

Deteção de posicionamento através dos “tweets” para as eleições presidenciais brasileiras de 2022 no segundo turno.

Diogo P. de F. Silva

Centro de Informática - Universidade Federal de Pernambuco, Av. Jorn. Aníbal Fernandes,
s/n - Cidade Universitária, Recife - PE, 50740-560

1. Apresentação

1.1 Contexto

De acordo com a pesquisa “TIC Domicílios 2020 (Edição COVID-19 - Metodologia Adaptada)”, o Brasil tem 81% das pessoas como usuárias de internet. Naturalmente, as discussões e interações que outrora aconteciam no mundo físico passaram a acontecer na internet, e os assuntos políticos estão inclusos. As redes sociais, que acompanharam o crescente uso da internet, tornaram-se um dos principais palcos de discussões políticas desde a última década. De acordo com o portal G1¹, a eleição brasileira de 2014 gerou 40 milhões de “Tweets” relacionados.

Dentre as principais plataformas que proporcionam o debate público, destaca-se o Twitter, tendo cerca de 319 milhões de usuários ativos, o que o posiciona como umas das redes sociais mais utilizadas do mundo [1] e é uma ótima fonte de entendimento das opiniões de internautas [2].

Em outubro de 2022, aconteceu o primeiro turno das eleições presidenciais brasileiras e, como é de costume, os debates políticos têm a tendência de serem mais frequentes e mais focados nos candidatos ao pleito.

1.2 Objetivo

O objetivo deste trabalho é analisar os sentimentos de “tweets” da população brasileira a respeito dos posicionamentos políticos, isto é, julgamentos morais e comentários de cunho político-ideológico, dos candidatos à presidência do Brasil neste segundo turno. Tal escolha se deu com base na divergência das pesquisas com a intenção de voto, realizando uma análise do tweet podemos identificar a opinião pública da rede e a sua intenção de voto, para que a assessoria do candidato possa mudar a estratégia durante a eleição para promover o candidato como também, identificar o possível presidente pelo comportamento na rede.

Dado um tweet sobre as eleições presidenciais em relação ao 2º turno de 2022, o modelo deve identificar se aquele tweet é a favor do candidato Bolsonaro, Lula ou Neutro, caso o candidato se posicione de forma neutro deve realizar críticas de forma a não favorecer a nenhum candidato ou uma ideia de dúvida.

1.3 Justificativa

1.3.1 Trabalhos relacionados de análise de sentimento

Em [4], Kušen e Strembeck utilizaram o algoritmo SentiStrength e o léxico de palavras relacionadas a emoções NRC para realizar uma análise de sentimentos a partir de “tweets” sobre a eleição presidencial austríaca de 2016. Especificamente, analisaram (1) características temporais dos “tweets” publicados pelos candidatos, (2) quais os principais

estilos de engajamento e (3) se existe evidência dos diferentes tipos de campanha (negativas, positivas ou neutras).

No contexto das eleições presidenciais norte-americanas de 2012 [5], Wang et al. propuseram uma análise em tempo real do sentimento do público. Sua metodologia não fez distinção entre temas das publicações dos usuários. Foi adotada uma pequena janela de tempo que permite detectar variações de sentimentos relacionados a determinado político e relacionar com determinado assunto que esteja sendo comentado no momento.

A formação de opinião política no Twitter diante do período das eleições foi um dos assuntos abordados em [6]. Usuários influentes são capazes de direcionar e formar novas opiniões e processos sobre quaisquer decisões. Nesse contexto, foram aplicadas técnicas de regressão a fim de entender se o sentimento expressado de diferentes formas nos conteúdos de mídia, seja ele positivo ou negativo, possui correlação com a velocidade e a quantidade de retweets.

Agora se referindo a artigos sobre eleições presidenciais brasileiras, o artigo produzido por André [9], demonstra a análise de sentimento durante a eleição presidencial de 2018, utilizando o SVM para identificar o posicionamento político do candidato.

1.3.2 Modelo Escolhido - BERT

Diferentemente dos outros artigos das eleições presidenciais brasileiras, como os citados na seção anterior, foi utilizado uma nova metodologia de estado da arte para identificar o posicionamento político do eleitorado e assim conseguir obter melhores resultados para a análise de sentimento. Através dos trabalhos relacionados podemos identificar o uso de diversos modelos, conjunto de dados a respeito sobre análise de sentimento, e os comentários relacionados a seus atores.

O BERT (Bidirectional Encoder Representations from Transformers)[7], algoritmo publicado em 2018 como uma nova técnica de aprendizagem de máquina, é considerado o estado da arte quando se trata de processamento de linguagem natural, até o ano da publicação deste trabalho, por ter uma performance superior aos algoritmos concorrentes, como demonstrado por Wang [8].

Dado que BERT processa sentenças da língua inglesa, utilizaremos o modelo Bertimbau, que é o BERT pré-treinado com sentenças da língua portuguesa. A versão escolhida foi a BERT Large do Bertimbau [3] com um total de 12 camadas, que incluem um total de 335 milhões de parâmetros, intitulado no artigo como neuralmind/bert-large-portuguese-cased, onde será usado transfer learning com o nosso dataset de dados rotulados.

2. Metodologia

2.1 Base de Dados

2.1.1 Coleta de dados

Inicialmente, foi necessário criar o conjunto de dados a ser analisado, os candidatos à presidência para segundo turno são diferentes se comparados a da eleição passada de 2018, além disso o atual presidente possui um histórico que é usado tanto a favor como contra ele, o que resulta em tweets com o contexto totalmente diferente e criando a necessidade de novos dados para serem trabalhados. Para isso, definimos as palavras-chave que deveriam estar presentes no “tweet” a fim de torná-lo parte do conjunto de dados. A proposta de

palavras-chave foram as seguintes: #Lula13, #Bolsonaro22, #BrasilDaEsperança, #Eleição2022, #Nulo, essas palavras-chave foram baseadas no artigo sobre detecção de posicionamento e discurso de ódio que também utilizou Bert mas para as eleições presidenciais americanas de 2020 [10] Lara Grimminger.

Além disso, realizamos uma limitação temporal para busca, considerando “tweets” realizados após a eleição do primeiro turno, para conseguir apenas tweets do segundo turno e exclusivamente dos candidatos, outro critério utilizado na extração foi a exclusão de “retweets”. Para obtenção dos “tweets” baseada nos critérios explicados anteriormente, fez-se uso da APIv2 do Twitter, que é disponibilizada de forma pública e gratuita. A extração com a aplicação dos critérios retornou um total de 628 “tweets”. Por conta das limitações da API quando realizados, uma aquisição grande é limitada, parte dos dados foi incrementado de forma manual e não extraído de forma única pelo código.

2.1.2 Processamento dos dados

Em posse dos “tweets” extraídos, percebemos que seria necessário realizar a exclusão de parte dos "tweets" por terem uma ou mais das seguintes condições: (1) mensagens curtas com apenas mencionando o nome do candidato, (2) não serem publicados a partir de uma conta de um usuário pessoa física (isto é, foram publicados por um portal de notícias, jornais, páginas de fofoca ou outras contas similares), (3) mensagens ambíguas, (4) acompanhadas de imagens ou vídeos que juntos complementam o sentido da sua opinião (ex: minha opinião, e logo em seguida um vídeo de propaganda política), (5) Tweets que claramente são spams, que possui # de vários tópicos em alta sem nenhuma mensagem, (6) Foi removido no processamento, os emojis nos comentários, a razão disso é que um único emoji pode mudar completamente o sentido do tweet, dessa forma uma base de dados mais limitada como o nosso poderia trazer resultados insatisfatório, (7) Limitamos que os tweets tenha mensagens com tamanho máximo de 160 caracteres e tamanho mínimo de 60, isso é importante para o modelo e foi baseado no gráfico demonstrado no código colab, (8) parte dos tweets que possuía um “@usuario” mencionando foi removida e outras foram mantidas usando uma função aleatória com proporção de 30% para remover.

Além disso, realizamos também a simplificação de alguns termos com grandes repetições de caracteres. Ao final do processo, o conjunto de dados podem ser visualizados no GitHub¹, bem como todos os protocolos utilizados na fase de processamento dos dados. Na apresentação do slide, foram inseridas prints de alguns dados removidos por conta dos problemas descritos anteriormente.

2.1.3 Tarefa de rotulação

Dos dados extraídos e rotulados temos um total de 239 tweets que possui sentimento em favor de Lula, 241 tweets com sentimentos a favor de Bolsonaro e um total de 148 tweets com sentimento Neutro.

¹ Disponível em: <https://github.com/DiogoPedro/Bert-Transformers-IA>

A rotulação dos dados foi realizada de forma paralela ao processamento dos dados, isto é, a varredura realizada por mim neste trabalho teve por objetivo remover "tweets" indesejados e rotular os 628 tweets para o treinamento e teste do modelo.

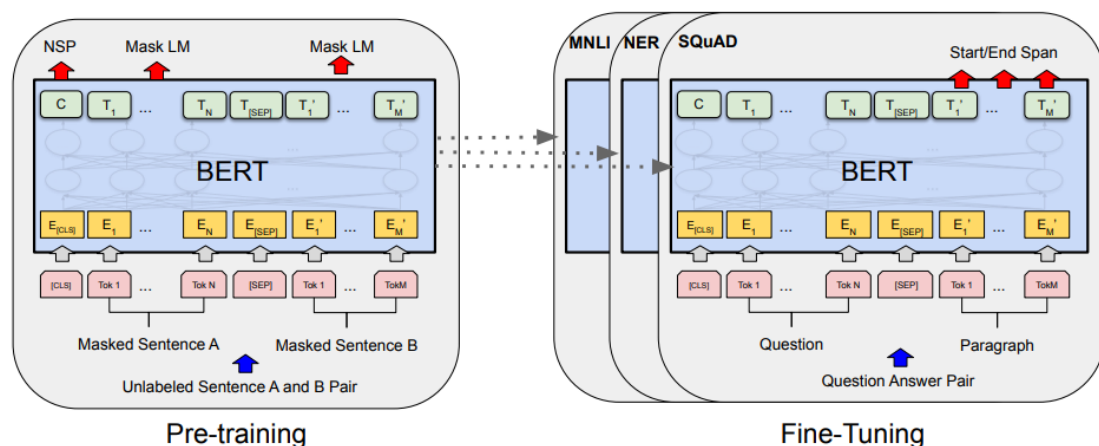
Na etapa de classificar o sentimento, dado o texto ele é classificado entre uma das três seguintes opções: Favor de Lula, quando o texto argumenta em favor do candidato Lula a coluna aparece como 1; Favor de Bolsonaro, quando o texto argumenta em favor do candidato Bolsonaro a coluna aparece como 1; Neutro, quando o texto não declara qualquer tipo de apoio aos candidatos citados anteriormente, caso algum usuário apenas faça crítica a um determinado político de forma explícita mas não menciona que é favor a certo candidato nem faz uma sugestão disto, seu texto será analisado e classificado como neutro, onde a coluna aparece com o valor 1.

2.2. Modelo Utilizado

O como dito anteriormente, o modelo escolhido foi o BERT no artigo original em que foi publicado, já é citado como novo estado da arte e por esse motivo foi escolhido como diferencial em relação a trabalhos em português relacionados a detecção de posicionamento na eleição presidencial, o artigo de Mickel [11] demonstra um estudo detalhado das competições SemEval de 2015 a 2016 e demonstra os resultados que impulsionaram o Bert a ser considerado o estado da arte, com a diferenciação em realizar previsão de contexto em uma próxima frase. O Bert superou concorrentes no mesmo período que foi publicado, o caso do ELMO (Peters et al.,2018), que utilizava a contextualização de palavras, derivado de uma LSTM.

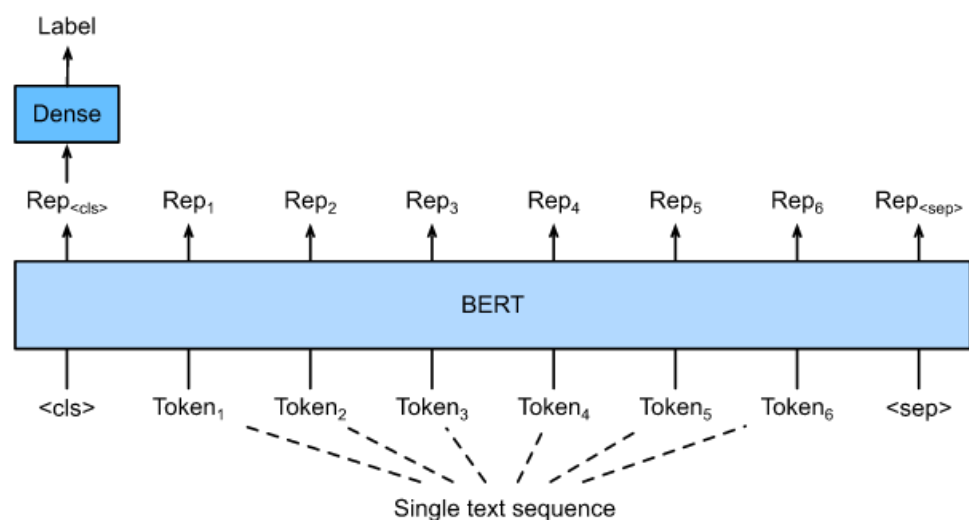
Existem duas estratégias existentes para aplicar representações de linguagem pré-treinadas a tarefas de downstream: elas são feature-based e fine-tuning, o Bert utiliza a representação com fine-tuning, isso ocorre porque a utilização de feature-based é unidirecional, o modelo Bert utiliza a idéia de que essa restrição de análise unidirecional não são ideais para tarefas em nível de sentença e podem ser muito prejudiciais ao aplicar abordagens baseadas em ajuste fino para tarefas em nível de token, como resposta a perguntas, onde é crucial incorporar o contexto de ambas as direções.

O BERT: Bidirectional Encoder Representations from Transformers, alivia a restrição de unidirecionalidade mencionada anteriormente usando um objetivo de pré-treinamento de “modelo de linguagem mascarado” (MLM), inspirado na tarefa Cloze. O modelo de idioma mascarado mascara aleatoriamente alguns dos tokens da entrada e o objetivo é prever o ID do vocabulário original da palavra mascarada com base apenas em seu contexto. Ao contrário do pré-treinamento do modelo de linguagem da esquerda para a direita, o objetivo de MLM permite a representação diferenciada do contexto esquerdo e direito, o que nos permite pré-treinar um transformador bidirecional profundo. Além do modelo de linguagem mascarada, também usamos uma tarefa de “previsão da próxima frase” que pré-treina conjuntamente representações de pares de texto.

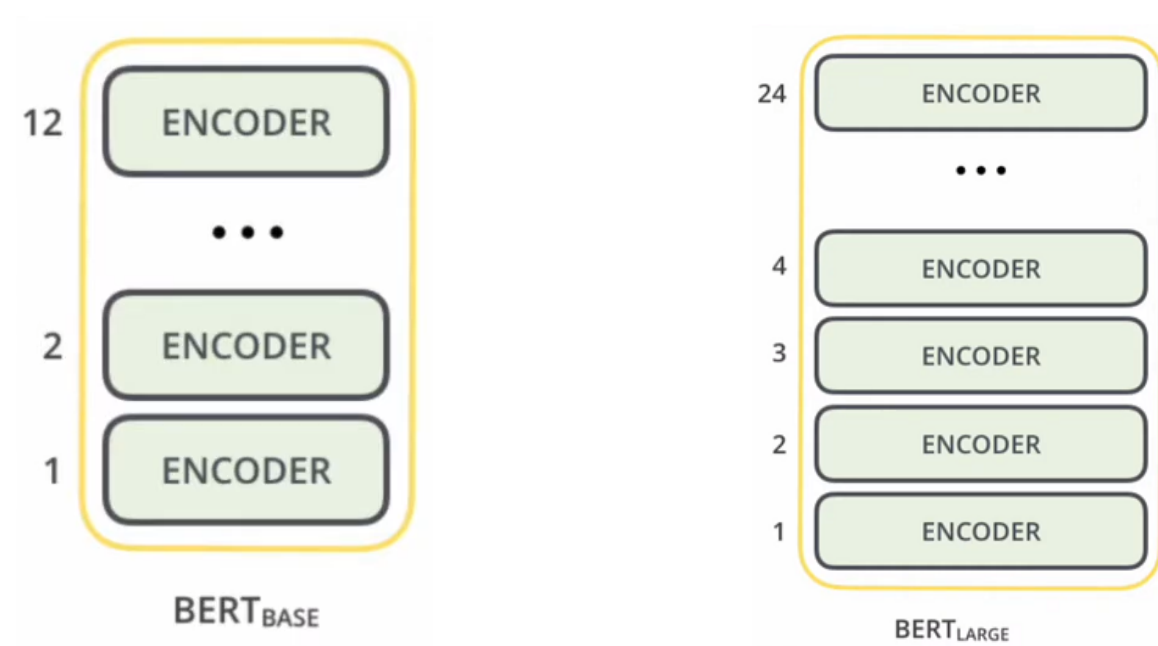


Bert realiza a tokenização do texto, atribuindo um valor para cada palavra, utilizando um token especial para representar o inicial da palavra, vazio e o fim, sendo limitado em certo tamanho. Uma vez definida todos os tokens, o processo é realizado da seguinte forma, dado o modelo treinado, os tokens iram entrar na camada encoder, será analisado o contexto de cada token é uma representação de saída que irá alimentar a entrada de outro token na camada posterior do encoder, na última camada, o token produzido pela CLS, que é a token que sugeri início da frase, vai trazer todo o contexto e o significado naquela frase, que vai servir para que possamos classificar através da saída.

Um dos grandes diferenciais do Bert em relação aos modelos anteriores é do Embeddings Dinâmicos, enquanto modelos anteriores após o treinamento dos dados, criavam uma relação entre os dados e através de alguma média, criava um **Embedding Fixo** para representar o seu valor, no Bert isso não ocorre, cada texto é analisado o seu contexto, possuindo valores de **Embedding Dinâmicos**, isso significa que mesmo frases cujo sentido seja muito próximo o Bert vai armazenar de forma diferente, já que o contexto não é igual, isso gafrantiu uma melhor representação dos dados.



A arquitetura do Bert pode incluir um total de 12 encoders para o caso do Bert Base, ou um total de 24 encoders no que é chamado de Bert Large. A diferença de resultados para a arquitetura Base para Large é de 2 pontos percentuais ou menos, apresentado no artigo original em diversos dataset, sendo a maior diferença encontrado no RTE (Recognizing Textual Entailment is a binary entailment task similar to MNLI, but with much less training data) com 3,7 pontos percentuais.



O modelo Bertimbau é uma versão do Bert pré-treinados com a língua portuguesa, produzido a partir de um banco de dados em português produzido através de um Crawler com mais de 60 milhões de páginas, sendo destas um total de 3,5 milhões selecionados, com um total de mais de 2,7 bilhões de tokens.

Model	Arch.	#Layers	#Params
neuralmind/bert-base-portuguese-cased	BERT-Base	12	110M
neuralmind/bert-large-portuguese-cased	BERT-Large	24	335M

Como as publicações do nossos tweets são em português foi utilizado o modelo Bertimbau para trabalhar e fazer a análise de sentimentos do textos, informações sobre o modelo escolhido a seguir.

2.3 Experimentos

As bibliotecas utilizadas e algumas versões da implementação utilizada:

```
import transformers
from transformers import BertModel, BertTokenizer, AdamW, get_linear_schedule_with_warmup
import pandas as pd
import numpy as np
import warnings
import seaborn as sns
import matplotlib.pyplot as plt
import torch
import torch.nn as nn
import numpy as np
from torch.utils.data import Dataset, DataLoader
from sklearn.model_selection import train_test_split
import torch.nn.functional as F
from sklearn.metrics import confusion_matrix, classification_report
from collections import defaultdict
from textwrap import wrap
```

```
%reload_ext watermark
%watermark -v -p numpy,pandas,torch,transformers
warnings.filterwarnings("ignore")
```

```
Python implementation: CPython
Python version        : 3.7.15
IPython version       : 7.9.0
```

```
numpy      : 1.21.6
pandas     : 1.3.5
torch      : 1.12.1+cu113
transformers: 4.23.1
```

Foi utilizado a versão base, do Berimbau:

```
[ ] PRE_TRAINED_MODEL_NAME = 'neuralmind/bert-base-portuguese-cased'

tokenizer = BertTokenizer.from_pretrained(PRE_TRAINED_MODEL_NAME)
```

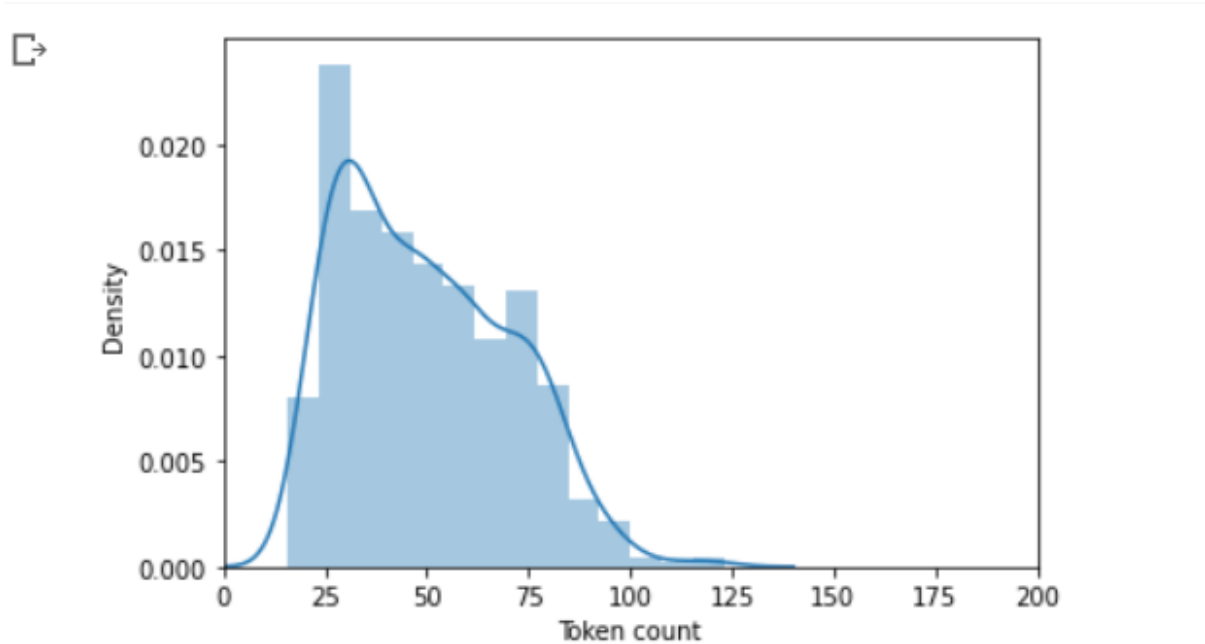
O banco de dados utilizado foi:

```
df = pd.read_csv("Dados_processadoPrincipal - Dados_processadoV3.csv")
df = df.dropna()
df = df.drop('Unnamed: 0', axis=1)
df = df.reset_index()
```

A representação dos dados informa o número do tweet, nome do usuário, texto, Favor de Lula, Favor de Bolsonaro e Neutro:

index		Nome	Twitter	Favor do candidato Lula	Favor do candidato Bolsonaro	Neutro
0	0	helo tavares	wilsonpratts eu tenho fe. eu acredito.lula'ind...	1	0	0
1	1	marco siqueira	sen_cristovam bora deixar essa discussao pra d...	1	0	0
2	2	leonardo lula silva	cordeironereida bom dia. estou te seguindo, me...	1	0	0
3	3	ana	as mulheres;'na liberdade religiosa;'no lgbtqi...	1	0	0
4	4	claudio ou nene (meu apelido e nene mesmo)	mbendixens vote lula13 p manicure, empreendedo...	1	0	0
...
616	622	jaissontc	choquei se der um zoom na conta bancaria da ch...	0	0	1
617	623	cil77733322	cristinagraemi , inumeros canais estao replica...	0	1	0
618	624	silas fiorotti	"filha de bolsonaro [laurinha] entrara em cole...	0	0	1
619	625	fiorentino	priscillalleite aqui nao tem espaco para ladao...	0	1	0
620	626	murakami baldwin	lula e bolsonaro sao dois mentirosos pensam qu...	0	0	1
...

Foi necessário definir o tamanho máximo que o modelo Bertimbau iria trabalhar, para isso foi feita uma análise da quantidade de caracteres em média, e foi definido como tamanho máximo de 160 caracteres, já que o twitter não permite a ultrapassagem de 140 caracteres para publicação, podemos observar que a maior parte dos dados se restringe a um número próximo a 75 caracteres.



Foi dividido os dados em dois grupos, sendo o primeiro responsável por treinamento e o segundo para realizar testes:

```
[ ] df_train, df_test = train_test_split(df, test_size=0.2, random_state=RANDOM_SEED)
    df_val, df_test = train_test_split(df_test, test_size=0.5, random_state=RANDOM_SEED)
```

O tamanho do Batch como também a tokenização:

```
BATCH_SIZE = 16

train_data_loader = create_data_loader(df_train, tokenizer, MAX_LEN, BATCH_SIZE)
val_data_loader = create_data_loader(df_val, tokenizer, MAX_LEN, BATCH_SIZE)
test_data_loader = create_data_loader(df_test, tokenizer, MAX_LEN, BATCH_SIZE)
```

A quantidade de épocas, otimizador e a taxa de aprendizagem:

```
[29] EPOCHS = 10

optimizer = AdamW(model.parameters(), lr=3e-5, correct_bias=False)
total_steps = len(train_data_loader) * EPOCHS

scheduler = get_linear_schedule_with_warmup(
    optimizer,
    num_warmup_steps=0,
    num_training_steps=total_steps
)

loss_fn = nn.CrossEntropyLoss().to(device)
```

3. Resultados

3.1 Resultado dos Experimentos

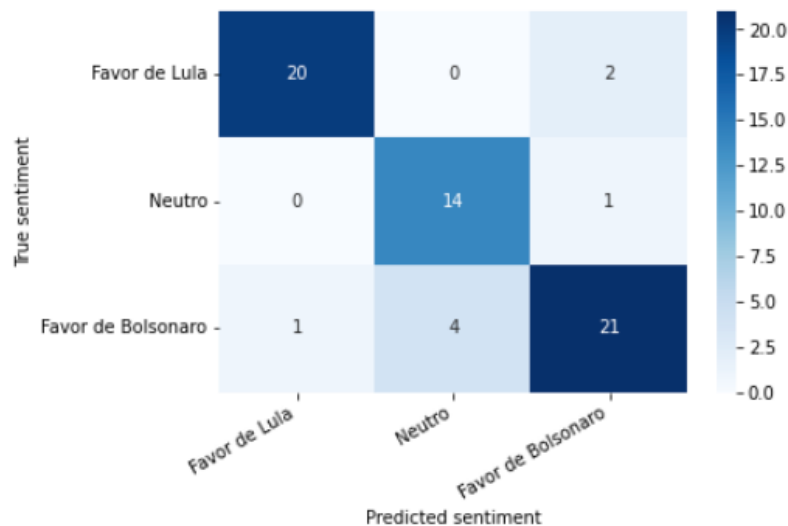
Os resultados das Epoch, logo abaixo:

Epoch 1/10 ----- Train loss 0.7591765138410753 accuracy 0.6330645161290323 Val loss 0.33632767572999 accuracy 0.8870967741935484	Epoch 6/10 ----- Train loss 0.05829935713911489 accuracy 0.9858870967741935 Val loss 0.5291077205911279 accuracy 0.9193548387096774
Epoch 2/10 ----- Train loss 0.42168154122848667 accuracy 0.8508064516129032 Val loss 0.23378240317106247 accuracy 0.9354838709677419	Epoch 7/10 ----- Train loss 0.028267954844200322 accuracy 0.9919354838709677 Val loss 0.4752477420552168 accuracy 0.9354838709677419
Epoch 3/10 ----- Train loss 0.2387208632163463 accuracy 0.9294354838709677 Val loss 0.39284786442294717 accuracy 0.9193548387096774	Epoch 8/10 ----- Train loss 0.022652344920882774 accuracy 0.9919354838709677 Val loss 0.5028113530715927 accuracy 0.9354838709677419
Epoch 4/10 ----- Train loss 0.13995074450729356 accuracy 0.9637096774193548 Val loss 0.45259765337686986 accuracy 0.9032258064516129	Epoch 9/10 ----- Train loss 0.009641421271578198 accuracy 0.9979838709677419 Val loss 0.4967297180555761 accuracy 0.9354838709677419
Epoch 5/10 ----- Train loss 0.08193011288981765 accuracy 0.9778225806451613 Val loss 0.46342231816379353 accuracy 0.9354838709677419	Epoch 10/10 ----- Train loss 0.0022445900399508255 accuracy 1.0 Val loss 0.5023523359704996 accuracy 0.9354838709677419

Podemos observar na imagem em seguida a precisão, recall e f1-score do modelo:

	precision	recall	f1-score	support
Favor de Lula	0.95	0.91	0.93	22
Neutro	0.78	0.93	0.85	15
Favor de Bolsonaro	0.88	0.81	0.84	26
accuracy			0.87	63
macro avg	0.87	0.88	0.87	63
weighted avg	0.88	0.87	0.87	63

Os resultados dos testes encontrados na matriz de confusão:



Logo em seguida, foi criado um total de 7 textos, simulando uma mensagem do twitter e observamos o comportamento do modelo:

```
Review text: Lula para presidente
Favor de Lula      0.990285
Neutro             0.008259
Favor de Bolsonaro 0.001456
Name: 0, dtype: float64
=====
```

```
Review text: vote 22 pelo bem do brasil!
Favor de Lula      0.000369
Neutro             0.000303
Favor de Bolsonaro 0.999328
Name: 0, dtype: float64
=====
```

```
Review text: Na moral, todos os candidatos são pessimos.
Favor de Lula      0.000983
Neutro             0.998557
Favor de Bolsonaro 0.000460
Name: 0, dtype: float64
=====
```

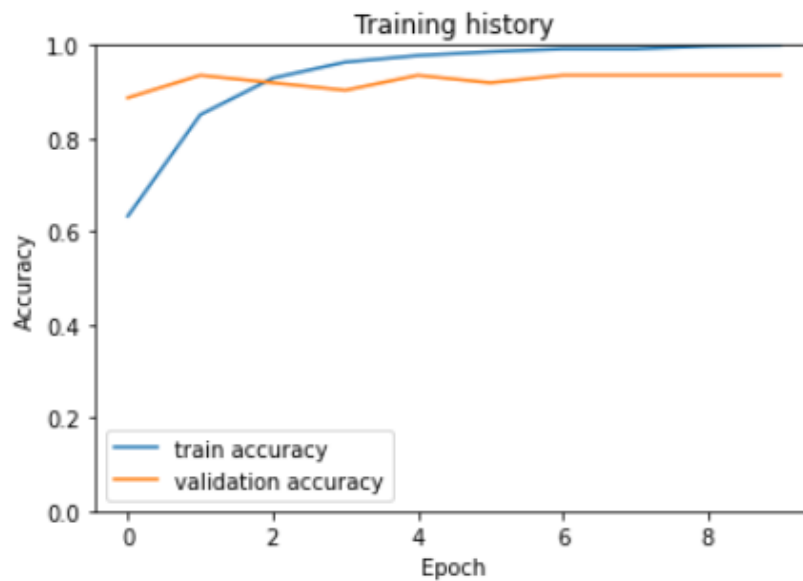
```
Review text: Lula é tao competente quanto bolsonaro
Favor de Lula      0.000503
Neutro             0.000888
Favor de Bolsonaro 0.998609
Name: 0, dtype: float64
=====
```

```
Review text: Precisamos do fim da esquerda
Favor de Lula      0.994512
Neutro             0.004675
Favor de Bolsonaro 0.000813
Name: 0, dtype: float64
=====
```

```
Review text: Bolsonaro é pessimo
Favor de Lula      0.734492
Neutro             0.263123
Favor de Bolsonaro 0.002386
Name: 0, dtype: float64
=====
```

```
Review text: A esquerda destrui o Brasil, a salvação é #bolsonaro22
Favor de Lula      0.000156
Neutro             0.000309
Favor de Bolsonaro 0.999535
Name: 0, dtype: float64
=====
```

Gráfico da acurácia de treinamento e validação:

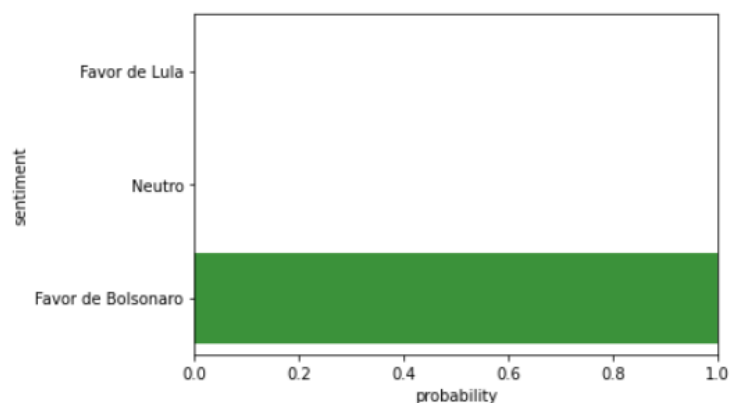


Observando a forma que foi avaliada do banco de dados:

```
neymar ganhou pontos comigo, se mostrou um patriota, um brasileiro,
que se preocupa com os valores cristaos e com a familia, seguimos
juntos pra reeleger nosso presidente bolsonaro22
```

True sentiment: Favor de Bolsonaro

```
[84] sns.barplot(x='values', y='class_names', data=pred_df, orient='h')
plt.ylabel('sentiment')
plt.xlabel('probability')
plt.xlim([0, 1]);
```



3.1 Análise dos Resultados

Com base nos estudos realizados na base de dados, pude observar que partes dos dados não são concisos, eles refletem a opinião através de um fato ou ocorrência a um candidato de forma bem específica, como foi o caso da entrevista na sbt, parte dos dados extraídos são respostas a publicações, dados escritos de forma errada, abreviada, analogias matemáticas (ex: bolsonaro = lixo, lula = potência econômica), por conta disso acredito que a base de dados não é bem representativa e pequena, acredito que com o grande número de tweets, o modelo não sofreria de overfitting, contudo o modelo representou de forma satisfatória quando realizado testes de classificação neutro, o que não foi bem o caso de Lula ou Bolsonaro.

Segue abaixo algumas imagens de dados rotulados:

13runo miranda	presidente otario = sem aumento de salario presidente legal = salario com aumento real lula lulapresidente eleicao2022
----------------	------------------------------------------------------------------------------------------------------------------------------

sobre "pintou um clima", e preciso destacar q o assunto e muito serio e nojento, pois um idoso se convidou p entrar na casa de meninas menores de idade e depois saiu falando q eram garotas de programa!! cade as notas de repudio?? venezuelanas eleicao2022 lulapresidente13

heytauat 1 turno/todos os votos

57.259.505

51.072.345

4.915.423

3.399.287

600.955

559.708

81.129

53.519

45.620

25.625

16.604

(provavelmente) os votos da simo e ciro vao para lula, quase tds da sora tabm como bozo ira ganhar?

fora bolsonaro forabolsonaro lula13

lulapresidente13

o cagao, digo, ladrao, arregou e deixou o caminho livre! eleicao2022 debatesbt

quem e o elenco de #pantandar perto da atuacao do jair ? #eleicao2022

4. Conclusão

Nesse projeto foi observado que em tópicos importantes como as eleições presidenciais possuem artigos com métodos que envelheceram mal e nós possibilitam o uso de tecnologias mais recentes e produzir modelos com maior acurácia e precisão. O modelo Bertimbau tem a capacidade de identificar bem a análise de sentimento, contudo o modelo apresentado no projeto não conseguiu generalizar bem, isso ocorreu por conta da base de dados ser pequena e com uma representativa questionável, a idéia de criar uma documentação de rotulação de dados e definir regras é a única forma de garantir uma extração efetiva de dados com uma representatividade real, já que parte dos dados extraídos são confusos para serem entendidos, até mesmo por nós humanos.

5. Referências

- [1] Cristiani, André; Lieira, Douglas; Camargo, Heloisa. A Sentiment Analysis of Brazilian Elections Tweets. In: SYMPOSIUM ON KNOWLEDGE DISCOVERY, MINING AND LEARNING (KDMILE), 8, 2020, Evento Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 153-160. ISSN 2763-8944.
- [2] Akilandeswari J, Jothi G, Sentiment Classification of Tweets with Non-Language Features, Procedia Computer Science, Volume 143, 2018, Pages 426-433, ISSN 1877-0509
- [3] Souza, F., Nogueira, R., Lotufo, R. (2020). BERTimbau: Pretrained BERT Models for Brazilian Portuguese. In: Cerri, R., Prati, R.C. (eds) Intelligent Systems. BRACIS 2020. Lecture Notes in Computer Science(), vol 12319. Springer, Cham.
- [4] Ema Kušen, Mark Strembeck. Politics, sentiments, and misinformation: An analysis of the Twitter discussion on the 2016 Austrian Presidential Elections. Online Social Networks and Media, Volume 5, 2018, Pages 37-50. ISSN 2468-6964.
- [5] Hao Wang, Dogan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. 2012. A system for real-time Twitter sentiment analysis of 2012 U.S. presidential election cycle. In Proceedings of the ACL 2012 System Demonstrations (ACL '12). Association for Computational Linguistics, USA, 115–120.
- [6] Stieglitz, Stefan & Dang-Xuan, Linh. (2013). Emotions and Information Diffusion in Social Media — Sentiment of Microblogs and Sharing Behavior. Journal of Management Information Systems. 29. 217-248. 10.2753/MIS0742-1222290408.
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- [8] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP:

Analyzing and Interpreting Neural Networks for NLP, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

[9] André L. Cristiani , Douglas D. Lieira, Heloisa A. Camargo. 2018. A Sentiment Analysis of Brazilian Elections Tweets.

[10] Lara Grimminger, Roman Klinger. 2018. 2020. Hate Towards the Political Opponent: A Twitter Corpus Study of the 2020 US Elections on the Basis of Offensive Speech and Stance Detection.

[11] Mickel Hoang. 2021. Aspect-Based Sentiment Analysis Using BERT.