UMinho

Scripting no Processamento de Linguagem Natural

TP1 - Análise de Polaridade em Português

Bernardo Costa
(*PG53699*)

Diana Teixeira
(*PG53766*)

Conteúdo

1. Introdução

No âmbito da Unidade Curricular Scripting no Processamento de Linguagem Natural realizamos um Trabalho Prático que se divide em duas partes. Primeiramente, construímos um módulo de *Sentiment Analysis* para o português baseado em léxicos de sentimento. Nesta parte pudemos explorar muito daquilo que está relacionado com técnicas de processamento de linguagem natural, análise de texto, e linguística computacional para identificar, extrair, quantificar e estudar estados emocionais. Vivemos numa era digital em que a quantidade de texto gerado por humanos à nossa disposição é imensa e crescente, abrangendo desde breves comentários em redes sociais até livros enormes, sendo que tivemos a oportunidade de explorar ambos os casos. Nesse vasto mar de palavras, a capacidade de discernir as opiniões e sentimentos subjacentes apresenta um desafio excitante.

Por um lado, entender o sentimento por trás de um texto pode fornecer *insights* valiosos para empresas, criadores de políticas e indivíduos, permitindo-lhes tomar decisões mais informadas com base na percepção pública. Por outro lado, a natureza subjetiva e multifacetada da linguagem humana torna a análise de sentimentos uma área complexa, repleta de nuances e ambiguidades.

Relativamente à segunda parte, conseguimos obter mais conhecimento e prática sobre como organizar projetos de média dimensão de modo a que seja mais fácil para qualquer leitor/avaliador entender onde estão as informações procuradas.

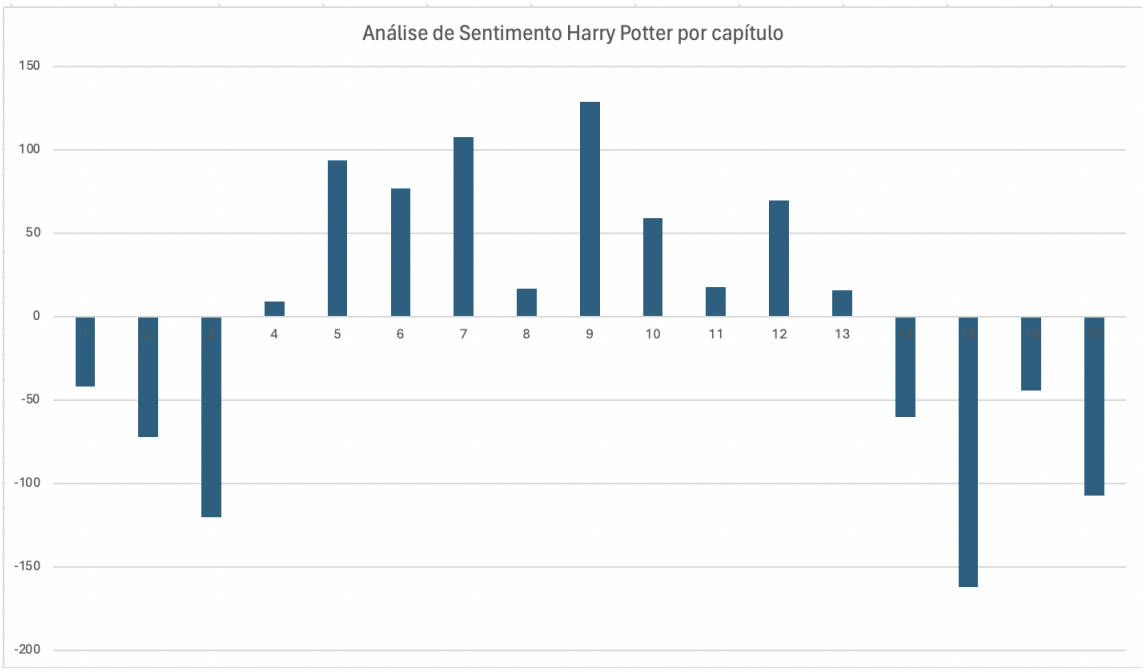
2. Análise de sentimento

Para realizar análise de sentimento em textos em português, aproveitamos recursos da biblioteca Spacy para processamento de linguagem natural. A ferramenta tem a peculariedade de analisar textos oriundos de duas fontes distintas: livros e tweets, perguntando qual estará a ser analisado. Isto deve-se ao facto de estarmos a trabalhar com portáteis de já alguma idade onde existe uma dificuldade latente em executar várias vezes o mesmo programa, e como acabamos por analisar por capítulos o livro do Harry Potter, optamos por resolver este problema desta forma. No entanto, isto não é de maneira alguma um impedimento, como se pode observar no README.

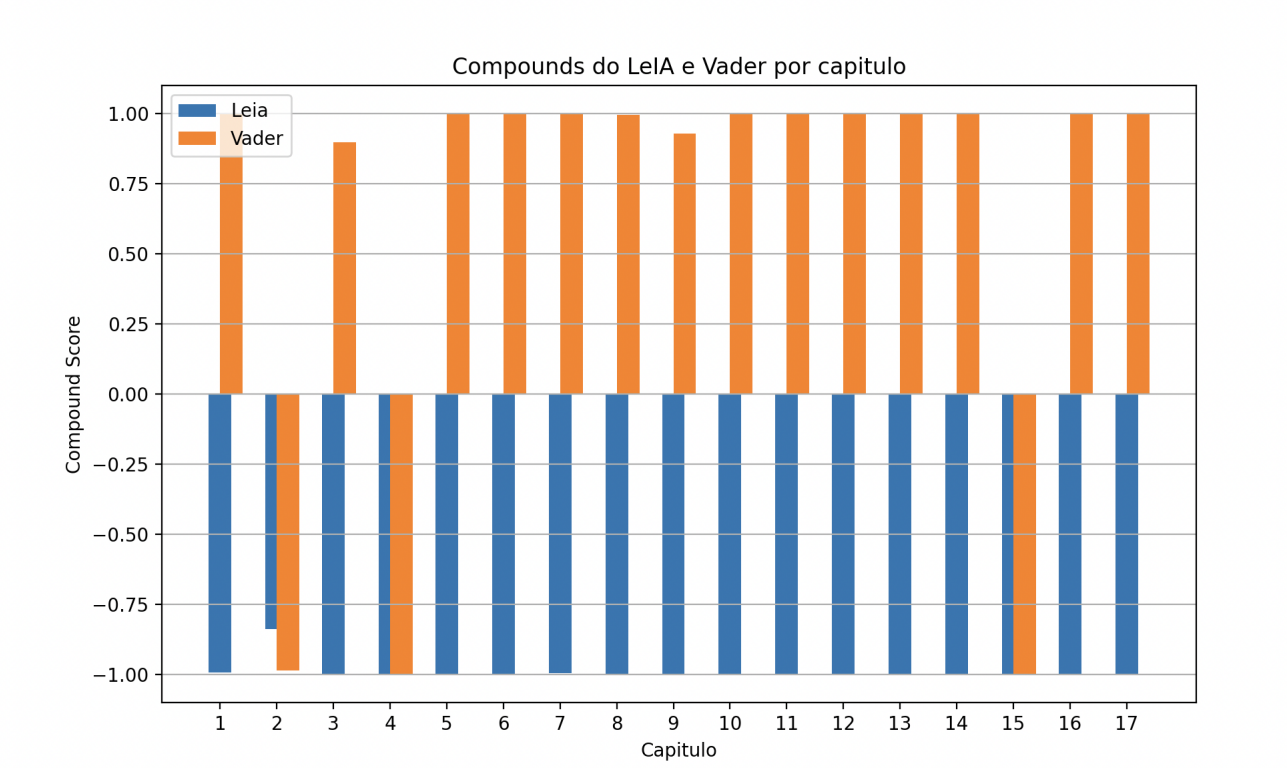
O script baseia-se em dicionários para emojis e para pontuações de sentimentos de palavras em português, armazenados no diretório data. Estes ficheiros são cruciais para o funcionamento do script, dado que contêm os dados necessários para a análise. Os emojis são substituídos pelas descrições que lhes correspondem, de modo a não perder a carga emocional que eles carregam. Após a substituição de emojis, o texto é tokenizado usando o Spacy, uma etapa fundamental para a análise de sentimento, pois permite isolar cada palavra para avaliação. A análise de sentimento propriamente dita é realizada com base no dicionário de sentimentos, atribuindo a cada palavra encontrada no texto uma pontuação. As pontuações das palavras num texto são somadas para chegar ao sentimento geral do texto.

Posteriormente, para realizar a análise de tweets, utilizamos um dataset que encontramos no Kaggle e que continha os tweets na íntegra e não apenas o seu ID. Como tal, extraímos os mesmos e baseamos as nossas 10 frases neles para testar a ferramenta de *Sentiment Analysis* aplicada a eles.

Para a análise do livro Harry Potter e a Pedra Filosofal (dividida em capítulos) com base na nossa ferramenta, obtivemos os seguintes resultados:



De seguida apresentamos também um histograma que compara a versão VADER e LeIA:



O que se vê pelo histograma é que o Vader e o LeIA têm resultados diametralmente opostos para a maioria dos capítulos, coincidindo apenas nos capítulos 2, 4 e 15. Isto deixou-nos surpreendidos.

Já a nossa ferramenta tende a concordar mais com o Vader dos capítulos 5 ao 13, sendo que nos restantes está mais próximo do LeIA.

No que diz respeito ao loading dos dados, o ficheiro que se refere a isso é o load_data. Esta ferramenta que desenvolvemos é destinada a consolidar dados de diferentes coleções num único ficheiro chamado sentiment_dict. A principal finalidade da load_data é reunir dados de várias fontes num único dicionário, facilitando assim o acesso e a manipulação de informações relacionadas a sentimentos de palavras ou expressões. As coleções manipuladas pela ferramenta incluem:

- Booster: Palavras que intensificam o sentimento de uma frase, originárias da biblioteca LeIA.
- Expressions: Expressões em português marcadas como positivas ou negativas
- Lex: Um conjunto de palavras e seus sentimentos associados
- Negate: Contém negações e expressões negativas

Esta ferramenta recorre ao spellchecker apenas na coleção lex, dado que o tamanho das outras é pequeno e fácil de corrigir.

3. Calibração

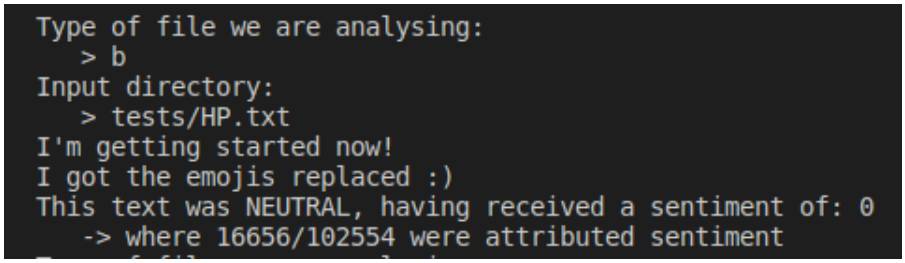
Após isto tudo, obtemos um dicionário de (palavra,sentimento) a partir do ficheiro que entitulamos de sentiment_dict_or.txt. A partir do qual, tal como podemos ver pelo out_or.txt, resultante do comando "sort -n res.txt > out.txt", obtivemos um sentimento muito positivo, resultante de um registo, às vezes incorreto (ex: linha 15607, "ID: mal ; 2.0"), e às vezes sem sentido (ex: linha 1, "ID: a ; -1.6"), visto este nosso dicionário se encontrar mal balanceado.

O que nos leva então à parte de calibração do nosso calculador de polaridade dos sentimentos, o que foi, sem dúvida, a parte em que o nosso grupo teve de tomar decisões mais arriscadas e até drásticas, visto o nosso objetivo ser obter uma polaridade no Harry Potter de 0.

Dito isto, o nosso grupo terá então, ao longo deste processo:

- Removendo palavras do dicionário, como é o caso de palavras como "a", "o", "os";
- Adicionando palavras novas, as quais são visíveis, na sua maioria, no final do ficheiro, a seguir à palavra "nunca"(linha 5271 do ficheiro sentiments_used.txt). Possuindo estas um sentimento que, entre nós e até com o auxílio de colegas, elegemos;
- Modificamos o sentimento de algumas palavras, nomeadamente:
  - Todas as palavras com valor -0.293 (boosters negativos) e -0.74 (todas as negações), que descemos para -2.0, visto o sentimento total do livro, no início estar a ser de +9000;
  - Algumas palavras, como "ele", "eu", "ela", "lhe", "mal", "mas", "matar", entre outras, que estavam com valores muito diferentes daqueles que achamos esperados.

Tendo estes passos vindo, não só da análise do ficheiro output obtido, como também do próprio dicionário, e até das palavras presentes no livro. E assim sendo, após muito tentativa e erro, teremos chegado uma versão final do nosso dicionário, o ficheiro sentiments_used.txt, que se encontra calibrado e cuja utilização resulta nos seguintes resultados, para o livro, e para as 10 frases (tweets), que utilizamos:

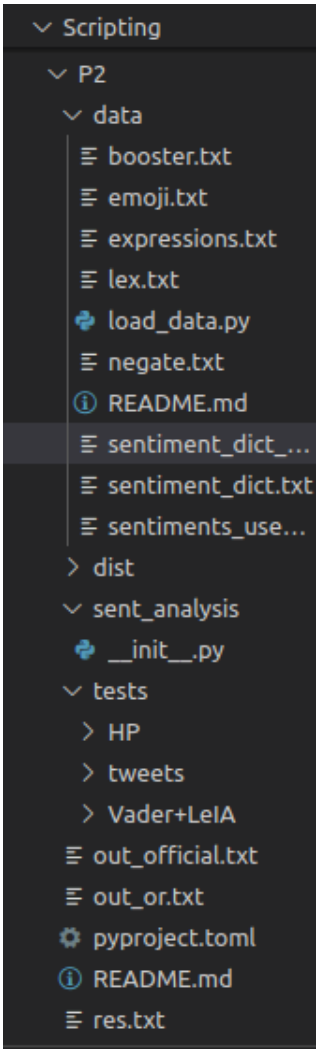


```
Type of file we are analysing:
   > t
Input directory:
   > tests/test10.txt
I'm getting started now!
I got the emojis replaced :)
This text was POSITIVE, having received a sentiment of: 1
  -> where 1/6 were attributed sentiment
This text was NEGATIVE, having received a sentiment of: -3
  -> where 5/21 were attributed sentiment
This text was NEGATIVE, having received a sentiment of: -4
  -> where 4/9 were attributed sentiment
This text was POSITIVE, having received a sentiment of: 6
  -> where 7/19 were attributed sentiment
This text was NEGATIVE, having received a sentiment of: -3
  -> where 3/5 were attributed sentiment
This text was NEGATIVE, having received a sentiment of: -1
  -> where 1/4 were attributed sentiment
This text was NEGATIVE, having received a sentiment of: -8
  -> where 5/14 were attributed sentiment
This text was NEGATIVE, having received a sentiment of: -2
  -> where 3/8 were attributed sentiment
This text was POSITIVE, having received a sentiment of: 1
  -> where 8/19 were attributed sentiment
This text was POSITIVE, having received a sentiment of: 2
  -> where 12/36 were attributed sentiment
This text was NEUTRAL, having received a sentiment of: 0
   -> where 0/1 were attributed sentiment
Type of file we are analysing:
   >
```

4. Sistema gerador com base em template-multifile

Para esta segunda parte do projeto, teremos optado por seguir o exemplo 2 que o professor terá falado, organizando o nosso trabalho num módulo python, semelhante ao do word_freq, realizado nas aulas, utilizando o jinja2 para a criação de templates para os README.md presentes e para o pyproject.toml.

O que nos leva então, à seguinte árvore de directorias:



As quais são visíveis no nosso github e de onde verificamos que, de acordo com os README, estão bem interligadas e enquadradas no nosso projeto.

Por fim, temos também os ficheiros makepyproject.py e readme.py que utilizam o jinja2 e geram os templates pretendidos para esta fase do trabalho, sendo que o makepyproject.py foi reaproveitado das aulas e, por sua vez, o readme.py foi criado com base num README genérico, mas que ia de acordo com aquilo que envisionamos e que achamos necessário ter.

5. Conclusão

Ao concluir esta tarefa no âmbito da análise de sentimentos em português, consideramos que realizamos um trabalho bem desenvolvido e bem documentado. A implementação desta ferramenta exigiu uma pesquisa detalhada e a análise de recursos como o o LeIA, fornecendo-nos uma base sólida para a construção de um léxico abrangente. O processo de construção da ferramenta envolveu algumas etapas mais críticas como a calibração, mas no cômputo geral, consideramos que realizamos uma análise precisa e refinada de textos em português.

O estudo do caso aplicado ao livro "Harry Potter e a Pedra Filosofal"foi particularmente ilustrativo, permitindo uma comparação direta entre a análise de sentimentos em diferentes idiomas e um teste extensivo da ferramenta que desenolvemos. A divisão do livro em capítulos e o cálculo das suas polaridades proporcionaram *insights* valiosos sobre a variação emocional ao longo da narrativa. Sendo que algo que nos surpreendeu foi a diferença de *compound* entre LeIA e Vader.

Lamentamos o atraso na entrega que ocorreu por motivos de força maior. No entanto, julgamos que conseguimos, apesar disso, desenvolver um trabalho competente e bem estruturado.