Mestrado em Engenharia Informática
Scripting no Processamento de Linguagem Natural Abril, 2024

TP1 - SentiPT

Grupo 4

Vasco Oliveira (pg54269), João Loureiro (pg53924) and Luís Fernandes (pg54019)

Resumo: Este relatório aborda o desenvolvimento de um Trabalho Prático para a cadeira SPLN, que consistiu na construção de uma ferramenta de *sentiment analysis* de texto que calcula a intensidade de um texto, conforme um *dataset* fornecido. Possui também uma comparação com ferramentas semelhantes, de forma a medir o seu desempenho.

Palavras-chave: *Sentiment Analysis*, *spacey*, polaridade, sentimento, LEIA, VADER, texto, palavras

Table of Contents

1 Introdução

Este trabalho teve como objetivo criar uma *script* para determinar o sentimento de um dado texto, que pode ser encontrada no ficheiro *sentiPT.py*, que, através de um *dataset* contendo várias palavras e o seu valor positivo ou negativo, calcula a polaridade de uma determinada frase ou texto, isto é, se é uma frase de sentimento positivo ou negativo.

2 Dataset

2.1 Criação

Para podermos calcular com maior precisão a polaridade de cada frase, foi necessário criar um *dataset* único com base em vários *datasets* diferentes de análise de sentimento: o *final_lema.csv*.

O *CSV* contém três colunas: **palavra** (a palavra em questão), **polaridade** (valor de polaridade da palavra) e **tipo** (tipo da palavra, se aplicável - a aprofundar na próxima secção). Usando o *dataset* do *LEIA* como base, decidimos que os valores de polaridade se iam enquadrar numa escala de -4 (sentimento negativo) a 4 (sentimento positivo), tomando o 0 um valor neutro.

Após pesquisarmos por vários *datasets* e observarmos o seu conteúdo, utilizamos aqueles que seguiam formatos consistentes e racionais:

- ***vader_lexicon_ptbr.txt***: *dataset* utilizado pela biblioteca *LEIA*, versão portuguesa do *dataset* uitilizado pela biblioteca *VADER*.
- ***sentilex2.txt***: *dataset* fornecido pelo o docente na diretoria *LinguaKit*.
- ***sentiment_lexicon_twitch.csv***: *dataset* encontrado no site *Zenodo* que contem alguns *emojis* frequentemente usados na plataforma de *streaming Twitch* com a sua polaridade associada.

O *datset vader* não necessitou de tratamento adicional, já que se encontravam no formato pretendido para o *dataset* final. Apenas foi necessário guardar no *dataset* cada palavra ou *emoticon* e a polaridade média a ela associada.

No caso dos *emojis*, estes também se encontravam no formato pretendido, no entanto continham *strings* relacionadas a *emojis* exclusivos à *Twitch* e que não possuem correspondente no formato *unicode* (por exemplo *'residentsleeper'*). Assim, extraímos, através do uso de expressões regulares, os *emojis unicode* bem como a polaridade a eles associada, que são posteriormente acrescentados ao *dataset* final.

Finalmente, o *dataset sentilex2*, em vez de apresentar o valor da polaridade de cada palavra, apenas identificava se esta era positiva ou negativa. Como rever manualmente cada palavra é um processo demorado e inconcebível para o tempo estipulado para a conclusão deste trabalho, caso seja detetada uma palavra ausente do *dataset* final, decidimos atribuir um valor de polaridade escolhido ao calhas entre intervalos de valor médios conforme o seu identificador: entre [-1.3, -0.5] se for negativo e [0.5, 1.3] se for positivo.

Ainda sobre este *dataset*, ao analisá-lo, reparamos que todas as palavras a partir da linha 3789 são explícitas. De forma a diferenciar estas das restantes palavras negativas, aumentamos os valores dos intervalos negativos para [-3.4, -2.2], para que o seu efeito no

cálculo de polaridade seja mais notável. A palavra *'trepar'*, apesar de se encontrar no meio dos explícitos, encontra-se isenta deste acréscimo, já que o seu significado não é tão explicito em português europeu como é no português brasileiro, em qual está escrito este *dataset*.

A *script* utilizada para gerar este *dataset* encontra-se no ficheiro *generateDataset.py* que tem como *output* o *dataset* intermédio *sentiment.csv*, não possuindo as palavras tipadas.

2.2 Tipagem

Depois de termos coletado informação e definido o formato do *dataset*, estava na altura de definir os diferentes **tipos** de cada palavra:

- **EMOJI:** este tipo já é definido no processo de criação do *dataset*, e define se uma palavra é simplesmente um *emoji*.
- **NEG:** é o tipo que define se uma palavra é uma negação, isto é, se a palavra irá negar a próxima palavra, dando à próxima palavra polaridade negativa se ela tiver polaridade positiva e polaridade positiva se tiver polaridade negativa. exemplo: "não". As palavras que são consideradas negações estão contidas no ficheiro "negate.txt".
- **NEGT**: é o tipo que define se uma palavra é uma negação e um termo ao mesmo tempo, isto é se a palavra contém mais do que duas palavras e se nega a próxima palavra exemplo: "não querer". As palavras que são consideradas negações estão contidas no ficheiro "negate.txt".
- **INCR**: é o tipo que define se uma palavra é um *booster* positivo, isto é, se intensifica a próxima palavra. exemplo: excepcionalmente. As palavras que são consideradas *boosters* estão contidas no ficheiro "booster.txt".
- **DECR**: é o tipo que define se uma palavra é um *booster* negativo, isto é, se diminui a intensidade da próxima palavra. exemplo: excepcionalmente. As palavras que são consideradas *boosters* estão contidas no ficheiro "booster.txt".
- **TERM**: caso uma palavra não seja nenhum dos tipos apresentados anteriormente, e contenha mais do que duas palavras, ela será definido como um termo. exemplo: muito adorável
- **(tipo vazio):** caso uma palavra não seja nenhum dos tipos apresentados anteriormente, ela não terá um tipo. exemplo: chá

  Depois de ter-se atribuido os tipos ao *dataset sentiment.csv*, também foram adicionados todos os negadores e *boosters* que não estavam nesse *dataset*, passando este novo *dataset* a ser chamado de *final.csv*.

  Por fim, também foi criado o *dataset final_lema.csv*, que é igual ao *dataset final.csv* mas todos os verbos estão no seu *lemma*.

3 Lógica do Programa

O *dataset final_lemma* foi guardado numa estrutura de dados para posteriormente ser utilizado no cálculo da polaridade dos textos.

A nova estrutura, divide as diferentes palavras por tipo, sendo essa estrutura, um dicionário com os tipos a *keys*.

Este tratamento foi efetuado, pois diferentes tipos exigem diferentes tratamentos.

Relativamente ao texto recebido como input, sendo ele uma frase ou um ficheiro. Para analisar a polaridade do texto, é necessário fazer alguns tratamentos, relativamente aos verbos e aos acentos.

De forma a simplificar a análise sobre o texto, todos os verbos existentes nele são transformado para a sua forma *lemma* utilizando a biblioteca *spacy*, assim em vez de ter uma entrada por cada conjugação do verbo, apenas temos uma para o lema do verbo. Uma outra transformação foi a remoção de todos os acentos existentes no texto.

Após este tratamento, a análise da polaridade do texto é inicializada.

Como paço inicial, iremos percorrer uma lista das palavras do tipo *TERM*, *EMOJI* e sem tipo. Isto é feito desta forma, pois esta palavras são dependentes dos outros tipos, isto é, o restantes tipo afetam a polaridade destas, logo é necessário haver uma análise cuidadosa.

Por cada palavra encontrada, iremos procurar as palavras anteriores a esta, com o intuito de encontrar palavras dos tipos *NEG*, *NEGT* *INCR* e *DECR* com o objetivo de identificar se a polaridade da palavra sofrerá alterações.

Com isto temos diferentes casos, que podem ocorrer nesta análise:

- Não encontra nenhum ⇒ a polaridade da palavra é mantida;
- Existe um *DECR* ⇒ Polaridade original é reduzida para metade;
- Existe um *INCR* ⇒ Polaridade original é aumentada para o dobro;
- Existe um *NEG* ou *NEGT* ⇒ Polaridade original é negada;

Assim sendo, sempre que uma palavra do *dataset* é encontrada, o que foi explicado anteriormente será efetuado. Contudo, no caso de *EMOJI* não acontece o que foi descrito.

No final da análise, o valor de polaridade obtido é normalizado, e escrito no output.

4 Flags do Projeto/Comandos

Para melhorar a experiência do utilizador decidimos criar flags opcionais ao projeto, sendo estas as flags definindas (mesmo com ou sem flags o programa dará sempre a polaridade do texto):

- **-a:** Fornece o número total de palavras no texto onde for classificada a polaridade.
- **-n:** Fornece o número total de "palavras" que tiveram um efeito negativo na polaridade.
- **-p:** Fornece o número total de "palavras" que tiveram um efeito positivo na polaridade.
- **-f:** Para além da flag é necessário introduzir a diretoria dum ficheiro .txt. O programa irá calcular a polaridade desse que foi introduzido.

- **-t:** Para além da flag é necessário introduzir uma string dentro de aspas. O programa irá calcular a polaridade dessa string que foi introduzida.

Para usar o programa basta começar com o comando "flit install" para instalar o pacote sentiPT, e de seguida introduzir no terminal "sentiPT" mais as flags que forem pretendidas a serem usadas.

5 Comparação com outras ferramentas

Para poder comparar o *LEIA*, *Vader* e a nossa ferramenta, *SentiPT*, decidimos calcular a polaridade de sentimento para cada capítulo do livro *'Harry Potter e a Pedra Filosofal'*, tendo os resultados sido organizados nos seguintes histogramas:

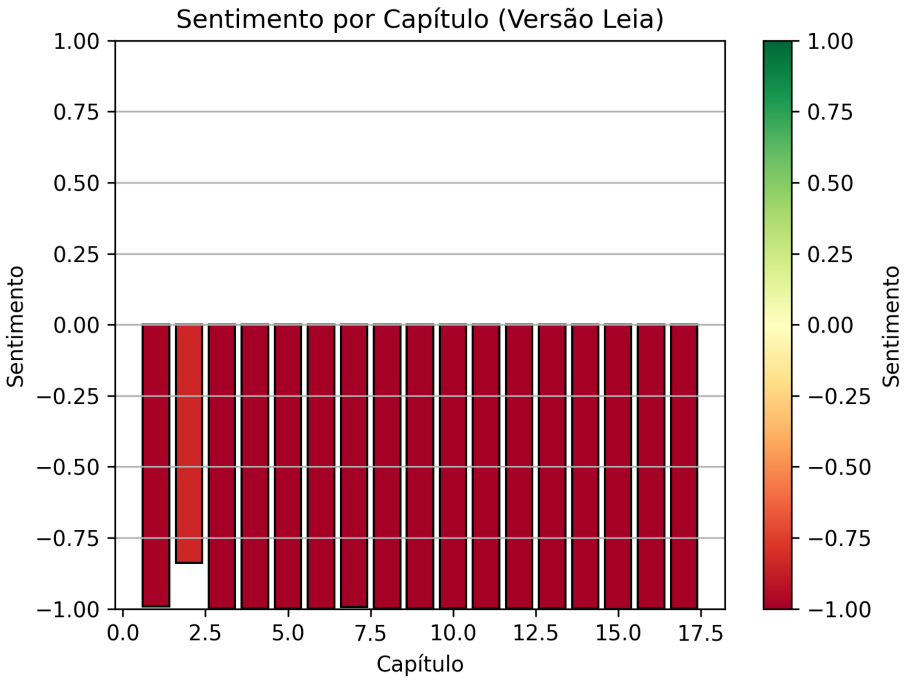

Figure 1: Variação de Polaridade/Capítulo usando o *LeIA*

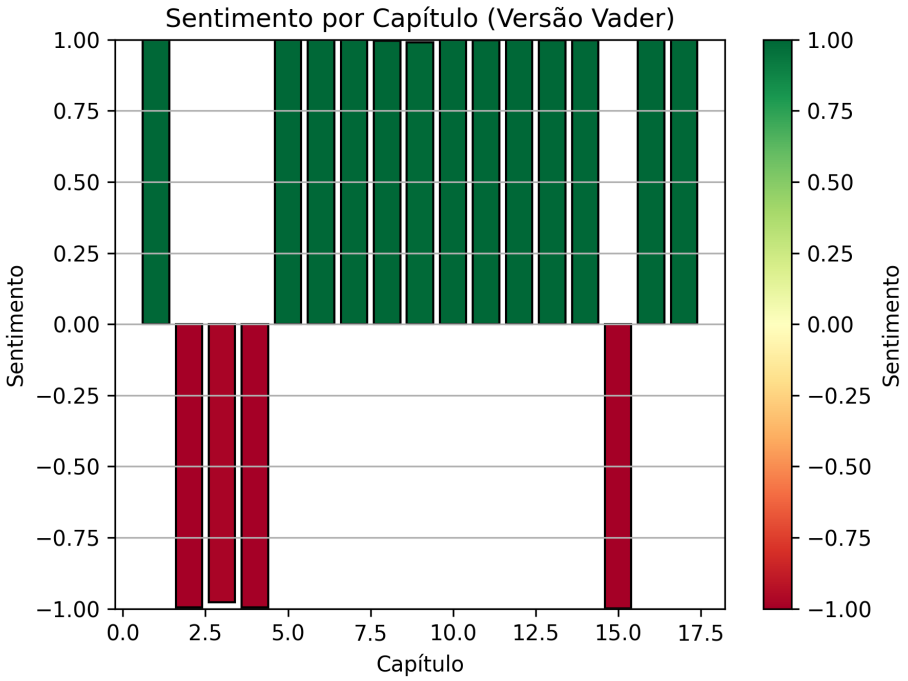

Figure 2: Variação de Polaridade/Capítulo usando o *vader*

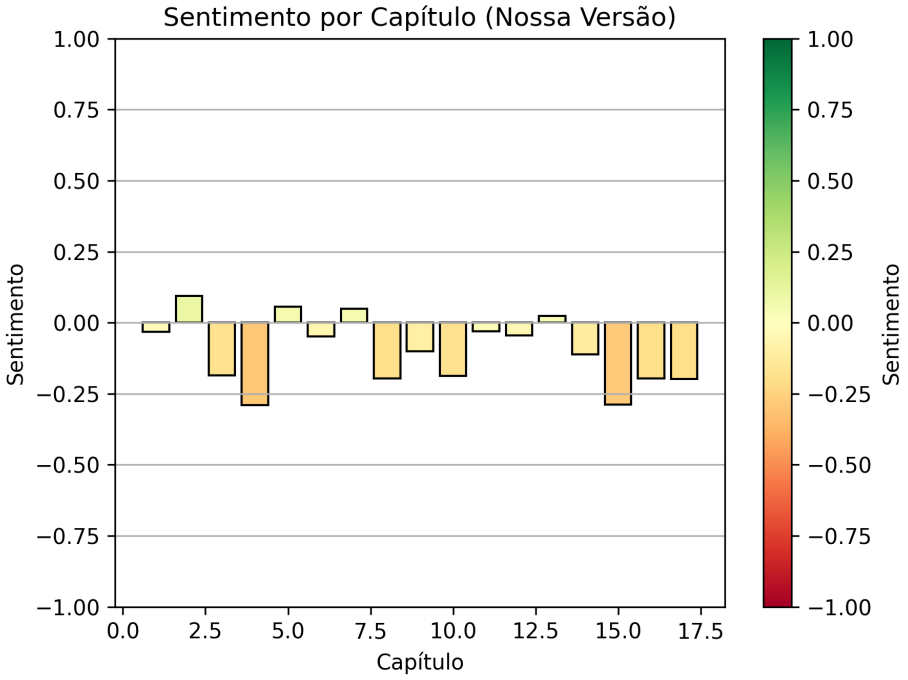

Figure 3: Variação de Polaridade/Capítulo usando o nosso programa

Podemos observar que ambos *Leia* e *Vader* apenas apresentam valores extremos para cada um dos capítulos, tendo o *Leia* apenas apresentado valores negativos, enquanto que o *Vader* também apresenta valores positivos.

Já a nossa ferramenta, apresenta valores muito mais centralizados, não possuindo nenhum valor no extremo que qualquer intervalo. Estes resultados resultam das diferentes filosofias de cálculo entre as ferramentas: o *vader* e o *leia* funcionam mais pela percentagem de palavras positivas ou negativas que foram encontradas (sendo esta percentagem muito baixa, visto que a maior parte das palavras era neutra, acabando por encontrar muito poucas palavras positivas ou negativos num capítulo), já no nosso caso a polaridade está constantemente a aumentar e a diminuir, acabando por não ter uma grande divergência na polaridade negativa e positiva dos diferentes capítulos.

6 Conclusão

Com este projeto conseguimos aplicar alguns dos conhecimentos adquiridos nas aulas desta cadeira, e conseguimos desenvolver uma ferramenta, que calcula, para a maioria dos casos, o sentimento de um texto com alguma precisão.

Em termos de aspetos a melhorar, apesar de acharmos que o *dataset* gerado tem uma quantidade considerável de entradas, este podia ser sempre expandido com mais *datasets* que pudessem eventualmente surgir. Podíamos também ter aplicado mais funcionalidades da biblioteca *spacy* que foram eventualmente lecionadas, mas não a tempo de serem integradas neste projeto.

Concluindo, estamos satisfeitos com o resultado final da ferramenta, e temos a certeza que, com mais tempo, podíamos torná-la numa ferrementa mais robusta e fiável.