isso, implementamos uma barra de pesquisa, capaz de analisar os valores TF-IDF da palavra e, de seguida, ordenar as frases com base nos valores das restantes palavras. Com isto, é feito um calculo do somatório do TF-IDF, dividido pela quantidade de palavras distintas, presentes na frase.

Figura 6: Barra de Pesquisa

3.2 Página de Configuração

Como foi supra indicado, a página de configuração passa pelo processo de criação de um spider (script em python executado pelo scrapy) sendo que, permite ao utilizador criar o seu próprio scraper com base na estrutura da página e o caminho XPATH para os elementos pretendidos.

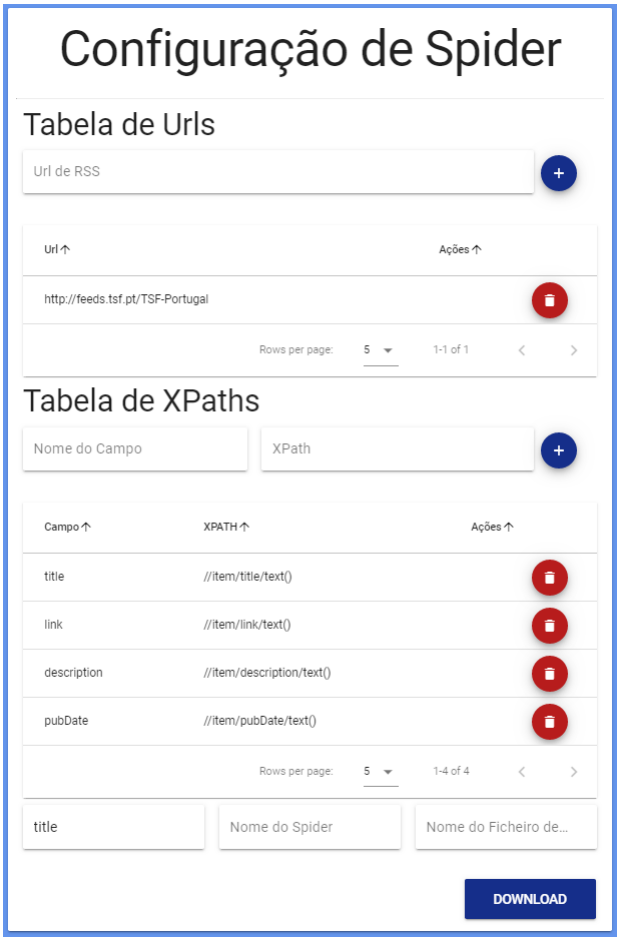


Figura 7: Página de Configuração

4 Conclusão

O presente documento foi redigido com vista a resumir o breve estudo da Scrapy com especial foco no potencial de utilização da mesma, realçado no segundo ponto do relatório.
Sendo o Scrapy uma ferramenta de extração textual, aquando da exemplificação de casos de utilização no contexto de NLP, a sua função reside no fornecimento dos pedaços de texto cujo processamento é atingido através de comandos gerais de python. O estudo da ferramenta em questão permitiu perceber o potencial do Scrapy, a facilidade com que se pode executar o varrimento dos urls fornecidos e que mesmo scripts de pequena dimensão permitem extrair grandes quantidades de informação.