MINERAÇÃO DE TEXTOS

Parte 4 - ESCORE DE SENTIMENTOS

Instalação do pacote AFINN:

Nesta etapa, a primeira providência é instalar o pacote **AFINN**. Isso permite medir a intensidade das palavras classificadas como "positivas" ou "negativas", num escore que varia de **-5 (muito negativo)** até **+5 (muito positivo)**. O comando de instalação é

- pip install afinn

Obs: ele só precisa ser instalado uma única vez por máquina. Feito isso, pode-se importar a sua biblioteca.

Importação das bibliotecas:

In [1]:
```
import pandas as pd
import nltk
import re
import matplotlib.pyplot as plt
from afinn import Afinn

from nltk.corpus import stopwords
from nltk.corpus import opinion_lexicon
from nltk.tokenize import word_tokenize

nltk.download('stopwords', quiet=True)
nltk.download('opinion_lexicon', quiet=True)
nltk.download('punkt', quiet=True)

plt.style.use('ggplot')
```

Importação do texto de Stevenson:

In [2]:
```
df = pd.read_csv('data/TheAmateurEmigrant.txt',sep='\t')\
    .dropna()

df.head()
```

Out[2]:

	text
0	THE AMATEUR EMIGRANT
1	THE SECOND CABIN
2	I first encountered my fellow-passengers on th...
3	Thence we descended the Clyde in no familiar s...
4	on each other as on possible enemies. A few S...

Preparação do texto::

In [3]:

```
# Adicionando linhas com números para dividir o texto em seções:
df['line'] = range(1, len(df) + 1)

df.head()
```

Out[3]:

	text	line
0	THE AMATEUR EMIGRANT	1
1	THE SECOND CABIN	2
2	I first encountered my fellow-passengers on th...	3
3	Thence we descended the Clyde in no familiar s...	4
4	on each other as on possible enemies. A few S...	5

In [4]:

```
# Limpando e tokenizando o texto:
def clean_text(text):
    text = text.lower()
    text = text.replace("'", '')
    text = re.sub(r'[^\w]', ' ', text)
    text = re.sub(r'\s+', ' ', text)
    text = text.strip()
    return text

df['text'] = df['text'].map(clean_text)
df['text'] = df['text'].map(word_tokenize)

df.head()
```

Out[4]:

	text	line
0	[the, amateur, emigrant]	1
1	[the, second, cabin]	2
2	[i, first, encountered, my, fellow, passengers...	3
3	[thence, we, descended, the, clyde, in, no, fa...	4
4	[on, each, other, as, on, possible, enemies, a...	5

In [5]:

```
df = df.explode('text').rename(columns={'text': 'token'})

df.head()
```

Out[5]:

	token	line
0	the	1
0	amateur	1
0	emigrant	1
1	the	2
1	second	2

Criação do Escore de Sentimentos

In [6]:
```
afinn_scorer = Afinn()

df['score'] = df['token'].map(afinn_scorer.score).astype(int)
df = df[df['score'] != 0]
```

Obs: o uso do != 0 acima faz com que as palavras de escore 0 (neutro) sejam excluídas da análise.

Tabela de Frequência do Escore de Sentimentos

In [7]:
```
score_freq = df.score.value_counts().sort_index().to_frame('n')

score_freq
```

Out[7]:

	n
-5	1
-4	6
-3	103
-2	350
-1	258
1	198
2	465
3	194
4	9

Gráfico com a frequência do Escore de Sentimentos

In [8]:
```
score_freq.plot.bar(
    legend=False,
    figsize=(8, 4),
    grid=False,
    color='darkblue')
plt.xlabel('Escore dos Sentimentos')
plt.ylabel('Frequência das Palavras')
plt.title('Escore de Sentimento por Palavra', loc='left')
plt.title('Escore de Sentimentos por Palavras', loc='left')
plt.xticks(rotation=0);
```

Dividindo o texto em seções de 100 linhas

- Cálculo do escore de sentimento para cada seção.

In [9]:
```
score_acc = df.groupby(df['line'] // 100)\
    .score.mean()\
    .to_frame('score')\
    .rename_axis('section')

score_acc.head(8)
```

Out[9]:

section	score
0	0.392157
1	0.266667
2	0.266667
3	0.602151
4	0.661017
5	-0.442857
6	0.654321
7	0.470588

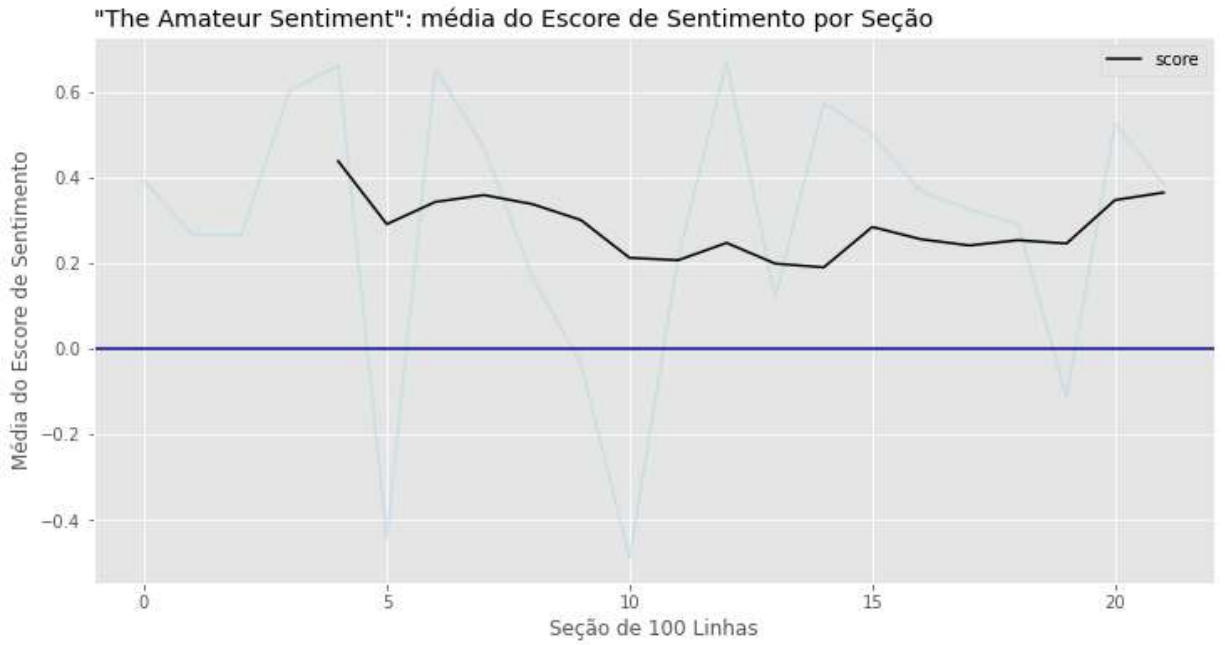
Gráfico do Arco da Narrativa com Escore de Sentimento por Seção

No gráfico abaixo:

- a **linha azul claro** mostra a média do escore de sentimento de cada seção (conforme números apresentados na tabela acima);
- a **linha preta** mostra a média móvel das seções;
- a **linha azul escuro** situa onde o 0.0 se encontra, separando o escore negativo do positivo.

In [10]:
```
ax = score_acc.plot.line(legend=False, figsize=(12, 6), grid=False, alpha=0.5, color
```

```
score_acc.rolling(10, min_periods=5).mean().plot.line(ax=ax, color='black')
plt.xlabel('Seção de 100 Linhas')
plt.ylabel('Média do Escore de Sentimento')
plt.title('"The Amateur Sentiment": média do Escore de Sentimento por Seção', loc='l
plt.axhline(0, color='darkblue')
plt.xticks(rotation=0);
```



A aplicação do **Escore de Sentimentos** na história de Stevenson mostra que, apesar de algumas seções terem predominância de palavras muito negativas (veja o movimento da linha azul claro no gráfico), a média das palavras com pesos entre +1 e +3 é bem alta e faz com que essa narrativa tenha uma conotação mais otimista do que pessimista do início ao fim (a média móvel, ou linha preta, mantem-se acima de 0.0).

Sumarizando esse projeto, o uso de algoritmos de Mineração de Textos na análise de "The Amateur Emigrant" foi fundamental para que a Cinetour Publishing decidisse publicar essa história (carro-chefe do livro "Essays of Travel", de Robert Louis Stevenson), visto que a editora procurava por uma narrativa de viagem que fosse real e predominantemente positiva para compor o seu portfolio de publicações.

Mineração de Textos na resolução de problemas de negócios: o uso de técnicas e algoritmos de mineração de textos permite que profissionais que trabalham com análise de conteúdo (como os editores que recebem muitos drafts de textos longos para lerem e avaliarem) possam ter uma ideia antecipada do material que têm em mãos antes mesmo de dedicarem-se a uma leitura completa dele, ganhando tempo por focar no que interessa mais à sua empresa e economizando dinheiro nesse processo.