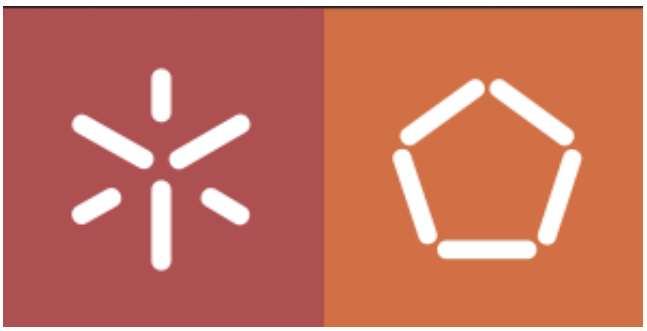

Universidade do Minho

Mestrado em Engenharia Informática

Scripting no Processamento de Linguagem Natural

Análise de Poemas - Python

João Vieira

A76516

29 de Junho de 2020

Conteúdo

1 Introdução

Este relatório é relativo ao último trabalho individual para a U.C. de *Scripting* no Processamento de Linguagem Natural. Foram apresentados variados temas no âmbito da disciplina, e o tema por mim escolhido foi o de **Análise de Poemas**. Esta ferramenta em *Python* serve para realizar o processamento de um poema, identificar as várias propriedades do poema e no final imprimir para um ficheiro *markdown* estas informações.

2 Execução do Programa

Primeiramente, vamos explicitar como se realiza a execução do *script* assim como a estrutura que o ficheiro de *input* tem que ter.

Para o processamento correr com sucesso estrutura do ficheiro de texto terá que ser como no exemplo a seguir:

```
AUTOR: Antero de Quental
TITULO: A UM POETA
POEMA:
Tu, que dormes, espírito sereno,
Posto à sombra dos cedros seculares,
Como um levita à sombra dos altares,
Longe da luta e do fragor terreno,

Acorda! é tempo! O Sol, já alto e pleno,
Afugentou as larvas tumulares...
Para surgir do seio desses mares,
Um mundo novo espera só um aceno...

Escuta! é a grande voz das multidões.
São teus irmãos que se erguem! são canções...
Mas de guerra... e são vozes de rebate!

Ergue-te, pois, soldado do Futuro,
E dos raios de luz do sonho puro,
Sonhador, faze espada de combate!
```

Com o devido ficheiro de *input* (poema3.txt) é só executar o seguinte comando para realizar o processamento:

```
python3 processor.py inputs/poema3.txt
```

No final o *script* cria o devido ficheiro *markdown* com as informações pretendidas. Esse novo ficheiro terá o mesmo nome do ficheiro dado como argumento, e estará na mesma diretoria do *script*. Por ex: **poema3.txt - poema3.md**.

Recomenda-se a leitura do ficheiro *markdown* num qualquer visualizador produzido para o efeito na *web*.

3 Desenvolvimento

Nesta secção, irei explicitar o desenvolvimento feito para obter cada secção de informação do poema.

- **Detalhes**

  Nesta secção é feita a contagem do nº de estrofes e versos. Também são definidos os tipos (denominação) de cada estrofe quanto ao nº de versos. A identificação do título, autor e poema (estrofes e versos) são feitas através de expressões regulares.

- **Rimas**

  Quanto às rimas, devido à grande quantidade de casos rimáticos que podemos ter num poema, apenas foram definidos os casos mais gerais que acontecem em estrofes do tipo quadra. Sendo assim, o *script* parte as estrofes em versos, e compara as últimas palavras de cada um para ver se rimam. Consoante o esquema rimático, a rima pode ser cruzada, interpolada, emparelhada, ou versos soltos.

- **Aliterações**

  As aliterações são figuras de estilo que produzem repetições sonoras, neste caso de uma dada consoante.

  Foi necessário analisar cada verso do poema por este fenómeno, identificando as várias consoantes presentes no ínicio de palavras e o seu nº de ocorrências. Caso o nº de ocorrências de uma consoante seja maior ou igual a 3, então esse verso é caracterizado como uma aliteração.

- **Assonâncias**

  As assonâncias, tal como as aliterações, são figuras de estilo que produzem repetições sonoras, mas neste caso são sons de vogais. Foi novamente necessário analisar cada verso do poema, e identificar as vogais presentes e o seu nº de ocorrências. Como a quantidade de vogais é menor que a de consoantes, o nº de ocorrências para ser dado como assonância é maior ou igual que 5. Como é normal, este fenómeno é mais comum do que a aliteração.

- **Classes Gramaticais**

  Para esta secção, foram agrupadas a maior parte das palavras do poema na sua respetiva classe gramatical. Para manter esta informação mais concisa, apenas foram consideradas as seguintes classes: substantivos, advérbios, adjetivos e verbos (infinitivo).

  Como tal, como esta análise depende do processamento de uma linguagem natural, neste caso a língua portuguesa, foi utilizada a biblioteca *spacy*. Esta ferramenta permitiu dividir o poema em *tokens* (palavras) e consoante o *part-of-speech* de cada um, inserir na respetiva classe caso seja substantivo, advérbio, adjetivo ou verbo.

- **Famílias Semânticas**

  Por fim, nesta última secção o objetivo foi tentar agrupar palavras do poema que estariam na mesma família semântica ou que têm alguma similaridade semântica.

  Para tal, foi novamente usada a ferramenta *spacy*, que permite identificar a similaridade entre diferentes palavras através da função *similarity()*. Cada um desses *tokens* são comparados com os restantes presentes no poema, e caso o índice de similaridade seja maior ou igual a 0.50 (similaridade razoável), estes são agrupados em família.

4 Conclusão

A resolução deste projeto permitiu aplicar várias formas e ferramentas de processamento textual, no âmbito da linguagem natural. Desde a utilização de expressões regulares para identificação de fenómenos ou partes textuais, até a utilização de ferramentas como o *spacy* que permitem analisar diferentes propriedades de palavras.

Em suma, o trabalho foi bastante enriquecedor para consolidar algumas das matérias lecionadas durante a U.C., mas também para ganhar uma maior experiência na utilização desta linguagem de *scripting* tão importante como é o **Python**.