

# Trabalho prático 4 - TP4

## Nuvens de palavras e análise de sentimentos

Sandro Ricardo De Souza

2025-05-23

### Descrição do trabalho

- Disciplina: Visualização de Dados
- Professor: Dr. Richard Godinez Tello
- Semestre: 2025/1

### Item 1

Valor: 5 pontos

Com base no Material Complementar 4, desenvolva um algoritmo para exibir uma nuvem de palavras com coloração personalizada (diferente da escala de cinzas), utilizando o formato (máscara) de sua preferência. A coloração pode seguir um dos seguintes estilos:

- Degradê de uma única cor: variação de intensidade (ex.: tons de azul),
- Degradê multicolorido: transição entre múltiplas cores (ex.: rosa → lilás → azul escuro).

Utilize dados baseados em feedbacks educacionais simulados ou reais, ou ainda escolher um conjunto de dados com o tema que desejar. Dica: caso prefira, utilize a figura de um livro colorido, que será disponibilizada no AVA, como base para aplicar o efeito de coloração.

### Item 2

Valor: 5 pontos

Crie uma base de dados simulada contendo textos curtos (como comentários, avaliações ou postagens) com polaridades variadas (positiva, negativa e neutra). Em seguida, desenvolva um algoritmo de análise de sentimentos utilizando um modelo pré-treinado baseado em Transformers (Hugging Face). Por fim, compare os resultados obtidos com pelo menos dois modelos

pré-treinados (biblioteca **Transformers** - *Hugging Face*), avaliando diferenças nas classificações através dos níveis de confiança (score).

## Resolução do item 1 - Nuvem de palavras

Não criamos uma base baseada em *feedbacks* educacionais. Ao invés disto, utilizamos a base disponibilizada no [Kaggle](#), em que o autor realizou um estudo para análise de sentimentos. O dataset foi utilizado aqui tanto para produzir a nuvem de palavras quanto para a análise de sentimentos.

O estudo utiliza uma base de dados focada na análise de sentimento de manchetes de notícias brasileiras. O objetivo principal é identificar a orientação política das agências de notícias em relação a figuras políticas, avaliando quais são mais ou menos favoráveis a determinados nomes.

A coleta das manchetes é realizada por um *scraper* que vasculha diversos sites de notícias brasileiras a cada trinta minutos. As manchetes são traduzidas para o inglês, e a análise de sentimento é então aplicada usando a API de Linguagem Natural do Google, tanto nas versões em português quanto em inglês. Os resultados são armazenados em um banco de dados que também inclui informações sobre o tempo de exibição online das manchetes.

Os sites de notícias pesquisados e utilizados na coleta de dados são:

- **UOL** (<https://www.uol.com.br/>)
- **Folha de São Paulo** (<https://www.folha.uol.com.br/>)
- **O Antagonista** (<https://www.oantagonista.com/>)
- **G1** (<https://g1.globo.com/>)
- **R7** (<https://www.r7.com/>)
- **Senso Incomum** (<https://www.sensoincomum.org>)
- **Terça Livre** (<https://www.tercalivre.com.br/>)

As palavras-chave monitoradas no processo de coleta incluem nomes de políticos como Bolsonaro, Moro, Lula, e outros presidentes, ex-presidentes, governadores, ministros, ex-ministros e possíveis candidatos a prefeito de São Paulo e do Rio de Janeiro. A análise de sentimento em inglês é recomendada devido à maior confiabilidade da API para este idioma.

```
# -*- coding: utf-8 -*-

import pandas as pd
from wordcloud import WordCloud
from nltk.corpus import stopwords
import matplotlib.pyplot as plt
import nltk
import numpy as np
```

```

from PIL import Image

# Baixar stopwords em português (executar só 1 vez)
nltk.download('stopwords')

# Usar stopwords em português
stopwords_pt = set(stopwords.words('portuguese'))

# Ler o dataframe
df = pd.read_csv(r'..\..\datasets\brazilian_headlines_sentiments.csv')

# Carregar a imagem que será usada como máscara
mascara = np.array(Image.open(r'../figuras/brasil4.jpg'))

# Concatenar o dataframe, na coluna de textos, em um único texto
text = " ".join(df['headlinePortuguese'])

# Gerar a nuvem com stopwords em português
wordcloud = WordCloud(
    width=2000,
    height=2000,
    random_state=42,
    background_color='white',
    mask=mascara,
    colormap='Reds',
    collocations=False,
    stopwords=stopwords_pt
).generate(text)

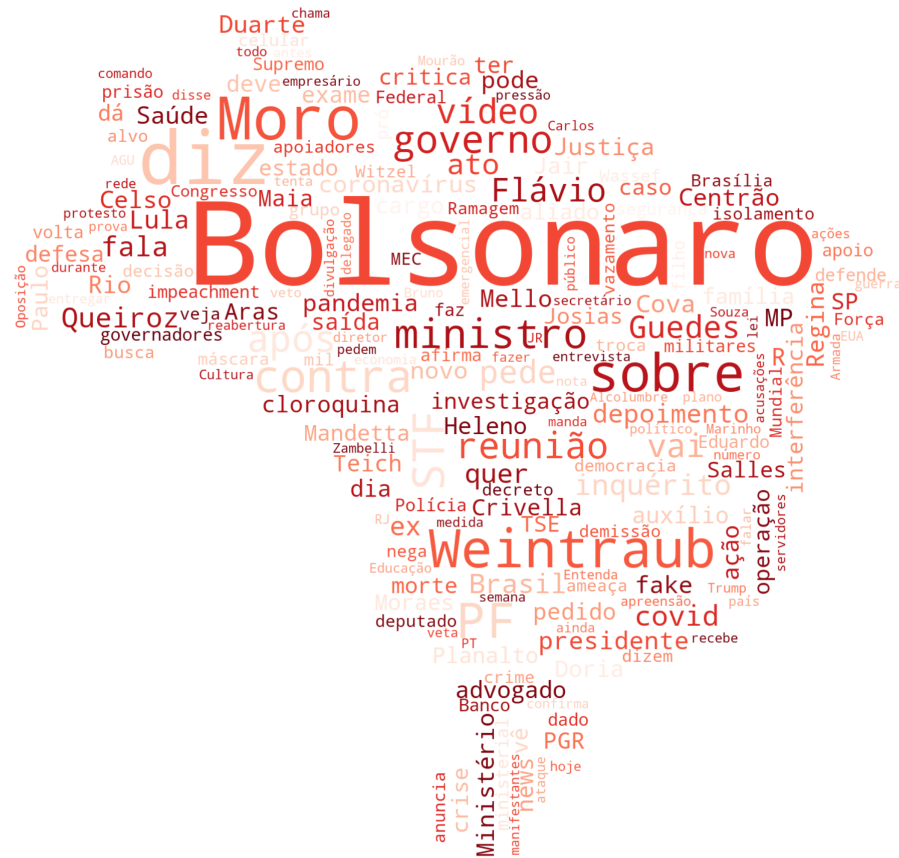
# Plotar a nuvem
plt.figure(figsize=(15, 10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
#plt.savefig(r'..\figuras\nuvem_palavras.png', bbox_inches='tight', dpi=300)

```

```

[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\sandr\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!

```



## Resolução do item 2 - Análise de sentimentos

Não criamos uma base simulada. Ao invés disto, utilizamos a base disponibilizada no Kaggle, já usada para o item 1.

1. Preparação dos Dados: Filtraremos o dataframe para incluir apenas as manchetes que mencionam “Bolsonaro”.

2. Seleção e Carregamento dos Modelos: Escolheremos dois modelos de análise de sentimentos em português da Hugging Face.
3. Aplicação dos Modelos: Executaremos a análise de sentimentos em cada manchete filtrada usando os dois modelos.
4. Análise dos Resultados: Agregaremos os resultados para determinar o sentimento geral em relação a Bolsonaro e o sentimento por cada website
5. Plotar resultados: Plotaremos a quantidade de sentimentos negativos, positivos e neutros, entre todos os veículos de notícias pesquisados. Também plotaremos o maior sentimento por website pesquisado.

```
# -*- coding: utf-8 -*-

import pandas as pd
from transformers import pipeline
import matplotlib.pyplot as plt
import seaborn as sns

# -----
# Funções
# -----

# Função para aplicar a análise de sentimento e extrair rótulo e score
def analisar_sentimento(texto, analisador):
    if analisador is None or not texto or pd.isna(texto):
        return None, None # Retorna None se o analisador não carregou ou o texto é inválido
    try:
        resultado = analisador(texto)
        # O resultado é uma lista com um dicionário, e.g., [{'label': 'POSITIVE', 'score': 0.9}]
        # Alguns modelos podem retornar labels como '1 star', '5 stars', etc.
        # É importante verificar o output específico do modelo ou mapear para POSITIVE/NEGATIVE/NEUTRAL
        label = resultado[0]['label']
        score = resultado[0]['score']

        # Padronização dos rótulos (exemplo, pode precisar de ajuste conforme os modelos)
        if label.upper() in ['POSITIVE', 'LABEL_2', 'Alegria', 'Positivo']: # Adicionar outros rótulos positivos aqui
            label_padronizado = 'Positivo'
        elif label.upper() in ['NEGATIVE', 'LABEL_0', 'Tristeza', 'Raiva', 'Medo', 'Negativo']: # Adicionar outros rótulos negativos aqui
            label_padronizado = 'Negativo'
        elif label.upper() in ['NEUTRAL', 'LABEL_1', 'Neutro']: # Adicionar outros rótulos neutros aqui
            label_padronizado = 'Neutro'
        else: # Se o rótulo não for reconhecido, manter o original para inspeção
            label_padronizado = label
        return label_padronizado, score
```

```

except Exception as e:
    print(f"Erro ao analisar texto: '{texto[:50]}...': {e}")
    return None, None

# -----
# Passo 1: Preparação dos Dados
# -----

# Carregar o dataframe
try:
    df = pd.read_csv(r'..\..\datasets\brazilian_headlines_sentiments.csv') # Ajuste o caminho
except FileNotFoundError:
    print("Arquivo 'brazilian_headlines_sentiments.csv' não encontrado. Por favor, verifique")

# Filtrar manchetes que contêm 'Bolsonaro' nas keywords
# Assegurando que a coluna 'keywords' é do tipo string antes de usar .str
df['keywords'] = df['keywords'].astype(str)
df_bolsonaro = df[df['keywords'].str.contains("'Bolsonaro'", case=False, na=False)].copy()

# Verificar o número de manchetes relacionadas a Bolsonaro
print(f"Número de manchetes relacionadas a Bolsonaro: {len(df_bolsonaro)}")
if not df_bolsonaro.empty:
    print("Amostra das manchetes filtradas:")
    print(df_bolsonaro[['website', 'keywords', 'headlinePortuguese']].head())
else:
    print("Nenhuma manchete relacionada a Bolsonaro foi encontrada com o critério de filtro.")

# -----
# Passo 2: Seleção e Carregamento dos Modelos
# -----

# Modelo 1
modelo_1_nome = "turing-usp/FinBertPTBR"
analizador_sentimento_1 = None # Inicializa como None
try:
    analizador_sentimento_1 = pipeline("sentiment-analysis", model=modelo_1_nome)
    print(f"Modelo 1 ({modelo_1_nome}) carregado com sucesso.")
except Exception as e:
    print(f"Erro ao carregar modelo 1 ({modelo_1_nome}): {e}")

```

```

    print("Verifique se o nome do modelo está correto, se há conexão com a internet e se as c

# Modelo 2
modelo_2_nome = "neuralmind/bert-base-portuguese-cased"
analizador_sentimento_2 = None # Inicializa como None
try:
    analisador_sentimento_2 = pipeline("sentiment-analysis", model=modelo_2_nome)
    print(f"Modelo 2 ({modelo_2_nome}) carregado com sucesso.")
except Exception as e:
    print(f"Erro ao carregar modelo 2 ({modelo_2_nome}): {e}")
    print("Verifique se o nome do modelo está correto, se há conexão com a internet e se as c

# -----
# Passo 3: Aplicação dos Modelos às Manchetes
# (Já definida a função acima)
# -----

# Aplicar Modelo 1 (se carregado)
if analisador_sentimento_1:
    print(f"\nAplicando Modelo 1 ({modelo_1_nome})...")
    # Usar .loc para evitar SettingWithCopyWarning
    df_bolsonaro.loc[:, 'sentimento_modelo1'] = None
    df_bolsonaro.loc[:, 'score_modelo1'] = None

    for index, row in df_bolsonaro.iterrows():
        label, score = analisar_sentimento(row['headlinePortuguese'], analisador_sentimento_1)
        df_bolsonaro.loc[index, 'sentimento_modelo1'] = label
        df_bolsonaro.loc[index, 'score_modelo1'] = score
    print("Modelo 1 aplicado.")
    print(df_bolsonaro[['headlinePortuguese', 'sentimento_modelo1', 'score_modelo1']].head())

# Aplicar Modelo 2 (se carregado)
if analisador_sentimento_2:
    print(f"\nAplicando Modelo 2 ({modelo_2_nome})...")
    df_bolsonaro.loc[:, 'sentimento_modelo2'] = None
    df_bolsonaro.loc[:, 'score_modelo2'] = None

    for index, row in df_bolsonaro.iterrows():
        label, score = analisar_sentimento(row['headlinePortuguese'], analisador_sentimento_2)
        df_bolsonaro.loc[index, 'sentimento_modelo2'] = label
        df_bolsonaro.loc[index, 'score_modelo2'] = score

```

```

    print("Modelo 2 aplicado.")
    print(df_bolsonaro[['headlinePortuguese', 'sentimento_modelo2', 'score_modelo2']].head())

# -----
# Passo 4: Análise dos Resultados
# -----

# Para esta etapa, vamos assumir que pelo menos um modelo foi carregado e aplicado.
# Se ambos foram carregados, podemos escolher um para a análise principal ou comparar.
# Vamos usar o Modelo 1 como exemplo, se disponível, caso contrário o Modelo 2.

coluna_sentimento_ativa = None
if 'sentimento_modelo1' in df_bolsonaro.columns and analisador_sentimento_1:
    coluna_sentimento_ativa = 'sentimento_modelo1'
    print(f"\nAnalisando resultados com base no Modelo 1 ({modelo_1_nome}).")
elif 'sentimento_modelo2' in df_bolsonaro.columns and analisador_sentimento_2:
    coluna_sentimento_ativa = 'sentimento_modelo2'
    print(f"\nAnalisando resultados com base no Modelo 2 ({modelo_2_nome}).")
else:
    print("\nNenhum modelo de sentimento foi aplicado com sucesso. Não é possível analisar os")

if coluna_sentimento_ativa:
    # Pergunta 1: Qual é o sentimento geral dos jornalistas para o presidente Bolsonaro?
    print("\n--- Sentimento Geral sobre Bolsonaro ---")
    sentimento_geral_bolsonaro = df_bolsonaro[coluna_sentimento_ativa].value_counts(normalize=True)
    print(sentimento_geral_bolsonaro)
    print(f"Contagem total: {df_bolsonaro[coluna_sentimento_ativa].value_counts().to_dict()}")

    # Pergunta 2: Qual é o sentimento por site?
    print("\n--- Sentimento sobre Bolsonaro por Site ---")
    sentimento_por_site = df_bolsonaro.groupby('website')[coluna_sentimento_ativa].value_counts(normalize=True)
    # Para melhor visualização, podemos adicionar a contagem total por site também
    contagem_por_site = df_bolsonaro.groupby('website')[coluna_sentimento_ativa].count()
    sentimento_por_site_com_contagem = pd.concat([sentimento_por_site, contagem_por_site.rename('contagem')], axis=1)

    print(sentimento_por_site_com_contagem)

# Se ambos os modelos foram aplicados, você pode querer comparar as contagens
if 'sentimento_modelo1' in df_bolsonaro.columns and 'sentimento_modelo2' in df_bolsonaro.columns:
    print("\n--- Comparação entre Modelos (Contagem Geral) ---")
    print("Modelo 1:")

```



```

print(df_bolsonaro['sentimento_modelo1'].value_counts())
print("\nModelo 2:")
print(df_bolsonaro['sentimento_modelo2'].value_counts())

# Comparação de concordância (exemplo simples)
concordancia = (df_bolsonaro['sentimento_modelo1'] == df_bolsonaro['sentimento_modelo2'])
print(f"\nPercentual de concordância entre Modelo 1 e Modelo 2: {concordancia:.2f}%")

# Casos de discordância para inspeção (primeiros 5)
discordantes = df_bolsonaro[df_bolsonaro['sentimento_modelo1'] != df_bolsonaro['sentimento_modelo2']]
if not discordantes.empty:
    print("\nExemplos de manchetes com sentimentos discordantes entre os modelos:")
    print(discordantes[['headlinePortuguese', 'sentimento_modelo1', 'score_modelo1', 'score_modelo2']])
else:
    print("\nNão houve discordância entre os modelos ou apenas um foi executado.")

# -----
# Passo 5: Plotar resultados
# -----

# Configurar um estilo visual mais agradável para os gráficos
sns.set_style("whitegrid")
plt.rcParams['font.family'] = 'sans-serif' # Garante uma fonte padrão consistente

if coluna_sentimento_ativa:
    # --- Plotagem dos Resultados ---

    # 1. Gráfico Geral de Sentimentos sobre Bolsonaro
    print("\n--- Visualização: Sentimento Geral sobre Bolsonaro ---")
    plt.figure(figsize=(8, 6))
    # Usar value_counts() diretamente para obter as contagens por categoria
    sentimento_counts = df_bolsonaro[coluna_sentimento_ativa].value_counts()

    # Definir uma ordem preferencial para as categorias e cores consistentes
    ordem_sentimentos = ['Positivo', 'Neutro', 'Negativo']
    cores_sentimentos = {'Positivo': 'green', 'Neutro': 'lightgray', 'Negativo': 'red'}

    # Filtrar para categorias presentes e reordenar
    sentimento_counts = sentimento_counts.reindex(ordem_sentimentos).dropna()

    sns.barplot(x=sentimento_counts.index, y=sentimento_counts.values, palette=[cores_sentimentos])

```

```

plt.title(f'Distribuição Geral de Sentimentos sobre Bolsonaro\n(Modelo: {coluna_sentimento_geral})')
plt.ylabel('Número de Manchetes', fontsize=12)
plt.xlabel('Sentimento', fontsize=12)
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)
# Adicionar os valores no topo das barras
for i, v in enumerate(sentimento_counts.values):
    plt.text(i, v + (sentimento_counts.values.max()*0.01), str(v), color='black', ha='center')
plt.tight_layout() # Ajusta o layout para evitar sobreposição
plt.show()

# 2. Gráfico de Sentimento Predominante por Site (mostrando a distribuição percentual)
print("\n--- Visualização: Distribuição Percentual de Sentimentos sobre Bolsonaro por Site")

# Calcular a distribuição percentual de sentimentos por site
sentimento_por_site_percentual = df_bolsonaro.groupby('website')[coluna_sentimento_ativa].value_counts().unstack()

# Reordenar colunas para consistência (Positivo, Neutro, Negativo)
sentimento_por_site_percentual = sentimento_por_site_percentual.reindex(columns=ordem_sentimentos)

if not sentimento_por_site_percentual.empty:
    sentimento_por_site_percentual.plot(kind='bar', stacked=True, figsize=(14, 8),
                                         color=[cores_sentimentos.get(s, 'blue') for s in ordem_sentimentos],
    plt.title(f'Distribuição Percentual de Sentimentos sobre Bolsonaro por Site\n(Modelo: {coluna_sentimento_ativa})')
    plt.ylabel('Percentual de Manchetes (%)', fontsize=12)
    plt.xlabel('Website', fontsize=12)
    plt.xticks(rotation=45, ha='right', fontsize=10)
    plt.yticks(fontsize=10)
    plt.legend(title='Sentimento', bbox_to_anchor=(1.02, 1), loc='upper left', fontsize=10)
    plt.tight_layout(rect=[0, 0, 0.85, 1]) # Ajustar layout para dar espaço à legenda
    plt.show()

# Para mostrar o "maior sentimento por site" de forma tabular, podemos fazer:
maior_sentimento_por_site = sentimento_por_site_percentual.idxmax(axis=1)
contagem_total_por_site = df_bolsonaro.groupby('website')[coluna_sentimento_ativa].count()

resultado_maior_sentimento = pd.DataFrame({
    'Maior_Sentimento': maior_sentimento_por_site,
    'Percentual_Maior_Sentimento': sentimento_por_site_percentual.max(axis=1).round(2),
    'Total_Manchetes_Bolsonaro': contagem_total_por_site
})
print("\n--- Sentimento Predominante sobre Bolsonaro por Site (com base no maior percentual)")

```

```

        print(resultado_maior_sentimento.sort_values(by='Total_Manchetes_Bolsonaro', ascending=False))

    else:
        print("Não há dados suficientes para gerar o gráfico de sentimento por site.")

else:
    print("\nPlotagem não realizada pois nenhuma coluna de sentimento ativa foi definida (problema de configuração)")

```

Número de manchetes relacionadas a Bolsonaro: 6425

Amostra das manchetes filtradas:

|   | website       | keywords \            |
|---|---------------|-----------------------|
| 0 | UOL           | ['Bolsonaro']         |
| 1 | O Antagonista | ['Bolsonaro']         |
| 2 | G1            | ['Bolsonaro', 'Moro'] |
| 3 | Folha de SP   | ['Bolsonaro']         |
| 6 | UOL           | ['Bolsonaro']         |

|   | headlinePortuguese                                |
|---|---|
| 0 | Bolsonaro se desculpa após mandar repórter 'ca... |
| 1 | "Bolsonaro, faça um exame de consciência"         |
| 2 | Moro dá 'unfollow' em Bolsonaro em rede social... |
| 3 | Inquérito que investiga Bolsonaro adota 'sabat... |
| 6 | Secretário troca comandante da Rota crítico a ... |

WARNING:tensorflow:From C:\Users\sandr\miniconda3\Lib\site-packages\tf\_keras\src\losses.py:207: tf.nn.softmax\_cross\_entropy\_with\_logits\_v2 is deprecated and will be removed in a future version. Use tf.nn.softmax\_cross\_entropy\_with\_logits instead.

Device set to use cpu

Modelo 1 (turing-usp/FinBertPTBR) carregado com sucesso.

Some weights of BertForSequenceClassification were not initialized from the model checkpoint turing-usp/FinBertPTBR. You should probably TRAIN this model on a down-stream task to be able to use it for predictions.

Device set to use cpu

Modelo 2 (neuralmind/bert-base-portuguese-cased) carregado com sucesso.

Aplicando Modelo 1 (turing-usp/FinBertPTBR)...

Modelo 1 aplicado.

|   | headlinePortuguese                                | sentimento_modelo1 \ |
|---|---|----------------------|
| 0 | Bolsonaro se desculpa após mandar repórter 'ca... | Negativo             |
| 1 | "Bolsonaro, faça um exame de consciência"         | Neutro               |

|   |   |          |
|---|---|----------|
| 2 | Moro dá 'unfollow' em Bolsonaro em rede social... | Neutro   |
| 3 | Inquérito que investiga Bolsonaro adota 'sabat... | Positivo |
| 6 | Secretário troca comandante da Rota crítico a ... | Positivo |

```

score_modelo1
0      0.702547
1      0.507402
2      0.478755
3      0.45196
6      0.55918

```

Aplicando Modelo 2 (neuralmind/bert-base-portuguese-cased)...

Modelo 2 aplicado.

|   | headlinePortuguese                                | sentimento_modelo2 | \ |
|---|---|--------------------|---|
| 0 | Bolsonaro se desculpa após mandar repórter 'ca... | Neutro             |   |
| 1 | "Bolsonaro, faça um exame de consciência"         | Neutro             |   |
| 2 | Moro dá 'unfollow' em Bolsonaro em rede social... | Neutro             |   |
| 3 | Inquérito que investiga Bolsonaro adota 'sabat... | Neutro             |   |
| 6 | Secretário troca comandante da Rota crítico a ... | Neutro             |   |

```

score_modelo2
0      0.684839
1      0.67154
2      0.651763
3      0.66579
6      0.647706

```

Analisando resultados com base no Modelo 1 (turing-usp/FinBertPTBR).

--- Sentimento Geral sobre Bolsonaro ---

```

sentimento_modelo1
Negativo    45.307393
Neutro      30.754864
Positivo    23.937743
Name: proportion, dtype: float64
Contagem total: {'Negativo': 2911, 'Neutro': 1976, 'Positivo': 1538}

```

--- Sentimento sobre Bolsonaro por Site ---

|             | Negativo  | Neutro    | Positivo  | Total_Manchetes_Bolsonaro |
|-------------|-----------|-----------|-----------|---------------------------|
| website     |           |           |           |                           |
| CNN Brasil  | 0.000000  | 66.666667 | 33.333333 | 3                         |
| Folha de SP | 50.924918 | 24.591948 | 24.483134 | 919                       |
| G1          | 43.656716 | 33.283582 | 23.059701 | 1340                      |

|               |           |           |           |      |
|---------------|-----------|-----------|-----------|------|
| Nexo          | 50.000000 | 50.000000 | 0.000000  | 2    |
| 0 Antagonista | 37.588652 | 40.661939 | 21.749409 | 1269 |
| R7            | 35.772358 | 29.471545 | 34.756098 | 492  |
| Senso Incomum | 66.666667 | 33.333333 | 0.000000  | 3    |
| Terça Livre   | 18.750000 | 42.187500 | 39.062500 | 64   |
| UOL           | 51.007287 | 26.232319 | 22.760394 | 2333 |

--- Comparação entre Modelos (Contagem Geral) ---

Modelo 1:

sentimento\_modelo1

Negativo 2911

Neutro 1976

Positivo 1538

Name: count, dtype: int64

Modelo 2:

sentimento\_modelo2

Neutro 6030

Negativo 395

Name: count, dtype: int64

Percentual de concordância entre Modelo 1 e Modelo 2: 33.34%

Exemplos de manchetes com sentimentos discordantes entre os modelos:

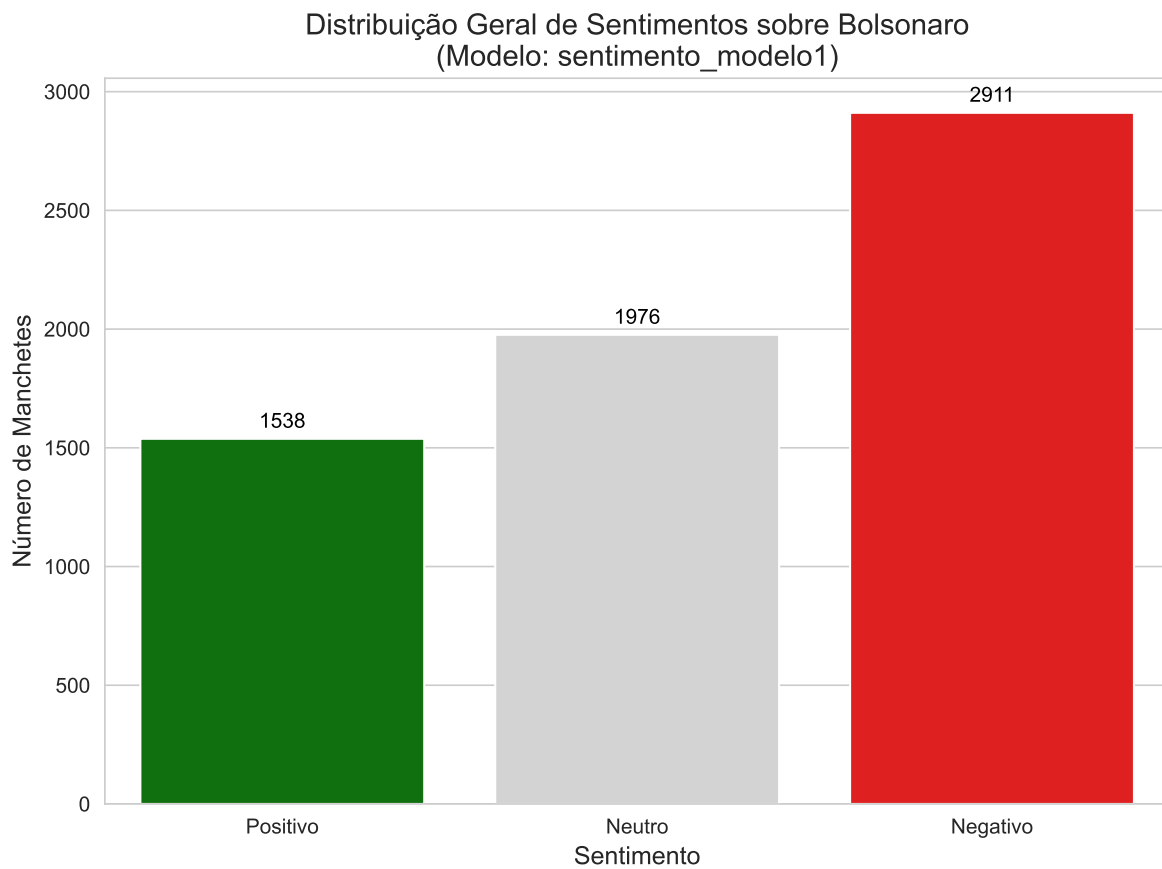
|   | headlinePortuguese                                | sentimento_modelo1 | \ |
|---|---|--------------------|---|
| 0 | Bolsonaro se desculpa após mandar repórter 'ca... | Negativo           |   |
| 3 | Inquérito que investiga Bolsonaro adota 'sabat... | Positivo           |   |
| 6 | Secretário troca comandante da Rota crítico a ... | Positivo           |   |
| 8 | Promessa feita por Jair Bolsonaro de cortar 30... | Positivo           |   |
| 9 | Bolsonaro critica participação de Weintraub em... | Negativo           |   |

|   | score_modelo1 | sentimento_modelo2 | score_modelo2 |
|---|---------------|--------------------|---------------|
| 0 | 0.702547      | Neutro             | 0.684839      |
| 3 | 0.45196       | Neutro             | 0.66579       |
| 6 | 0.55918       | Neutro             | 0.647706      |
| 8 | 0.495275      | Neutro             | 0.564153      |
| 9 | 0.554796      | Neutro             | 0.628477      |

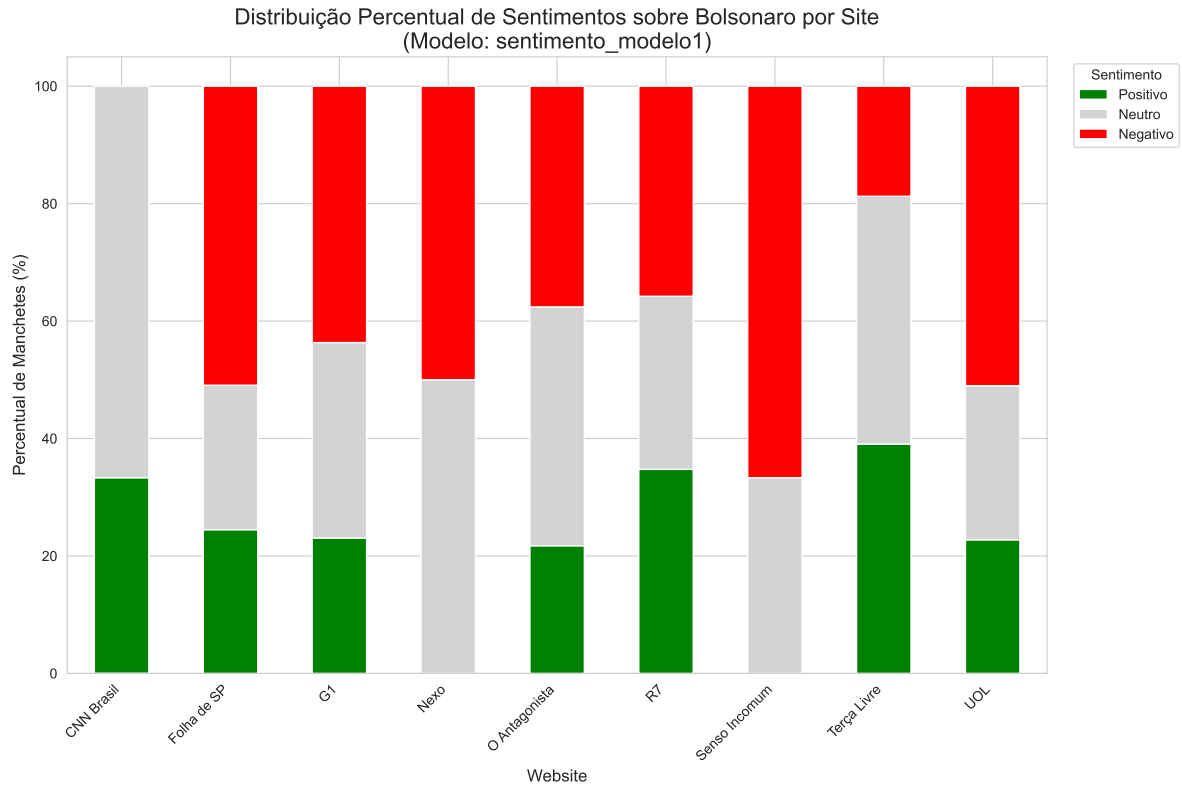
--- Visualização: Sentimento Geral sobre Bolsonaro ---

C:\Users\sandr\AppData\Local\Temp\ipykernel\_15364\2527189449.py:201: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign



--- Visualização: Distribuição Percentual de Sentimentos sobre Bolsonaro por Site ---



--- Sentimento Predominante sobre Bolsonaro por Site (com base no maior percentual) ---

| website       | Maior_Sentimento | Percentual_Maior_Sentimento \ |
|---------------|------------------|-------------------------------|
| UOL           | Negativo         | 51.01                         |
| G1            | Negativo         | 43.66                         |
| O Antagonista | Neutro           | 40.66                         |
| Folha de SP   | Negativo         | 50.92                         |
| R7            | Negativo         | 35.77                         |
| Terça Livre   | Neutro           | 42.19                         |
| CNN Brasil    | Neutro           | 66.67                         |
| Senso Incomum | Negativo         | 66.67                         |
| Nexo          | Neutro           | 50.00                         |

Total\_Manchetes\_Bolsonaro

| website       |      |
|---------------|------|
| UOL           | 2333 |
| G1            | 1340 |
| O Antagonista | 1269 |

|               |     |
|---------------|-----|
| Folha de SP   | 919 |
| R7            | 492 |
| Terça Livre   | 64  |
| CNN Brasil    | 3   |
| Senso Incomum | 3   |
| Nexo          | 2   |