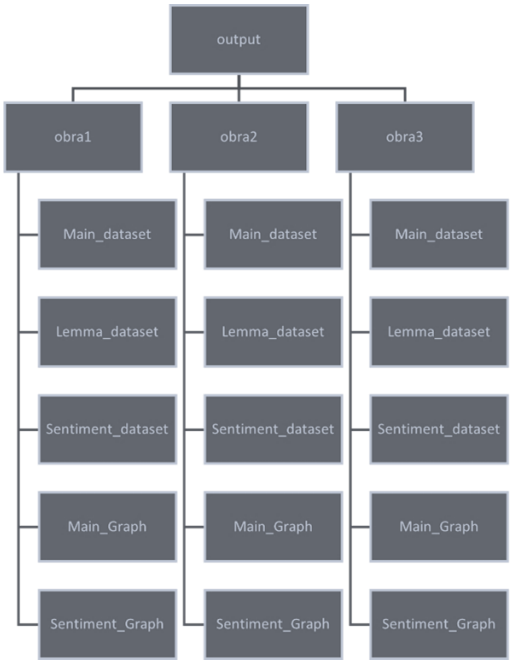
Metadados

Apresentação

Introduziu-se o desafio de criar um script de python que analisasse pelo menos 4 obras e que, a partir dos dados extraídos, se criasse gráficos afim de poder visualizar as informações de forma simples e que fosse, de preferência, o mais automatizado possível.

Como material de input foram usados os ficheiros das obras de Camilo Castelo Branco, pré-formatados em formato Markdown e que se encontram uniformizados pela turma. Estes ficheiros foram fornecidos ao script de python que, usando *Natural language processing* e um sentimentalizador, extrai dados gramaticais e sentimentais e exporta para um ficheiro CSV. Este ficheiro é posteriormente usado para permitir a criação e exportação de gráficos usando bibliotecas de python, os quais podem depois ser analisados para extrair informações para investigação.

Resumidamente, a utilização deste projeto é muito simples: Colocar as obras para análise (previamente formatadas em Markdown) na pasta obras; se pretender, é possível editar o script para extrair mais ou menos elementos; executar o script principal (Main_Script.py) e esperar que seja criada a árvore do output:



Escolha do NLP/Categorizador Gramatical

No começo do projeto decidi testar como se comparava o Linguakit em relação ao Spacy, tendo chegado à conclusão de que ambos têm alguma percentagem de erro associado, apesar do Linguakit falhar ligeiramente menos. No entanto, tendo em conta que já me encontrava familiarizado com o Spacy e já tinha mais fluidez neste, decidi escolher como modelo e iniciar de forma imediata o trabalho, evitando uma curva de aprendizagem.

Escolha do modelo sentimentalizador

Quando foi feita a proposta de trabalho final, os docentes da Unidade Curricular sugeriram usar o modelo sentimentalizador Sentilex para a análise de sentimento das obras. No entanto, depois de descobrir uma falha no script (que foi prontamente comunicada aos docentes e à turma) e uma mais detalhada análise da forma como o modelo analisa, achei que seria muito limitador e tentei procurar uma alternativa. Escolhi o modelo VaderSentiment, um modelo que à partida não seria o melhor pois é treinado para análise a texto moderno e mais coloquial, mas que se provou capaz de analisar obras com escrita antiga na mesma. Este modelo, ao contrário do Sentilex que só devolve sentimento negativo e positivo (e calcula o neutro com base nesses dois), fornece sentimento negativo, neutro, positivo e composto (valor unidimensional que reflete o valor do sentimento, calculado por regras usando os 3 outros valores de sentimento).

Ferramentas usadas

- Para verificação da integridade dos datasets usou-se o Microsoft Excel
- Para gestão (eliminação) da base de dados das obras usou-se um script de DOS (Batch)
- Para verificar a edição manual dos gráficos com alta qualidade e editabilidade pós-script usou-se o Adobe Illustrator

Bibliotecas de Python:

- Spacy (Natural Language Processing)
- VaderSentiment (Modelo Sentimentalizador)
- Counter (Selecionar os elementos necessários para extração e contar as ocorrências)
- JJCLI (Organização/gestão de ficheiros)
- RegularExpressions (Procura/substituição/repartição de textos)
- MatPlotLib (Criação dos Gráficos)
- Numpy (Criação de arrays para os gráficos)

Features extra

- Scripts inteiramente automáticos não sendo preciso intervenção humana em nenhuma das diferentes fases, o que permite automatização total.
- Os scripts automaticamente limpam os dados e corrigem erros que possam surgir, contanto não só com extração seletiva dos dados, como também por uma limpeza afim de impedir erros na base de dados.
- Organização dos datasets e gráficos numa estrutura hierárquica e clara (output).
- Possibilidade de ajustar facilmente a dimensão do dataset no script via variável de extração.
- Possibilidade de reescrever/regravar o output completo do script, permitindo alterações ao código *"on the fly"*, sem ter de apagar manualmente o output completo.
- Filtragem automática de diferentes nomes para a mesma personagens quando providenciada uma lista com os nomes de personagens principais.
- Possibilidade de correr os 4 scripts encadeados, ou apenas os scripts dos gráficos, o que permite ajustar apenas os gráficos.
- Criação de gráficos que se auto-ajustam aos datasets fornecidos.
- Gráficos criados com um aspeto visual limpo e apelativo no formato SVG (Vector Graphics) e EPS (Imagem com camadas editáveis).

Dificuldades encontradas

Ao longo da execução do trabalho fui confrontado com inúmeros problemas, alguns dos quais com resolução e outros que se revelaram desafiadores, sendo eles:

- Pouca familiarização com as bibliotecas de python relacionadas com matemática (MatPlotLib e Numpy) que contabilizam a maior parte do tempo dedicado ao projeto. Apesar da documentação sobre estas bibliotecas ser abundante, o trabalho investigativo neste trabalho não é muito comum, o que significa que não há scripts ou modelos já feitos e tem de se adaptar com cuidado todas as features e alterações gráficas dos gráficos. Como pretendia que estes scripts de python fossem o mais automáticos e autónomos possível, tive de criar um script que se conseguisse adaptar aos datasets (e logo às obras) fornecidos e que conseguisse produzir resultados com qualidade independentemente das dificuldades.
- Adaptabilidade, pois como os scripts de gráficos têm de se adaptar a datasets maiores ou menores, têm de conseguir manter um aspeto gráfico consistente, o que se prova difícil pois não se consegue prever exatamente como será um gráfico com mais ou menos elementos e se isso irá causar com

que os elementos choquem uns contra os outros e prejudiquem ou mesmo impeçam a visibilidade dos gráficos.

- A gestão da filtragem de nomes diferentes para a mesma personagem, pois é difícil (não só para a máquina como para um ser humano) distinguir se uma dada ocorrência de nome se refere à personagem principal da obra ou a uma personagem secundária. O mecanismo que criei, apesar de não ser perfeito, creio que mostra ser possível automatizar uma ação que seria, à primeira vista, estritamente manual e humana. Tentei também usar uma biblioteca da Wikipédia, mas esta tentativa revelou-se fútil ao não conseguir extrair a secção das personagens.
- Automatização na criação/eliminação/divisão do output, tendo em conta que se tem de organizar os ficheiros resultantes não só da extração dos metadados, mas também dos gráficos. Escolhi dividir o output por obra, o que dificulta a comparação de um determinado aspeto entre as várias obras, mas facilita a comparação de elementos dentro da própria obra, algo que considero mais importante.
- Falhas associadas às ferramentas usadas, nomeadamente spacy, que nem sempre conseguem detetar corretamente todos os elementos pedidos (falha bastante no reconhecimento de nomes), que infelizmente não podem ser corrigidas sem extensivo treino e correção. Tendo em conta que o trabalho se foca na automatização, achei que não deveria manipular o dataset manualmente e remover os elementos que, por ser um ser humano, sei que estão incorretos.

Automatização

Devido aos constrangimentos de tempo deste projeto, decidi priorizar criar um script funcional e o mais automático e autónomo possível, que fosse adaptativo e completo no seu percurso (Input de obras, output de gráficos e dataset estatístico tratado) ao invés de outros trabalhos que sejam mais focados na parte da apresentação dos dados, contendo gráficos mais adequados ou mais profissionais. Esta escolha faz sentido para mim, pois é uma na qual eu penso que consegui não só aprender mais do que se fizesse usando serviços de gráficos (LookerStudio, Excel, Tableau).

Este trabalho é também uma melhor prova dos meus conhecimentos técnicos e aptidões de resolução de problemas, pois consegui não só usar conhecimentos de processamento de linguagem natural como também conhecimentos de estatística (ambos adquiridos no primeiro semestre) e conhecimentos novos adquiridos na execução deste trabalho (nomeadamente sentimentalização, tratamento de dados estritamente usando métodos e bibliotecas matemáticas, automatização/encadeamento funcional de scripts).

Penso que se um dos objetivos deste trabalho é impressionar e inovar, devemos procurar sempre as soluções que nos trazem mais crescimento e que são mais distantes do *skillset* das pessoas de fora da unidade curricular, e devemos usar conhecimentos que a maioria das pessoas não possui.

Melhorias e ideias futuras

- Usar a biblioteca do ChatGPT do Python para gerar *prompts* a pedir a lista de personagens com nomes completos e depois usar essa lista para gerir os diferentes nomes para a mesam personagem nas obras.
- Atualizar o código dos 4 scripts para permitir executar usando modificadores de linha de comandos (por exemplo para modificar o número de amostras do dataset)
- Otimizar o código para ser mais rápido (ex. usar menos ciclos desnecessários) sem afetar a sua qualidade (aspetos como o NLP ser *large* em vez de *small* não podem ser alterados sem que efeitos negativos sejam sentidos no dataset)
- Usar o Linguakit em conjunto com o Spacy para produzir resultados mais fiéis (a probabilidade de falharem no mesmo elemento é baixa, e logo seria uma ótima forma de ter mais certezas de que se realizou uma extração total e completa dos dados a analisar).

Datasets e variáveis

O script principal gera três datasets:

- Dataset Main, que contém as variáveis "ID do livro", "nome do livro", "tipo de elemento extraído", "elemento extraído" e "contagem de ocorrências" sendo que de momento apenas as últimas 3 variáveis foram usadas. Ex. (2,Camilo-A_Brasileira_de_Prazins.md,locais,Braga,51)
- Dataset Lemma, que contém as variáveis "ID do livro", "nome do livro", "tipo de categoria gramatical do elemento extraído", "elemento extraído" e "contagem de ocorrências", sendo que de momento nenhuma das variáveis está em uso a não ser a contagem total dos tipos gramaticais. Ex. (2,Camilo-A_Brasileira_de_Prazins.md,Adjective,grande,96)
- Dataset Sentiment, que contém as variáveis "ID do livro", "nome do livro", "capítulo do livro", "tipo de sentimento", "estimativa do número de palavras com esse sentimento" e "percentagem de sentimento no capítulo" sendo que novamente de momento apenas a terceira, quarta e sexta variáveis foram usadas. Ex. (2,Camilo-A_Brasileira_de_Prazins.md,1,negative,889,0.015)

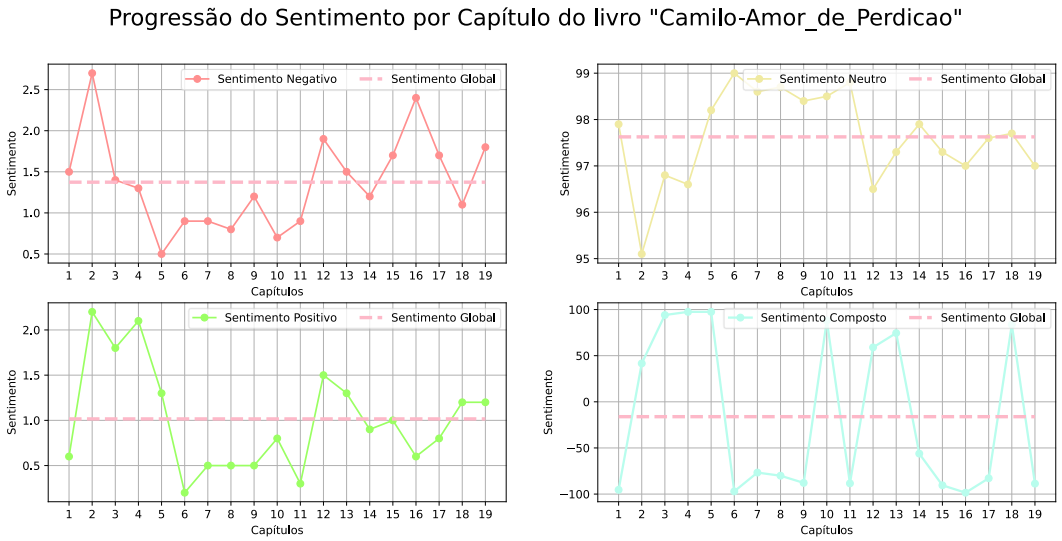
Análise dos gráficos

Para os gráficos de sentimento, decidi escolher uma matriz de gráficos 2 por 2, pois assim consegue-se sempre comparar lado a lado os 4 parâmetros do sentimento ao mesmo tempo, facilitando a sua análise. Defini as cores dos gráficos como o mais lógico pela psicologia humana, ou seja, vermelho para sentimento negativo, verde para positivo, amarelo como neutro, azul no parâmetro composto pois é uma cor que

contrasta bem com as outras e finalmente rosa-salmão para o sentimento global pois combina com as outras 4 cores.

A escala do eixo y do gráfico é ajustada automaticamente para se enquadrar aos pontos do dataset, pelo que varia de gráfico para gráfico. Ter a mesma escala iria implicar uma compressão forte da informação e iria impedir a sua fácil visualização (testei este método e como os valores de sentimento negativo e positivo são menores que o neutro, ficavam no gráfico como uma simples linha fina no fundo de todo). O sentimento global é calculado pela média de todos os pontos do dataset do gráfico e permite quantificar se um dado capítulo é mais ou menos intenso num dado sentimento.

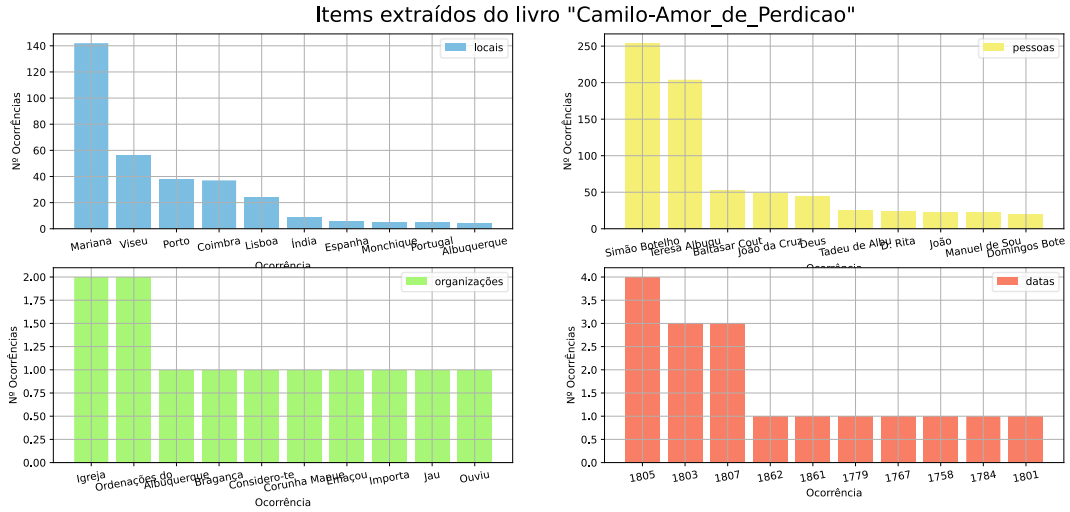
Escolhi fazer um gráfico de linhas pois é o mais indicado para retratar séries temporais, ou seja, dados relativos a variáveis que estão distribuídas ao longo de algo (neste caso capítulos). Uma outra opção seria fazer um gráfico de barras com valores positivos e negativos, mas na minha opinião, esse gráfico não ilustra tão bem a mudança de valor de um capítulo para o seguinte como o gráfico de linha que usa não só os pontos dos valores como a linha de declive entre os pontos para realçar se houve subida ou descida do dado valor.



No que toca aos gráficos do dataset main (novamente matriz, não só para manter a estética como também para potencialmente detetar algum tipo de correlação ou aproximação entre os elementos extraídos de cada tipo) escolhi a cor azul para locais pois é frequentemente associada à tranquilidade e serenidade; a cor amarela para as pessoas pois é uma cor que invoca energia e a vitalidade das pessoas; verde para as organizações pois é associada ao crescimento, estabilidade e confiabilidade; e vermelho para as datas pois é uma cor intensa e vibrante que transmite uma sensação de destaque e impacto. (providenciado pelo ChatGPT)

Novamente a escala do eixo y do gráfico é ajustada automaticamente para se enquadrar aos valores das barras do dataset, pelo que varia de gráfico para gráfico. Ter a mesma escala não iria contribuir de forma nenhuma para a visualização dos dados pois tratam-se de variáveis completamente distintas umas das outras.

Escolhi gráficos de barras pois são os mais indicados para variáveis qualitativas (categóricas) e demonstram de forma clara diferenças nos valores das variáveis.



Nesta última parte do trabalho, pode-se colocar a questão de porque ter extraído 10 elementos de cada categoria (e de seguida realizar 10 gráficos). A resposta seria que foi apenas um número de teste, e que o verdadeiro número a escolher, o número no qual os elementos começam a ser a mais e daí redundantes, pode ser facilmente determinado ao experimentar variar a variável “scrapenumb” no começo do script principal e ver que efeitos isso teria nos gráficos, podendo ser fácil e rapidamente ajustado a gosto, sem ter de realizar trabalho nenhum. De notar que os gráficos que não cumpram o requisito mínimo de entidades extraídas, ou seja, se uma dada obra não contém um número de elementos igual ou superior ao scrapenumb não será gerado gráfico, portanto há que tomar precauções com o ajuste do número.

Bibliografia

OpenAI. (2023). ChatGPT (May 24 version) [Large language model]. https://chat.openai.com/chat

Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014. https://github.com/cjhutto/vaderSentiment

Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*. https://spacy.io/

https://github.com/unsezeros/AVD2023-Ricardo/tree/main/Final_Trabalho_Final