

UNIVERSIDADE FEDERAL DE SÃO CARLOS
Departamento de Computação

Leonardo Capellaro

Análise de polaridade e de tópicos em tweets no
domínio da política no Brasil

SÃO CARLOS -SP
2021

Leonardo Capellaro

Análise de polaridade e de tópicos em tweets no domínio da política no Brasil

Trabalho de graduação apresentado ao departamento de computação da Universidade Federal de São Carlos, como parte dos requisitos para obtenção do título de bacharel em Engenharia de Computação.

Orientador: Profa. Dra. Helena de Medeiros Caseli

DEDICATÓRIA

Este trabalho é dedicado à minha família, amigos, namorada e a todos que me apoiaram e me deram suporte durante minha graduação.

RESUMO

O campo da política no Brasil é um dos mais movimentados e mais polêmicos da última década. Com o advento das redes sociais, um novo canal de comunicação entre os eleitores e os políticos foi criado, com os usuários tendo um espaço para publicar suas opiniões, crenças e também acompanhar publicações de outras pessoas. Neste contexto, este trabalho mostra como modelos BERT se saem na análise de polaridade (positiva, negativa ou neutra) e de tópicos de grandes quantidades de *tweets* relacionados à política do Brasil. Experimentos foram realizados com *tweets* relacionados às eleições de 2018, e melhores resultados (F1 = 96%) foram obtidos para a classificação de polaridade em comparação com trabalhos anteriores. A avaliação qualitativa dos tópicos (a primeira usando BERTopic neste *corpus*) também mostrou resultados promissores.

Palavras-chave: PLN. BERT. Análise de sentimentos. Análise de tópicos. Português do Brasil.

ABSTRACT

The political scenario in Brazil is one of the busiest and most controversial in the last decade. With the advent of social networks, a new communication channel between voters and politicians was created, with users having a space to publish their opinions, beliefs and also follow the posts of other people. In this context, this work shows the performance of BERT models in the polarity (positive, negative or neutral) and topic analysis of *tweets* related to Brazilian politics. Experiments were carried out with *tweets* related to the 2018 elections achieving results (F1 = 96%) better than the ones obtained by previous works for the polarity classification. The qualitative analysis of the topics (the first one using BERTopic in this corpus) also resulted in promising insights.

Keyword: PNL. BERT. Sentiment analysis. Topic Analysis.

LISTA DE ILUSTRAÇÕES

Figura 1 - Processo de pré-treinamento e <i>fine-tuning</i> do Bert, de acordo com Devlin et al. (2018).	22
Figura 2 - Matriz de Confusão.	26
Figura 3 - Fluxo de funcionamento do BERTopic.	32
Figura 4 - Comparação da acurácia F1 entre os diferentes tipos de pré-processamento.	39
Figura 5 - Comparação da F1 entre os diferentes tipos de classificação para os dois pré-processamentos escolhidos.	40
Figura 6 - Quantidade de <i>tweets</i> classificados em cada rótulo por candidato.	41
Figura 7 - Porcentagem de <i>tweets</i> classificados em cada label utilizando os métodos BERT e SVM.	42

LISTA DE TABELAS

Tabela 1 - Exemplo de textos presentes no <i>corpus</i> utilizado para o aprendizado.	29
Tabela 2 - Exemplo dos pré-processamentos aplicados em um <i>tweet</i> .	36
Tabela 2 - Análise dos principais tópicos citados positivamente e negativamente para cada candidato.	43
Tabela 4 - Classificações incorretas de <i>tweets</i> pelo modelo.	44

LISTA DE SIGLAS

PLN	Processamento de linguagem natural
RNA	Redes neurais artificiais
MLM	Masked Language Model
NSP	Next Sentence Prediction
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Analysis
NMF	Non-Negative Matrix Factorization
VP	Verdadeiro positivo
VN	Verdadeiro negativo
FP	Falso positivo
FN	Falso negativo
AM	Aprendizado de máquina
BERT	Bidirectional Encoder Representations from Transformers
TF-IDF	Term frequency – inverse document frequency
GPU	Graphics Processing Unit
TPU	Tensor processing unit
ASIC	Accelerator application-specific integrated circuit
BTM	Biterm topic model

SUMÁRIO

1 INTRODUÇÃO	9
1.1 CONTEXTUALIZAÇÃO	9
1.2 OBJETIVO	11
1.3 MOTIVAÇÃO	11
1.4 ORGANIZAÇÃO DESTE DOCUMENTO	12
2 REVISÃO BIBLIOGRÁFICA	15
2.1 PROCESSAMENTO DE LINGUAGEM NATURAL	15
2.2 APRENDIZADO DE MÁQUINA	16
2.3 REDES NEURAIS ARTIFICIAIS	18
2.4 BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS - BERT	20
2.5 ANÁLISE DE SENTIMENTOS	23
2.6 ANÁLISE DE TÓPICOS	25
2.7 MÉTODOS DE AVALIAÇÃO	26
3 MATERIAIS E MÉTODOS	29
3.1 CORPUS	29
3.2 FERRAMENTAS	30
3.2.1 Análise de polaridade com BERT	30
3.2.2 Análise de tópicos com o BERTopic	32
4 DESENVOLVIMENTO	35
4.1 PREPARAÇÃO DOS DADOS	35
4.2 COMPARAÇÃO DOS MODELOS DE CLASSIFICAÇÃO	37
4.3 ANÁLISE DE TÓPICOS	38
5 DISCUSSÃO DOS RESULTADOS	39
5.1 MÉTODOS DE PRÉ-PROCESSAMENTO	39
5.2 MÉTODOS DE CLASSIFICAÇÃO	40
5.3 ANÁLISE DE TÓPICOS	41
6 CONCLUSÃO	45
REFERÊNCIAS	47

1 INTRODUÇÃO

Neste capítulo o tema do trabalho é apresentado e o problema do trabalho é contextualizado, mostrando a relevância do estudo do tema e a aplicabilidade e eficiência das ferramentas de processamento de linguagem natural existentes na exploração e resolução do problema proposto.

1.1 CONTEXTUALIZAÇÃO

A inteligência artificial é um dos grandes campos da área de tecnologia da informação que vêm sendo explorados nos últimos anos. Diversos estudos, pesquisas e avanços na área vêm se concretizando na última década, como o surgimento de carros autoguiados (CIGANOVIC et al., 2018), robôs para atendimento automatizado de clientes de grandes companhias (PAIVA, 2018) e até algoritmos que detectam e ajudam na prevenção de doenças (SADOUGH, 2018). Esses trabalhos mostram o quanto este tema é benéfico e vêm causando um impacto positivo para a população mundial.

Dentro do campo da inteligência artificial, há uma subárea chamada aprendizado de máquina (AM). O aprendizado de máquina é um tema discutido há várias décadas, tendo como uma de suas primeiras definições ser o “campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados” (SAMUEL, 1959). Dessa forma, através da exploração da construção de algoritmos que podem aprender a partir de exemplos, o aprendizado de máquina permite realizarmos previsões sobre dados e auxiliar na tomada de decisão. No AM, um algoritmo é usado para criar um modelo computacional adaptável ao problema escolhido pelo usuário. Ao longo dos anos, várias técnicas foram desenvolvidas para gerar esses algoritmos, como as árvores de decisão, o agrupamento por vizinhos mais próximos, as redes bayesianas, as redes neurais, entre outras, que têm apresentado resultados bastante satisfatórios em problemas de processamento de linguagem natural (PLN). Recentemente uma nova frente de pesquisa surgiu no PLN com o uso de redes neurais, como no trabalho proposto por Devlin et al. (2018), onde foi introduzido o BERT (*Bidirectional Encoder Representations from Transformers*).

As redes neurais artificiais (RNA) são técnicas computacionais que apresentam um modelo matemático inspirado nas estruturas neurais de organismos

inteligentes, modificando sua estrutura interna de acordo com uma função objetivo proposta (GROSSI et al., 2008). As primeiras publicações sobre redes neurais artificiais surgiram nas décadas de 1940 e 1950, por McCulloch e Pitts (1943) e Rosenblatt (1957). Uma RNA pode possuir uma série de camadas que, simulando neurônios, processam fragmentos da informação e geram como resultado a compilação do processamento de cada um desses fragmentos.

Neste trabalho, foi utilizada uma arquitetura neural chamada BERT (*Bidirectional Encoder Representations from Transformers*) para a qual existem diversos modelos pré-treinados, para vários idiomas, inclusive o português do Brasil. Por ser uma rede pré-treinada, sua última camada é utilizada para o aprendizado do problema específico, onde é realizado o chamado *fine-tuning*. Por exemplo, neste trabalho, o *fine-tuning* foi realizado para a análise de polaridade.

A análise de polaridade consiste em determinar qual a polaridade (ou valência) da opinião do autor de um texto com relação à entidade ou objeto em discussão. Desse modo é possível, por exemplo, verificar se a opinião do autor é positiva, negativa ou neutra em relação à entidade ou objeto alvo do texto. As opiniões possuem uma influência muito grande sobre o comportamento das pessoas. Algumas decisões sobre qual comida pedir ou qual aparelho celular comprar muitas vezes são influenciadas por opiniões de pessoas próximas ou de análises feitas por outras pessoas em redes sociais ou sites de compra. As redes sociais, como o Twitter, trouxeram a possibilidade de visualizarmos diversas opiniões de inúmeras pessoas diferentes, permitindo termos acesso a opiniões de pessoas com culturas, ensinamentos e vivências diferentes do nosso círculo social (BECKER, 2013).

O último assunto abordado neste trabalho é a análise de tópicos. A análise de tópicos é uma abordagem utilizada para lidar com grandes quantidades de textos, análises e *feedbacks*, de forma a retornar os principais pontos abordados nos textos de acordo com sua importância e recorrência. Uma das estratégias mais utilizadas para encontrar os tópicos (palavras, termos) mais relevantes de um texto é a TF-IDF (*term frequency – inverse document frequency*, em português, frequência do termo – inverso da frequência nos documentos). O TF-IDF calcula a frequência de um termo no documento e a compara com a frequência do termo nos documentos. Desse modo, termos que aparecem com frequência em alguns

documentos (doc1) mas não em todos, são bons representantes dos tópicos mais relevantes desse conjunto (doc1). Por meio de TF-IDF, conseguimos identificar e quantificar a importância da palavra em um documento ou *corpus*.

1.2 OBJETIVO

Motivados pelo avanço dos algoritmos de processamento de linguagem natural e de redes neurais dos últimos anos, este trabalho visa realizar uma análise de polaridade a partir de textos em português obtidos do Twitter, com o intuito de classificá-los em três classes: positivo, negativo ou neutro. Além da análise de polaridade, este trabalho também investigou a análise dos tópicos mais relevantes citados em cada classe de polaridade.

1.3 MOTIVAÇÃO

Um estudo realizado pelo IBGE¹, em 2019, demonstrou que 82,7% dos domicílios brasileiros possuíam conexão com a internet. Outro estudo realizado no mesmo ano pela Comscore² apontou que aproximadamente 88% da população brasileira tinha acesso a algum tipo de rede social, com o Twitter possuindo 40,2 milhões de usuários ativos no país. Por ser uma rede social que limita o número de caracteres nos textos postados na rede, o Twitter se tornou um canal de comunicação rápida e simplificada, sendo uma boa ferramenta para saber o que as pessoas estão comentando espontaneamente sobre os assuntos do momento (SALES e BARBOSA, 2019).

No domínio da política, as redes sociais também passaram a ser uma importante vitrine para os candidatos divulgarem seus trabalhos, suas propostas e opiniões e atingirem mais rapidamente seus eleitores. De acordo com um estudo divulgado pelo próprio Twitter³ em de 2018, no período de 16 de agosto a 28 de outubro de 2018 foram contabilizados 165 milhões de *tweets* relacionados às eleições, um volume quatro vezes maior que o total das eleições anteriores, em 2014.

¹<https://educa.ibge.gov.br/jovens/materias-especiais/20787-uso-de-internet-televisao-e-celular-no-brasil.html>

²<https://olhardigital.com.br/2019/07/05/noticias/brasil-e-o-pais-que-mais-usa-redes-sociais-na-america-latina/>

³https://blog.twitter.com/pt_br/topics/company/2018/como-foram-as-eleicoes-2018-no-twitter

Saber “o que sentem” e “sobre o que falam” os usuários do Twitter, no domínio da política brasileira, é a grande motivação deste trabalho. Para tanto, este trabalho apresenta resultados da investigação de análise de polaridade e de tópicos em *tweets*, escritos em português, no domínio da política brasileira coletados em 2018 (CRISTIANI et al., 2020). A análise de polaridade consiste em determinar qual a polaridade (ou valência) da opinião do autor de um texto com relação à entidade ou assunto em discussão. Desse modo é possível, por exemplo, verificar se a opinião do autor é positiva, negativa ou neutra em relação à entidade ou assunto alvo do texto. A análise de tópicos, por sua vez, visa lidar com grandes quantidades de textos, análises e *feedbacks*, de forma a retornar os principais assuntos/tópicos dos textos de acordo com sua importância e recorrência

Alguns trabalhos anteriores foram realizados analisando sentimentos em *tweets* do campo da política, como os de Cristiani et al. (2020) e Moreira et al. (2020). Moreira et al. (2020) realizaram uma análise da polarização da elite em comparação com a massa no procedimento de impeachment da presidente do Brasil em 2016. Cristiani et al. (2020) fizeram a análise de sentimentos dos *tweets* das eleições presidenciais de 2018 no Brasil, onde foi estudada a relação entre as opiniões dos usuários que publicaram sobre os candidatos durante o período das eleições e seu resultado final. Entretanto, no trabalho de Cristiani et al. (2020) foram utilizados métodos mais clássicos para a classificação dos *tweets*, como o SVM e Naive Bayes.

Assim, este trabalho visa analisar o comportamento do BERT no mesmo *corpus* das eleições de 2018 extraído e anotado manualmente por Cristiani et al. (2020) utilizando o BERTimbau (SOUZA et al., 2020). O BERTimbau é um modelo de linguagem contextualizado (considerado o estado-da-arte para diversas aplicações de PLN) treinado para o português do Brasil. Este trabalho compara os valores de acurácia obtidos utilizando diferentes técnicas de pré-processamento dos textos para a classificação, e analisa qual foi o avanço em performance e acurácia com o uso destas novas técnicas. Também é proposta uma análise de tópicos utilizando o BERTopic, algo ainda inédito neste *corpus*.

1.4 ORGANIZAÇÃO DESTE DOCUMENTO

Tendo em mente o contexto apresentado, este documento descreve o

desenvolvimento de um modelo computacional de classificação de polaridade (positiva, negativa ou neutra) em *tweets* do domínio da política, em português do Brasil, seguida da identificação dos principais tópicos das polaridades opostas (positiva e negativa).

Para tanto, o Capítulo 2 apresenta a revisão bibliográfica que embasou este trabalho. Os Capítulos 3 e 4 trazem os materiais e métodos usados nos experimentos, bem como os resultados obtidos. O Capítulo 5 apresenta uma discussão sobre os resultados e o 6 finaliza este documento com algumas considerações finais e conclusões.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta os conceitos teóricos envolvidos no trabalho, assim como as ferramentas e soluções desenvolvidas em trabalhos relacionados.

2.1 PROCESSAMENTO DE LINGUAGEM NATURAL

O processamento de linguagem natural emprega técnicas que têm como objetivo o aprendizado, entendimento e produção de conteúdos produzidos em linguagem humana. As primeiras abordagens computacionais para o tema focaram na automação da análise da estrutura linguística dos idiomas e no desenvolvimento de tecnologias iniciais para aplicações como a tradução automática de textos e o reconhecimento e síntese da fala humana (HUANG e SMITH, 2006). Com o passar dos anos, estes algoritmos foram se tornando cada vez mais refinados e hoje essas ferramentas são utilizadas em diversos aplicativos do nosso cotidiano, como nas assistentes eletrônicas como a Siri, Cortana e Google Assistente; mecanismos de tradução em tempo real em vídeo chamadas; e na identificação de escrita para obtenção de resultados otimizados em sites de busca, por exemplo (POURJALALI, 2017).

Estas técnicas obtiveram uma evolução exponencial nas últimas décadas, especialmente devido a alguns fatores chaves, como o grande aumento de poder computacional, um número muito grande de dados linguísticos na internet e o desenvolvimento de algoritmos de aprendizado de máquina cada vez mais sofisticados.

As primeiras abordagens para o processamento de linguagem natural utilizavam modelos heurísticos para a obtenção do resultado, como a tradução baseada em regras, por exemplo. Através de uma série de regras pré estabelecidas, o algoritmo buscava qual a palavra com a maior probabilidade de ser a tradução correta para a palavra utilizada como entrada e a retornava como resposta. Entretanto, os métodos e regras heurísticas e estatísticas, em geral, não possuem muita robustez para as variações de linguagens, muito pelo fato de possuírem uma capacidade muito limitada de analisar o contexto do texto.

Desde a chamada revolução estatística, que ocorreu no fim dos anos 1980 e no início dos anos 1990, diversas pesquisas em PLN passaram a utilizar mais o

aprendizado de máquina. O paradigma de aprendizado de máquina utiliza inferência estatística para aprender automaticamente as regras de modelagem através da análise de grandes *corpora* de textos com exemplos de escrita do mundo real.⁴

2.2 APRENDIZADO DE MÁQUINA

A área da inteligência artificial (IA) começou a ser estudada antes do surgimento da tecnologia que tornou possível executá-la. Uma das primeiras menções a redes neurais foi feita por McCulloch e Pitts (1943), onde foram idealizadas estruturas de raciocínio artificiais em forma de modelo matemático que buscavam imitar o funcionamento de nosso sistema nervoso. A área teve diversos avanços, principalmente teóricos, do meio dos anos 70 até meados da década de 80, onde um período de baixa se sucedeu e ficou conhecido como o inverno da inteligência artificial (HUANG e SMITH, 2006). Esse período ocorreu principalmente pelo fato de haver muitos estudos acadêmicos e expectativas altas no tema, porém nem tudo era concreto ou tão rápido quanto o esperado, havendo uma baixa no setor. Outro fator importante que contribuiu para a baixa do tema durante o período foi a limitação do poder computacional existente na época, que impedia a obtenção de resultados satisfatórios em um tempo viável de execução.

Nos anos 1980, Feigenbaum (1988) propôs um novo conceito de “sistemas especialistas” como softwares capazes de realizar atividades complexas e específicas de um campo, fazendo o papel de humanos, o que reacendeu a área. Em 1997, uma IA desenvolvida pela IBM derrotou o campeão soviético Garry Kasparov em uma rodada de xadrez. Nos anos 2000, os estudos voltados a carros autônomos se intensificaram e, no final da década, com o advento dos smartphones, os estudos voltados ao processamento de linguagem natural se tornaram ainda mais essenciais. Na década de 2010, os estudos com processamento de imagem se intensificaram, assim como o surgimento de assistentes pessoais, como a Siri, Google Assistente, Cortana e Alexa (POURJALALI, 2017). Algoritmos voltados a diversas frentes evoluíram, carros autônomos já estão em produção e o futuro é cada vez mais promissor.

Parte do princípio do funcionamento da inteligência artificial é a do

⁴ Um *corpus* (cujo plural é *corpora*) é um conjunto de documentos de textos compilados com escrita humana ou de computador.

aprendizado, onde a máquina entende o contexto do problema proposto e fornece automaticamente, como saída, uma solução esperada pelo usuário. Dentre os tipos existentes de aprendizado de máquina, há quatro principais citados por Sah (2020):

- Supervisionado: A máquina utiliza um conjunto de dados rotulados com as entradas e as saídas desejadas, adaptando o modelo a esses dados. Feito isso, este modelo pode ser aplicado a outras entradas de dados, que não foram usadas no aprendizado original, gerando uma saída para casos que ainda não estavam rotulados.
- Não supervisionado: Utilizado em casos onde os resultados finais da tarefa desejada não são bem conhecidos, como o agrupamento de uma coleção de artigos de acordo com a similaridade da escrita ou número de páginas, por exemplo. Neste caso, não há rótulos nos dados de entrada, e são utilizadas técnicas de associação ou clusterização para resolver o problema.
- Semi-supervisionado: Utilizado em casos onde há uma base de dados rotulados junto a uma base de dados não rotulados, e procura-se estender o aprendizado de máquina para as duas bases. Para isso, são utilizados algoritmos de aproximação para incrementar a quantidade de exemplos a serem utilizados no aprendizado do modelo.
- Por reforço: Neste caso, o modelo aprende executando ações e avaliando as recompensas. Dado um ambiente, o agente realiza uma ação a partir do estado inicial. Dada esta ação, ele receberá uma recompensa, que pode ser negativa ou positiva. De acordo com a recompensa, ele tomará outra atitude, criando um ciclo que gera o aprendizado.

Os modelos de aprendizado de máquina também podem ser caracterizados de acordo com a saída que se deseja obter através dele. Dentre estas saídas, as principais são:

- Classificação: Os dados de entrada são divididos em duas ou mais classes, e os dados de aprendizado são rotulados de acordo com essas classes. O modelo se adapta ao conjunto de treinamento e se estende aos dados de entrada não classificados, gerando dados classificados. Essa abordagem em geral é realizada de modo supervisionado ou semi-supervisionado.
- Clusterização: Os dados de entrada são divididos em grupos de acordo com

uma técnica pré estabelecida de agrupamento, como por similaridade, por exemplo. Neste caso, os grupos não são conhecidos previamente, portanto este aprendizado é realizado de modo não supervisionado.

- Regressão: A regressão geralmente é utilizada em casos onde as saídas do algoritmo são contínuas, como na previsão de uma temperatura de ambiente, por exemplo. Neste caso, em geral, o algoritmo também é executado de modo supervisionado.

Neste trabalho investigou-se tanto o aprendizado supervisionado para a classificação da polaridade em 3 classes (negativo, neutro, positivo) como o não supervisionado, via clusterização para a identificação dos tópicos.

2.3 REDES NEURAIS ARTIFICIAIS

As Redes Neurais Artificiais (RNA) são técnicas computacionais que utilizam modelos matemáticos com o intuito de simular o funcionamento de uma estrutura neural orgânica presente em seres vivos inteligentes. Uma RNA é composta de uma cadeia de neurônios artificiais, onde a informação é processada em cada neurônio e uma complexa relação em cadeia é criada entre a entrada e a saída da rede.

Os neurônios biológicos se comunicam entre si através de sinapses, que é a região onde dois neurônios entram em contato e onde os impulsos nervosos são transmitidos entre eles. Os impulsos recebidos por um determinado neurônio são processados e, em dado momento, é atingido um limiar de ação, onde o neurônio dispara. Esse disparo é produzido através de uma substância neurotransmissora, e caso o neurônio esteja conectado a outro por sinapse, esse sinal é enviado para o neurônio adjacente da cadeia. De modo similar, uma RNA é composta por vários neurônios artificiais, que são as unidades básicas da rede. A comunicação entre os neurônios está diretamente relacionada a um valor de peso que é dado individualmente para cada conexão entre os neurônios. Cada um dos neurônios recebe os dados de uma entrada inicial ou de neurônio anterior, realiza os cálculos locais, e repassa o resultado do cálculo para o próximo neurônio, criando uma rede de interações que juntas levam a um resultado final.

O funcionamento de um neurônio individual pode ser dividido em 4 fases de acordo com McCulloch e Pitts (1943):

1. Entrada dos sinais, que podem ser providos de uma fonte inicial ou de outro neurônio.
2. Cada sinal recebido é multiplicado por um peso, que indica a influência dele na saída do neurônio.
3. Uma soma ponderada dos sinais com seus pesos é realizada, gerando um nível de atividade.
4. Se o nível de atividade obtido anteriormente for maior que um determinado limiar, a unidade produz uma determinada resposta de saída.

O que torna uma RNA adaptável a diferentes tipos de problemas com diferentes entradas é a possibilidade de variação dos pesos das conexões entre os neurônios de acordo com uma saída desejada. Assim, dado um conjunto rotulado de dados de treino, os pesos de cada uma das entradas são ajustados para se obter a saída desejada e essa RNA pode ser replicada para outras entradas desconhecidas.

Para o treinamento, na etapa inicial são distribuídos pesos aleatórios para cada sinapse. Geralmente um número de rodadas de treinamento ou um valor limite de erro são especificados como critério de parada, e a cada rodada esses pesos são recalculados de forma que o resultado convirja para o desejado.

Uma rede neural é organizada em camadas, tendo os neurônios de cada camada ao menos uma conexão com um neurônio da próxima camada. De forma macro, podemos separar uma RNA em três camadas principais:

- Camada de Entrada: Nesta camada é feita a entrada dos sinais na rede neural.
- Camada Intermediária: Nesta camada é realizada a maior parte do processamento dos dados na rede. Também conhecida como camada oculta, é nela onde são realizadas as conexões entre os neurônios e os cálculos ponderados.
- Camada de Saída: É a camada onde o resultado, após os cálculos e processamentos, é entregue.

Dessa forma, através das conexões entre os neurônios artificiais e do aprendizado da rede, podemos obter os resultados esperados simulando o funcionamento de um cérebro orgânico, e através disso realizar diferentes tarefas de agrupamento e classificação dos dados de acordo com as entradas.

De acordo com Surya (2019), o aumento de poder computacional na década de 2010 contribuiu muito para o uso de redes neurais profundas (*deep*). Diversos avanços foram realizados no campo de processamento paralelo, como a API CUDA⁵, que permite usar os núcleos de processamento de uma GPU (Graphics Processing Unit, em português, Unidade de Processamento Gráfico) para a paralelização de cálculos e a criação do TPU (*Tensor processing unit*), que é um ASIC (*accelerator application-specific integrated circuit*, em português, circuitos integrados de aplicação específica) focado em processamentos de inteligência artificial. Com estes avanços, o poder computacional deixou de ser um fator limitante na implementação de redes neurais profundas, as quais passaram a ser aplicadas em diversas tarefas, como em reconhecimento facial, de voz e em identificação de entidades em imagens, por exemplo (SURYA, 2019).

2.4 BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS - BERT

O BERT (*Bidirectional Encoder Representations from Transformers*, ou Representação de Codificador Bidirecional de Transformadores - em tradução livre) é uma arquitetura neural profunda proposta por Devlin et al. (2018) e possui como característica fundamental o fato de ser bidirecional, o que permite uma maior compreensão do contexto de uma palavra e do texto como um todo, analisando as palavras adjacentes em ambas as direções. Essa característica permite obtermos um desempenho muito bom mesmo utilizando amostras pequenas de texto, enquanto modelos unidirecionais necessitam de uma amostra maior para obter resultados satisfatórios por analisar somente as palavras à esquerda ou à direita.

A unidade básica do BERT são os *Transformers*, que são mecanismos que aprendem as relações contextuais entre palavras em um texto. Um *transformer* possui dois mecanismos principais: um codificador, que lê o texto de entrada; e um decodificador, que utiliza um modelo preditivo para realizar a compreensão do texto. Como o objetivo do BERT é gerar um modelo de linguagem, somente o mecanismo do codificador é utilizado.

O codificador do *transformer* é bidirecional, ou seja, ele não realiza a leitura do texto de entrada sequencialmente, e sim toda a entrada de uma vez em ambas

⁵ <https://developer.nvidia.com/cuda-zone>

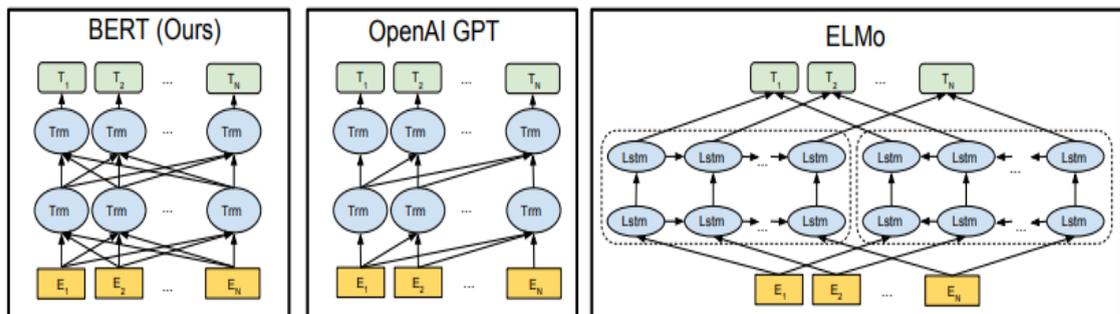
as direções. Dessa forma, é possível que o modelo compreenda o contexto de uma palavra analisando todas as informações que estão em sua adjacência. O texto de entrada é separado em uma sequência de *tokens*, que são a fragmentação de uma sentença. Esses *tokens* são transformados em vetores e são processados pela rede neural. Como saída, uma sequência de vetores é obtida, onde cada um dos vetores corresponde a um vetor da entrada com o mesmo índice.

Nos modelos de linguagem, há um grande desafio na definição do objetivo de predição. Em modelos unidirecionais, por exemplo, o modelo busca prever a próxima palavra de uma sentença. Entretanto, como o BERT é um modelo bidirecional, duas novas estratégias de predição foram implementadas: o modelo de linguagem mascarada (*Masked Language Model*, MLM) e a predição de próxima sentença (*Next Sentence Prediction*, NSP).

Na MLM, o algoritmo substitui 15% dos *tokens* de um texto por um *token* mascarado, que oculta a palavra original. Sendo assim, o algoritmo tenta prever quais eram as palavras originais corretas a partir do contexto delas. A função de perda do algoritmo de predição leva em consideração somente a predição das palavras que foram mascaradas, e ignora o valor de predição do restante das palavras. Isso gera como consequência negativa uma conversão mais lenta que a dos modelos direcionais, porém esta característica negativa é compensada com um tratamento muito melhor do contexto.

Já na abordagem NSP, temos como entrada do algoritmo pares de sentenças ou frases completas, e na modelagem ele aprende se a segunda frase da tupla é subsequente da primeira. Para o treinamento, metade das frases escolhidas são realmente subsequentes e a outra metade são frases aleatórias do texto sem conexão alguma com a primeira. O modelo consegue distinguir as duas sentenças no texto através do uso de dois *tokens* diferentes. No começo de cada primeira sentença é inserido um token (CLS) e no fim de cada uma das sentenças é inserido um token (SEP). Para a previsão, a sequência completa é processada por um *Transformer* que analisa o contexto da sequência, e calcula a probabilidade da segunda sentença ser subsequente da primeira.

Figura 1: Processo de pré-treinamento do BERT, OpenAI e ELMo de acordo com Devlin et al. (2018).



Fonte: (DEVLIN et al., 2018)

Na maioria das aplicações práticas são utilizados modelos pré-treinados do BERT, que podem ser treinados em uma única língua ou em diversas línguas diferentes. Um modelo pré-treinando passa por uma etapa de ajuste ou sintonia fina (*fine-tuning*). Há simplificada três técnicas de *fine-tuning* que podem ser utilizadas em redes neurais:

- Treino da arquitetura completa: Nesta técnica, todas as camadas da rede neural pré-treinada passam por um reajuste de parâmetros com base no novo dado de entrada e os pesos da rede são atualizados para a tarefa específica. É um processo que pode ser custoso computacionalmente caso a rede possua muitas camadas intermediárias.
- Treino de algumas camadas enquanto congela outras: Nesta técnica, as camadas iniciais da rede são mantidas intactas, enquanto as camadas mais superiores passam por um retreinamento que ajusta os pesos para a especificidade da tarefa.
- Congelamento de toda a arquitetura: Nesta técnica, todas as camadas da rede são congeladas e uma ou mais camadas adicionais são anexadas à rede e treinadas para a especificidade do problema. Somente os pesos das camadas anexadas são alterados nesse processo.

O BERT utiliza a técnica de congelamento de toda a arquitetura, onde uma camada adicional é anexada à rede é modelada para a especificidade do problema. Desta forma, o modelo pré-treinado é criado utilizando grandes *corpora* e máquinas com grandes capacidades de processamento, deixando para o usuário apenas a

necessidade de realizar o *fine-tuning* para um problema específico. Na Figura 1 é possível ver a principal diferença entre o BERT e as arquiteturas utilizadas em outros métodos. O BERT possui conexões bidirecionais entre os *transformes*, com cada um deles sendo conectado com os outros tanto do lado esquerdo quanto do lado direito. Em outros métodos, a conexão era unilateral, ou bidirecional separada por módulos, onde um grupo de neurônios era conectado aos outros da direita para a esquerda, outro grupo da esquerda para a direita, e havia uma camada final que fazia a conexão entre os módulos.

Para o presente trabalho, foi utilizado o modelo pré-treinando BERTimbau⁶, disponibilizado pela *neuralmind*, empresa localizada no parque científico e tecnológico da Unicamp. Para o *fine-tuning* do modelo, foi utilizado o *corpus* publicado no artigo de Cristiani et al. (2020).

2.5 ANÁLISE DE SENTIMENTOS

A análise de sentimentos é uma das tarefas mais comuns de classificação de textos. A partir de um texto dado como entrada, nesta tarefa busca-se analisar se o sentimento subjacente é positivo, negativo ou neutro.

Esta análise vem sendo empregada largamente no mercado, principalmente em sites de vendas de produtos, com o intuito de entender o comportamento dos comentários dos clientes e colher *feedbacks* de forma automatizada. No contexto deste trabalho, analisar sentimentos dos textos escritos em redes sociais nos traz diferentes perspectivas sobre os tópicos abordados e políticos envolvidos. Isso nos permite fazer uma análise para comparar entre determinados candidatos qual está mais bem ou mal falado.

O treinamento de um modelo de análise de sentimentos geralmente é supervisionado, onde são utilizados como dados de treinamento um conjunto contendo textos e suas respectivas emoções, que geralmente são anotadas manualmente por mais de um anotador para evitar algum tipo de viés na análise. Terminado o treinamento, podemos aplicar o modelo para textos não anotados, de forma que o modelo retorne qual o sentimento envolvido naquele texto.

Podemos encontrar diversos trabalhos que utilizam a análise de sentimentos

⁶ Disponível em: <https://huggingface.co/neuralmind/bert-base-portuguese-cased>. Acesso em: 16 set. 2021.

em *tweets* na língua portuguesa. Bouazizi e Ohtsuki (2016) mostram uma abordagem para a quantificação de sentimentos utilizando a técnica de classificação Random Forest, onde a F1-Score⁷ da classificação ficou em 81%. Nesta abordagem, foi utilizado um *corpus* com aproximadamente 40.000 *tweets* e, para cada um deles, foram atribuídos sentimentos de amor, felicidade, ódio, surpresa, alívio, entusiasmo, preocupação ou neutralidade.

Em Pereira (2019), o autor realizou uma análise dos sentimentos dos *tweets* para cada um dos candidatos das eleições de 2018, separados por evento, comparando o sentimento em todos os debates realizados no período e analisando a popularidade de cada candidato com base nos sentimentos positivos dos textos da rede social. Foram utilizados os métodos SVM e Naive Bayes, onde o método SVM atingiu 86% de acurácia no melhor caso, enquanto o Naive Bayes ficou em 85%.

Em Christie (2018) os autores utilizaram os algoritmos de Naive Bayes, SVM e Random Forest para realizar uma análise de posicionamento em *tweets* de diversos candidatos às eleições de 2018, classificando-os em contra, a favor ou neutro. O algoritmo SVM conseguiu uma F1 de 99% em um dos conjuntos utilizados na análise.

Já Cristiani et al. (2020) utilizaram as técnicas de classificação *Naive Bayes* e *SVM* em um *corpus* com aproximadamente 370.000 *tweets* sobre as eleições presidenciais de 2018. O trabalho buscava relacionar os sentimentos empregados nos *tweets* sobre cada candidato com o resultado final das eleições, e a F1 dos métodos Naive Bayes e SVM ficaram em 54,21% e 66,16% respectivamente. Os autores concluíram que o algoritmo SVM se mostrou uma boa escolha para a tarefa de classificação de textos utilizando aprendizado de máquina supervisionado, e que a porcentagem de *tweets* positivos da base analisada teve uma forte relação com as pesquisas realizadas para as eleições daquele ano.

O presente trabalho utiliza os mesmos *corpora* obtidos por Cristiani et al. (2020), entretanto buscando empregar o BERT como método de classificação, com o intuito compará-lo ao SVM e Naive Bayes utilizados pelos autores em seu trabalho.

⁷ A F1 é uma medida de avaliação utilizada para medir o desempenho de modelos classificatórios, detalhada na seção 2.7 deste trabalho.

2.6 ANÁLISE DE TÓPICOS

A análise de tópicos é uma tarefa de aprendizado de máquina que permite identificar os tópicos abordados em um texto. Este tipo de análise processa e classifica as informações de acordo com a importância no texto e assim permite gerar conhecimento através dos tópicos que mais aparecem nele.

Dou e Liu (2016) dizem que um dos grandes objetivos desta tarefa é obter tópicos de um texto que capturem as características do *corpus* e que sejam fáceis de se interpretar. Cada tópico consiste em um conjunto de termos que, quando combinados, geram um significado de alto nível. Para o presente trabalho, utilizaremos a modelagem de tópicos com o intuito de obter os principais tópicos em *tweets* pré-classificados como positivos e negativos no domínio da política.

Diversas abordagens da análise de tópicos foram empregadas em trabalhos relacionados. Pinto et al. (2020) utilizaram modelagem de tópicos e análise de sentimentos para analisar mensagens do Twitter durante a pandemia da COVID-19 em *tweets* da língua inglesa. Os pesquisadores utilizaram três algoritmos de aprendizagem de máquina: o *Latent Dirichlet Allocation* (LDA), o *Non-Negative Matrix Factorization* (NMF) e o *Latent Semantic Analysis* (LSA). Foi realizada uma análise qualitativa dos resultados, onde os autores buscaram identificar qual algoritmo gerou a melhor relação entre os termos e a maior coerência relacionada ao tema da pesquisa, e foi identificado que o algoritmo com o melhor resultado foi o NMF.

Em Moreira et al. (2020), foi proposta uma análise da polarização da elite em comparação com a massa no procedimento de impeachment da presidente do Brasil em 2016. Dentre os estudos realizados no trabalho, estavam o da análise temporal de tópicos a partir de *tweets* de parlamentares brasileiros, onde foi proposta uma análise da evolução dos tópicos discutidos pelos parlamentares ao longo do tempo. Para isto, foi utilizado o algoritmo BTM (*biterm topic model*) que busca aprender os tópicos em *corpora* de textos curtos. Devido às restrições para medir a coerência de tópicos para a análise de textos em português, foi realizada uma análise qualitativa para verificar a coerência dos tópicos, que mostrou termos relacionados ao impeachment da presidente e à crise política no Brasil. Na análise qualitativa, o algoritmo BTM demonstrou bons resultados na análise de tópicos,

retornando termos que faziam sentido com os acontecimentos políticos das datas analisadas.

Silveira et al. (2021) utilizaram o BERTopic⁸ (GROOTENDORST, 2020) para fazer uma análise de tópicos em documentos do meio jurídico. Para a validação foi realizada uma análise qualitativa, onde os especialistas no tema analisaram os tópicos gerados pelo algoritmo comparando-os com os principais tópicos dos documentos originais. Foi constatado que 84,6% dos tópicos gerados pelo modelo correspondiam aos temas principais dos documentos.

O BERTopic utiliza as *embeddings* de palavras presentes em um modelo BERT para gerar agrupamentos hierárquicos por densidade de palavras utilizando o HDBSCAN e, por fim, gerar tópicos com base na importância das palavras utilizando uma variação do TF-IDF (*term frequency–inverse document frequency*), chamada de c-TF-IDF, com o prefixo ‘c’ indicando a palavra *cluster*.

O presente trabalho aplica o mesmo método utilizado por Silveira et al. (2021) no *corpus* de (CRISTIANI et al., 2020), com o intuito de obter os principais tópicos positivos e negativos de cada candidato concorrente à eleição.

2.7 MÉTODOS DE AVALIAÇÃO

Treinar um modelo de classificação é uma parte importante do processo de aprendizado de máquina. Porém, sem uma medida de avaliação não conseguimos dizer o quão bom o modelo é e não podemos avaliar quais os melhores parâmetros a serem utilizados no treinamento.

Há algumas medidas de desempenho que nos ajudam a avaliar a performance de um modelo. Para entendê-las, primeiro precisamos introduzir alguns conceitos preliminares através de uma matriz de confusão. A classe real representa os valores reais para as instâncias do conjunto, os quais foram pré-determinados por anotadores humanos ou por algum outro tipo de representação da verdade. A classe esperada são os valores que são gerados pelos modelos de classificação.

Figura 2: Matriz de Confusão.

⁸ <https://maartengr.github.io/BERTopic/index.html>

		Classe esperada	
		Sim	Não
Real	Sim	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Fonte: Elaborado pelo autor.

O verdadeiro positivo (VP) e o verdadeiro negativo (VN) são os casos onde o valor previsto pelo modelo correspondeu ao valor esperado, sendo ele positivo ou negativo, respectivamente. Já os casos de falso positivo (FP) e falso negativo (FN) ocorrem quando o modelo prevê um valor diferente do que real.

A medida de avaliação utilizada no presente trabalho, para avaliar a classificação de polaridade, é o F1-Score. Para explicá-la, é preciso definir três outras medidas nas quais o F1 se baseia:

- Precisão: É a divisão do número de casos que foram corretamente identificados como positivos sobre o total de casos positivos preditos. Essa medida é bastante útil quando os falsos positivos devem ser evitados.
- Revocação: É a divisão do número de casos que foram corretamente identificados como positivos sobre o total de casos positivos reais, ou seja, verdadeiros positivos e falsos negativos. É bastante utilizada quando a penalidade de falsos negativos precisa ser alta.
- Acurácia: É a divisão dos verdadeiros positivos e verdadeiros negativos pelo total de predições. É utilizada quando todas as classes são igualmente importantes.
- F1-Score: É uma média harmônica da precisão e da revocação. A fórmula é dada por:

$$F1_Score = 2 * \frac{(Precisão * Revocação)}{(Precisão + Revocação)}$$

A F1-Score nos dá uma boa medida para casos classificados incorretamente, ou seja, para falsos positivos e falsos negativos. Ela é bastante utilizada em casos onde as classes não são balanceadas, que é o caso deste trabalho.

Já para a análise de tópicos, assim como nos trabalhos relacionados, foi utilizado um método qualitativo de avaliação, verificando através da análise individual de cada tópico se ele fazia ou não sentido com o contexto das bases estudadas. A análise qualitativa é usada neste cenário pois a análise de tópicos geralmente possui um conceito avaliativo subjetivo, onde é necessário que um especialista ou uma pessoa que conheça o domínio do documento verifique se os tópicos fazem sentido com o tema proposto.

3 MATERIAIS E MÉTODOS

Este capítulo descreve os materiais (*corpus* e ferramentas) e métodos (BERT) usados na análise de polaridade e de tópicos em *tweets* no domínio da política brasileira.

3.1 CORPUS

O *corpus* utilizado neste trabalho é o mesmo de Cristiani et al. (2020). Esse *corpus* está composto por 600 *tweets* anotados manualmente com suas respectivas polaridades: positivo, quando a mensagem publicada demonstra apoio ao candidato; negativo, quando a mensagem publicada demonstra rejeição ao candidato; e neutro, quando a mensagem publicada não demonstra opinião sobre o candidato. A Tabela 1 traz exemplos de *tweets* originais do *corpus* de Cristiani et al. (2020).

Tabela 1: Exemplo de textos presentes no *corpus* utilizado para o aprendizado.

Texto do <i>tweet</i>	Classe
a justiça deveria ser sempre ágil, como é para tudo relacionado ao lula. só acho.	neutro
fui de #ciro no primeiro turno. parte não é minha preferência. mas agora sou #haddadpresidente desde criancinha. #eleições2018	positivo
realmente bolsonaro não tem raciocínio lógico. não conseguiu sair da primeira pergunta. #bolsonaronejornalnacional	negativo

Fonte: Elaborado pelo autor.

Além do *corpus* com 600 *tweets* anotados com polaridade, neste trabalho também foi utilizado o *corpus* de 369 mil *tweets* de Cristiani et al. (2020) sem anotação de polaridade, que foram obtidos durante o segundo turno das eleições.

O *corpus* contendo os 600 *tweets* anotados foi utilizado para realizar o fine-tuning do modelo de classificação de polaridade utilizando o BERT. Já o *corpus* contendo os 369 mil *tweets* sem anotação foi classificado utilizando o modelo gerado na primeira etapa, dividido em polaridade por candidato (Haddad+, Haddad-, Bolsonaro+, Bolsonaro-) e, posteriormente, foram gerados tópicos utilizando o BERTopic para cada uma das bases divididas.

3.2 FERRAMENTAS

Toda a implementação deste trabalho foi realizada na plataforma Google Colaboratory⁹, que consiste de um *notebook* para execução de código Python sequencialmente e que possui máquinas robustas processando os algoritmos em nuvem. A GPU utilizada para o treinamento do modelo de classificação de polaridade foi uma NVIDIA Tesla T4 com 16GB de memória de vídeo e 2560 CUDA *cores* para processamento paralelo. Já a análise de tópicos utilizando o BERTopic foi realizada em uma máquina contendo uma GPU NVIDIA Geforce GTX 1080 com 2560 CUDA *cores* e 8GB de memória de vídeo, um processador Ryzen 7 3800x com 8 núcleos e 16 threads e 32GB de memória RAM.

As bibliotecas utilizadas na implementação foram:

- Pandas¹⁰: Utilizada para leitura e manipulação dos dados de entrada.
- Numpy¹¹: Utilizada para manipulação de séries e cálculos gerais.
- Sklearn¹²: Utilizada para calcular medidas de avaliação dos modelos e realizar a divisão dos dados em conjunto de treino e teste.
- Enelvo¹³ (BERTAGLIA, 2017): Utilizada para normalizar palavras em conteúdos de texto escritos por usuário em português.
- Transformers¹⁴: Utilizada para disponibilizar ferramentas para processamento e tokenização, bem como modelos pré treinados em português para classificação de textos.
- BERTopic¹⁵ (GROOTENDORST, 2020): Utilizada para realizar o agrupamento hierárquico e a geração de tópicos por c-TF-IDF.
- NLTK¹⁶: Utilizada para remover as *stopwords* dos *tweets*.

3.2.1 Análise de polaridade com BERT

Como citado na seção 2.4, o BERT possui modelos pré-treinados que nos permitem realizarmos o *fine-tuning*. Como citado na seção 2.4, no contexto do BERT o *fine-tuning* realiza a modelagem de uma última camada da rede neural que será

⁹ <https://colab.research.google.com/>

¹⁰ <https://pandas.pydata.org/>

¹¹ <https://numpy.org/>

¹² <https://scikit-learn.org/>

¹³ <https://github.com/thalesbertaglia/enelvo>

¹⁴ <https://huggingface.co/transformers/>

¹⁵ <https://github.com/MaartenGr/BERTopic>

¹⁶ <https://www.nltk.org/>

específica para o problema tratado pelo usuário, tendo as outras camadas já sido geradas através dos métodos MLM e NSP como citado.

Inicialmente, é definido qual será o comprimento máximo de cada texto utilizado como entrada. Quanto maior esse valor, mais memória precisará ser alocada para o treinamento do modelo. Como o *corpus* utilizado no presente trabalho possui textos de até 140 caracteres, esse foi o valor fixado.

Posteriormente, o modelo pré-treinado do BERT, o BERTimbau, é importado, assim como o tokenizador. Após a importação, os *encodings* do modelo são obtidos, gerando os *tokens*, que são fragmentações de uma uma sentença. Esses *tokens* são agrupados em uma matriz numérica que será processada pela rede neural.

Na etapa seguinte, os valores das classes passam por uma etapa de *one hot labeling*, etapa que transforma as labels em colunas e aplica valores binários indicando se o texto pertence àquela classe ou não. Feito isso, os dados são divididos entre dados de treino e dados de validação, e convertidos em *torch tensors*, uma matriz multidimensional contendo elementos com um único tipo de dados, capaz de ser processada por uma GPU. Após isso, é escolhido o tamanho do *batch*, que indica o número de amostras que serão propagadas pela rede neural por vez.

Em seguida, utilizamos o *BertForSequenceClassification*, que é o modelo do BERT que adiciona uma única camada linear no topo da rede neural para a classificação e escolhemos a função de otimização, que tem por intuito diminuir o erro entre os resultados obtidos pela rede em comparação com os resultados desejados durante a etapa de treinamento. Nesta etapa, também é escolhido o valor da taxa de aprendizagem da rede neural, que influencia diretamente até que ponto as informações recém-adquiridas pela rede irão substituir as informações antigas, definindo o tamanho do passo a cada iteração enquanto se move em direção a um valor mínimo de uma função de perda (MURPHY, 2012). Por essa influência, essa etapa define a velocidade com que um modelo de aprendizado de máquina se adapta à especificidade da tarefa.

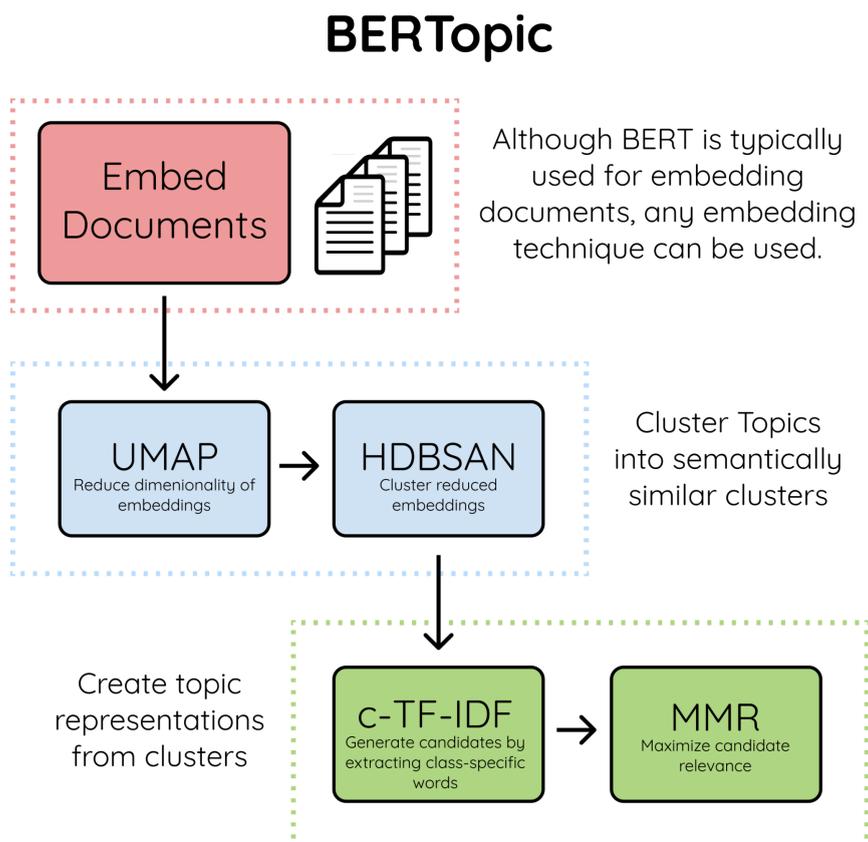
Por fim, a última camada gerada pelo *BertForSequenceClassification* é treinada com os parâmetros definidos anteriormente e com o treinamento sendo repetido n vezes, com n sendo o número de épocas de treinamento definida pelo usuário. O resultado final é um modelo treinado que pode ser aplicado a textos e

classificá-los com os rótulos das classes (*labels*) definidos anteriormente.

3.2.2 Análise de tópicos com o BERTopic

A primeira etapa do BERTopic converte os documentos em dados numéricos utilizando as técnicas de geração de *embeddings* do modelo BERT. Como etapa opcional, há a possibilidade de se realizar uma redução de dimensionalidade dos dados antes de realizar o agrupamento, que não foi efetuada neste trabalho. A segunda etapa é a de clusterização, que é realizada através do HDBSCAN, um algoritmo de agrupamento hierárquico por densidade proposto em Campello et al. (2013). Neste algoritmo, os documentos que possuem maior similaridade entre si são agrupados em *clusters* baseados na estabilidade do *cluster*. Uma das importantes características do HDBSCAN é o fato de ele não forçar a seleção de um dado para um determinado *cluster*. Caso o dado não se encaixe em nenhum grupo por similaridade, ele é considerado um *outlier*.

Figura 3: Fluxo de funcionamento do BERTopic.



Fonte: (GROOTENDORST, 2020)

A última etapa do processo é a seleção dos tópicos com base na importância

das palavras. Para isso, o autor desenvolveu uma técnica denominada c-TF-IDF. Esta técnica funciona de forma parecida com o TF-IDF original, que compara a importância das palavras analisando todo o *corpus*, entretanto, o c-TF-IDF utiliza os *clusters* gerados na etapa de agrupamento aplicando o TF-IDF em cada um deles. Esse processo classifica as palavras de acordo com a importância para cada grupo gerado no processo de agrupamento e extrai os principais tópicos de cada grupo. O cálculo é feito pela fórmula:

$$\text{c-TF-IDF} = \frac{t_i}{w_i} \cdot \log \frac{m}{\sum_j t_j}$$

onde a frequência de cada palavra t é extraída para classe i e dividida pelo número total de palavras da classe i . Essa etapa pode ser vista como uma forma de regularização das palavras frequentes na classe. Depois, esse valor é multiplicado pelo logaritmo do número total de documentos (m) dividido pela frequência total da palavra t ao longo de todas as classes n . Assim, é obtido um valor de importância para cada palavra em um *cluster* que será utilizada para criar os tópicos.

Para a análise, todos os textos dos *tweets* são convertidos em uma lista única encadeada e passada como parâmetro para o BERTopic. Após isso são realizadas todas as etapas descritas anteriormente, e um modelo é gerado como saída. Através deste modelo, podemos extrair os principais tópicos analisando a frequência e o valor de importância obtido anteriormente e gerar um mapa de distribuição de tópicos.

4 DESENVOLVIMENTO

Este capítulo descreve as etapas dos experimentos realizados para classificação de polaridade e análise de tópicos.

4.1 PREPARAÇÃO DOS DADOS

Para a preparação dos dados, foram realizados diversos experimentos a fim de se determinar o melhor método de pré-processamento para o estudo. No estudo realizado por Cristiani (2020) os textos do *corpus* foram colocados em caixa baixa, todos os *retweets* dos usuários foram removidos e qualquer tipo de hiperlink incluído na mensagem foi apagado. No presente trabalho, fizemos uma experimentação utilizando a ferramenta Enelvo¹⁷, desenvolvida por Bertaglia (2017). Esta ferramenta tem por objetivo tornar textos obtidos da internet padronizados, transformando palavras abreviadas em suas formas originais, removendo *hashtags* e marcações de usuário, normalizando o texto para caixa baixa e removendo emojis dos textos.

Neste trabalho, os experimentos de pré-processamento foram realizados variando quais características eram retiradas do texto e quais normalizações eram aplicadas. Os seguintes experimentos foram realizados com o auxílio da ferramenta Enelvo (BERTAGLIA, 2016):

- Original: Foram aplicados o pré-processamento utilizado por Cristiani (2020), que removem hiperlink, *retweets* e deixando o texto em caixa baixa.
- Enelvo puro: Os textos foram normalizados pela ferramenta Enelvo com os parâmetros padrão. Por padrão, a ferramenta troca algumas entidades do texto por *tags*, como o nome do usuário por “username”, *hashtags* por “*hashtag*” e números por “number”.
- Enelvo sem emoji: Neste caso, os textos foram normalizados com a mesma regra de substituição de entidades, porém removendo os emojis do texto.
- Enelvo sem emoji e *hashtag*: Segue o mesmo conceito anterior, porém removendo também as *hashtags* do texto.
- Enelvo Raw: O texto passa pelas etapas de normalização de abreviações, porém mantendo as entidades sem alteração, como usernames, *hashtags* e números.

¹⁷ <https://github.com/thalesbertaglia/enelvo>

- Enelvo Raw sem emoji: Segue o mesmo padrão do Enelvo Raw, exceto pela remoção dos *emojis*.
- Enelvo Raw sem emoji e *retweet*: Segue a normalização feita no Enelvo Raw, mas removendo *emojis* e *retweets*, mantendo *hashtags*.

Na tabela 2 podemos ver os diferentes tipos de pré-processamentos aplicados a um determinado *tweet* da base. É importante notar que o Enelvo falha em corrigir alguns termos pejorativos, como “bozonaro”, que foi trocado para “togado”. Outro fator importante foi a dificuldade em remover as *hashtags* sem o uso da ferramenta de substituição de entidades do Enelvo. As expressões regulares acabaram removendo pontuações ou causando falhas na remoção de algumas *hashtags*, portanto, as *hashtags* foram mantidas nas versões Raw.

Tabela 2: Exemplo dos pré-processamentos aplicados em um tweet.

<i>Tweet</i> original	RT @pqpasylum: 🌸 <i>tweet</i> para apreciar e enaltecer a beleza da mulher que matou o bozonaro essa noite 🌸 #bolsonaronojornalnacional https://t.co...
Pré-processamento realizado por Cristiani et al. (2020)	rt @pqpasylum: 🌸 <i>tweet</i> para apreciar e enaltecer a beleza da mulher que matou o bozonaro essa noite 🌸 #bolsonaronojornalnacional
Enelvo puro	retweeet username : 🌸 <i>tweet</i> para apreciar e enaltecer a beleza da mulher que matou o togado essa noite 🌸 <i>hashtag</i>
Enelvo sem emoji	retweeet username : <i>tweet</i> para apreciar e enaltecer a beleza da mulher que matou o togado essa noite <i>hashtag</i>
Enelvo sem emoji e <i>hashtag</i>	username : <i>tweet</i> para apreciar e enaltecer a beleza da mulher que matou o togado essa noite
Enelvo Raw	retweeet @pqpasylum : 🌸 <i>tweet</i> para apreciar e enaltecer a beleza da mulher que matou o togado essa noite 🌸 #bolsonaronojornalnacional
Enelvo Raw sem Emoji	retweeet @pqpasylum : <i>tweet</i> para apreciar e enaltecer a beleza da mulher que matou o togado essa noite #bolsonaronojornalnacional
Enelvo Raw sem emoji e <i>retweet</i>	@pqpasylum : <i>tweet</i> para apreciar e enaltecer a beleza da mulher que matou o togado essa noite #bolsonaronojornalnacional

Fonte: Elaborado pelo autor.

Com relação à base com 369 mil *tweets* não anotados, não foi possível processá-la com o Enelvo devido ao tamanho do *corpus*. Assim, um processamento

mais simples foi realizado com as seguintes etapas:

- Foram removidas *hashtags* e links através de expressões regulares.
- Todas as palavras foram colocadas em caixa baixa.
- Foram removidos os inícios de parágrafos representados por \n.
- Foram removidas vogais repetidas nas palavras (por exemplo, 'corruptooo' foi transformado em 'corrupto').
- Foram removidas as pontuações do texto (vírgula, interrogação, exclamação, pontos finais, dentre outras).
- Foram removidas as *stopwords* (como 'o', 'a', 'com', 'para', dentre outras) através da lista de *stopwords* presentes na biblioteca NLTK.
- Foram removidas as palavras com apenas uma letra.

4.2 COMPARAÇÃO DOS MODELOS DE CLASSIFICAÇÃO

Após a análise dos diferentes tipos de pré-processamento, o Enlvo Raw sem emoji e *retweet* foi escolhido para ser utilizado na etapa de treinamento do modelo BERT, por obter a melhor F1, como pode ser visto na Figura 4. Para a comparação, o BERT foi treinado utilizando os diferentes tipos de pré-processamento e foi calculada a F1 para cada uma das execuções. Com o intuito de comparar os resultados do pré-processamento e dos modelos utilizados por Cristiani et al. (2020) e o pré-processamento utilizado neste trabalho e aplicado no BERT, foi montado um experimento comparativo. Para tanto, foram utilizados os *corpora* pré-processados por Cristiani et al. (2020) e o Enlvo Raw sem emoji e *retweet*, além dos modelos de classificação BERT, Naive Bayes e SVM, sendo os dois últimos utilizados no trabalho de Cristiani et al. (2020).

O experimento foi dividido em duas etapas:

1. Foi preparado o *corpus* com o pré-processamento original e aplicado nos métodos de classificação BERT, Naive Bayes e SVM.
2. Foi preparado o *corpus* com o pré-processamento Enlvo Raw sem emoji e *retweet* e aplicado nos métodos de classificação BERT, Naive Bayes e SVM.

Foram anotados os resultados de F1 para cada um dos diferentes métodos de classificação e de pré-processamento. A ideia, neste caso, é replicar os

experimentos realizados por Cristiani et al. (2020) e testar os mesmos algoritmos utilizados por eles com o nosso pré-processamento utilizando o Enelvo. Além disso, os dois pré-processamentos foram utilizados para treinar um modelo BERT, o que nos permitiu comparar a diferença entre os modelos utilizados no trabalho original e neste trabalho.

4.3 ANÁLISE DE TÓPICOS

Para a análise de tópicos, o pré-processamento utilizado foi o citado na seção 4.1. O modelo BERT treinado com os 600 *tweets* anotados foi aplicado à base com 369 mil *tweets*, onde cada um deles recebeu uma classificação de sentimentos ou recebeu o valor nulo caso não houvesse classificação. Para a extração de tópicos, a base final classificada foi dividida em 4 bases distintas, sendo elas:

- *Tweets* positivos relacionados ao candidato Fernando Haddad (Haddad+).
- *Tweets* negativos relacionados ao candidato Fernando Haddad (Haddad-).
- *Tweets* positivos relacionados ao candidato Jair Bolsonaro (Bolsonaro+).
- *Tweets* negativos relacionados ao candidato Jair Bolsonaro (Bolsonaro-).

Posteriormente, utilizou-se o BERTopic com o modelo BERTimbau limitando o número de tópicos a 20, quantidade que foi definida empiricamente analisando os resultados gerados pelo modelo completo, que mostrou possuir aproximadamente este número de tópicos. Por ser um método de classificação não supervisionado, não é necessário realizar nenhum treinamento, utilizando apenas dados não rotulados.

5 DISCUSSÃO DOS RESULTADOS

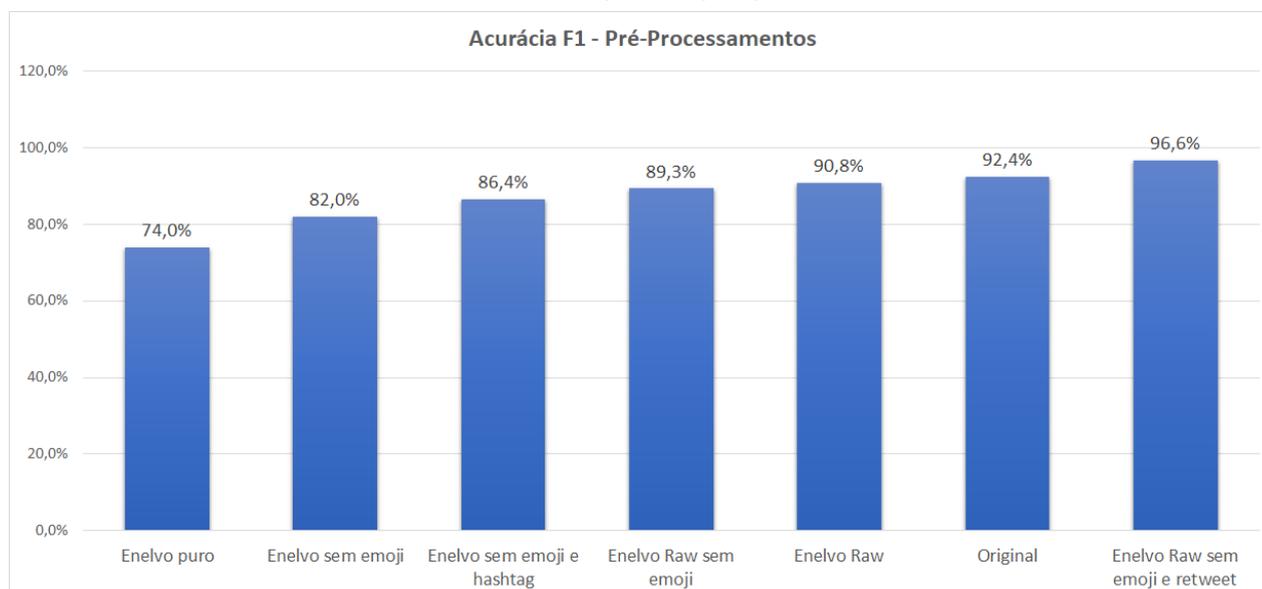
Este capítulo apresenta os resultados quantitativos e qualitativos dos experimentos realizados para classificação de polaridade e análise de tópicos.

5.1 MÉTODOS DE PRÉ-PROCESSAMENTO

O *corpus* foi dividido em duas partes: a base de treino e a base de teste. A base para o treinamento do modelo foi estabelecida para compor 90% do *corpus*, enquanto a base de teste possui os 10% restantes. Foram gerados sete arquivos distintos, cada um seguindo uma técnica distinta de pré-processamento conforme especificado na seção 4.1.

Com as sete bases pré-processadas prontas, foi aplicado o algoritmo BERT utilizando 15 épocas para cada uma das sete bases pré-processadas. Cada época gerava um resultado de F1, e para a comparação dos resultados foi anotada a maior F1 dentre as épocas.

Figura 4: Comparação dos valores de F1 obtidos para a classificação de polaridade, com o BERT, usando diferentes tipos de pré-processamento.



Fonte: Elaborado pelo autor.

Como pode ser visto na Figura 4, os resultados foram favoráveis para o Enelvo Raw sem emoji e *retweet*, que mantém as entidades originais do texto inalteradas, mas desconsidera emojis e *hashtags*. A melhor F1 utilizando o BERT para este caso foi de 96,6%. O pré-processamento original utilizado por Cristiani et

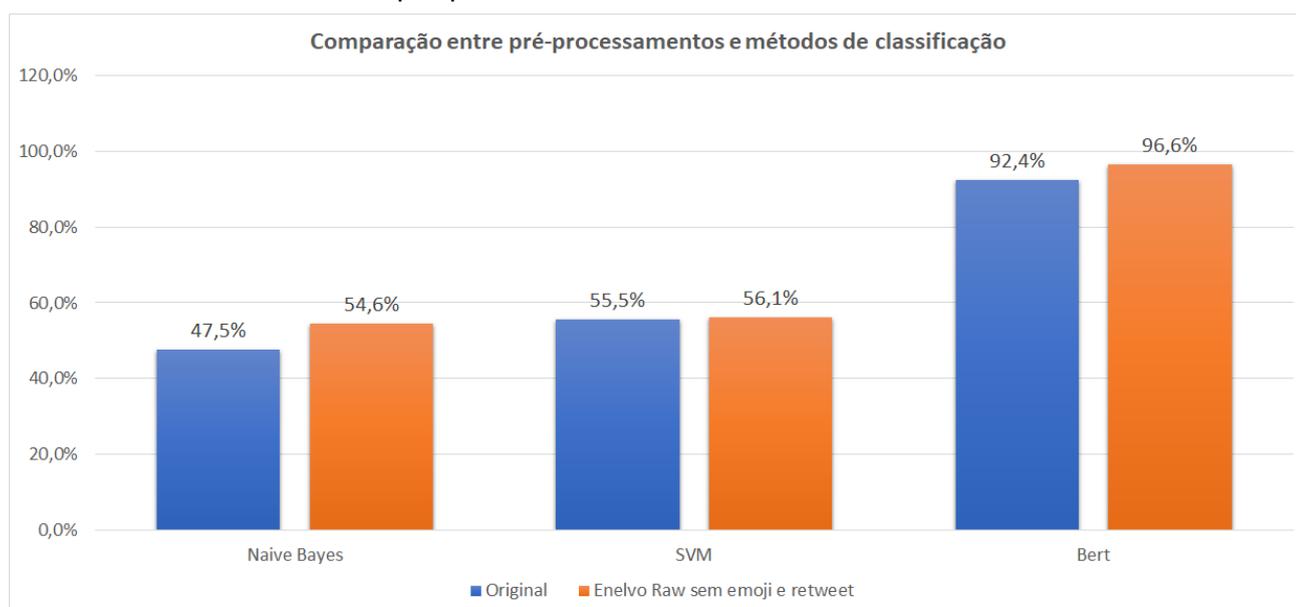
al. (2020) ficou em segundo lugar, com uma F1 de 92,4%. O pior caso ficou para a base que utilizava o Enelvo puro, com 74% de F1.

Vale ressaltar que as bases Enelvo Raw, que são as que não substituem as entidades como nome, número, *hashtags* e *retweets* por palavras-padrão, obtiveram resultados melhores que as bases que tiveram as entidades substituídas.

5.2 MÉTODOS DE CLASSIFICAÇÃO

Tendo as duas bases finais de pré-processamento escolhidas, ambas foram aplicadas nos três diferentes métodos de classificação, dois utilizados no trabalho de Cristiani et al. (2020), que são o Naive Bayes e o SVM, e o escolhido para o presente trabalho, o BERT. O resultado pode ser visto na Figura 5.

Figura 5: Comparação da F1 entre os diferentes tipos de classificação para os dois pré-processamentos escolhidos.



Fonte: Elaborado pelo autor.

Como é possível observar, o BERT se mostrou o melhor método de classificação para ambos os pré-processamentos, com um aumento percentual de F1 de 67% em comparação com o método SVM, que obteve o segundo melhor desempenho. Em comparação com o Naive Bayes, o Bert foi 82% melhor em F1, se mostrando o melhor método para classificação de textos com uma margem considerável.

Ambas as bases pré-processadas se mostraram próximas para os diferentes

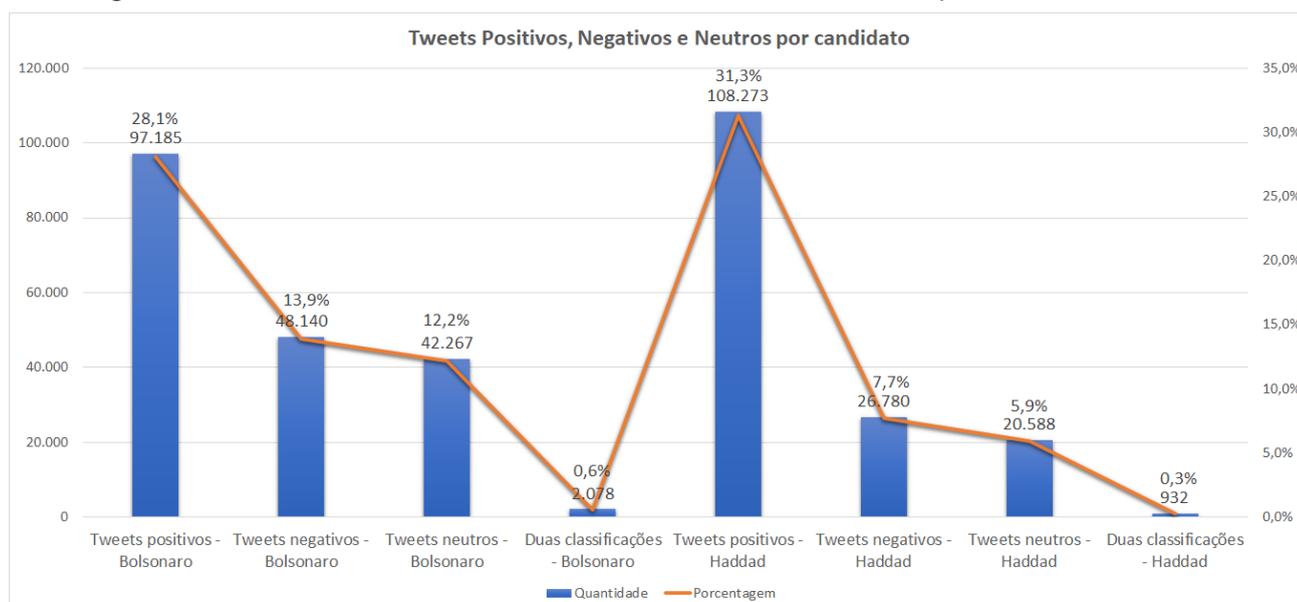
modelos, tendo a base Enelvo Raw sem emoji e *retweet* se saído um pouco a frente em F1 em todos os métodos, especialmente no Naive Bayes.

5.3 ANÁLISE DE TÓPICOS

Para a análise qualitativa de tópicos utilizando o BERTopic, o *corpus* com 369.800 *tweets* descrito na seção 4.1 foi processado pelo modelo de análise de polaridade treinado com o BERT. Dos 369.800 *tweets*, 346.243 obtiveram rótulos gerados pelo modelo, gerando um aproveitamento de 93,6%. Alguns *tweets* não foram rotulados pelo modelo por não terem se encaixado em nenhum dos rótulos durante o processamento. Destes, 54,7% são de *tweets* citando o candidato Jair Bolsonaro e 45,3% citando o candidato Fernando Haddad. Cada um dos *tweets* foi classificado em positivo, negativo, neutro ou em mais de um rótulo, devido ao algoritmo ser de classificação multirrótulo.

A quantidade de *tweets* rotulados como positivos (59,4%) é maior do que a de *tweets* negativos (21,6%) e neutros (18,1%). O candidato Fernando Haddad ficou com a maior quantidade de *tweets* positivos (31,3% contra 28,1%) e uma menor quantidade de *tweets* negativos (7,7% contra 13,9%) comparado ao candidato Jair Bolsonaro, o que indica sentimentos mais positivos nas falas sobre este candidato.

Figura 6: Quantidade de *tweets* classificados em cada rótulo por candidato.

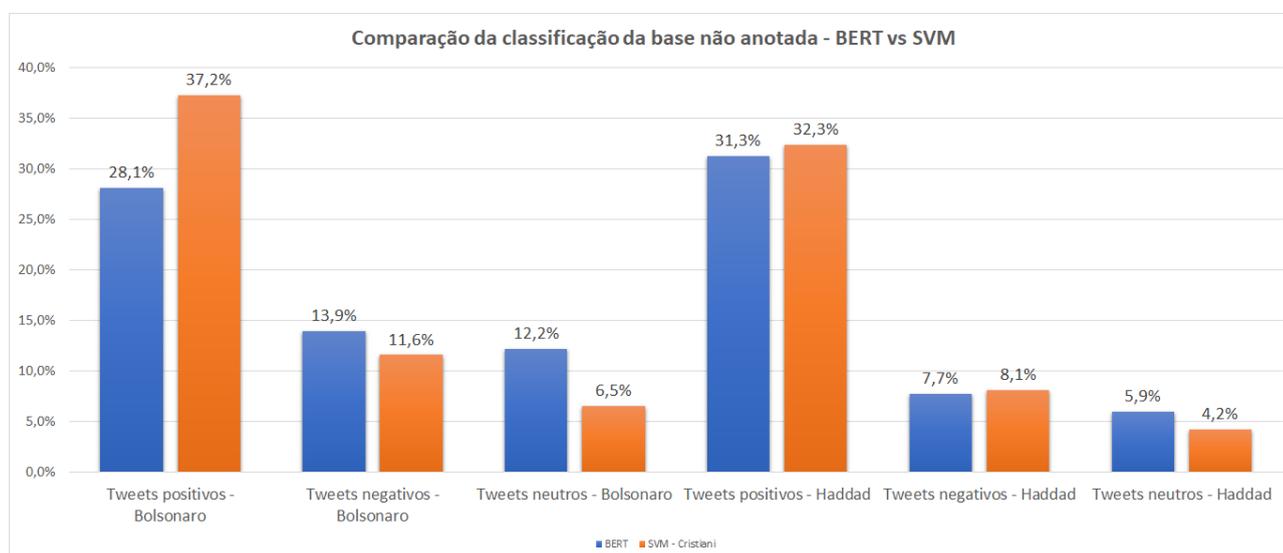


Fonte: Elaborado pelo autor.

Comparando os resultados com os obtidos por Cristiani et al. (2020), o modelo utilizando BERT classificou 9,1% dos comentários a menos como positivos para o candidato Jair Bolsonaro, e 2,3% a mais como negativos. Os comentários neutros também tiveram uma diferença, sendo 5,7% maiores na classificação realizada pelo BERT considerando a base toda.

Já com os comentários feitos sobre o candidato Fernando Haddad, o modelo utilizando BERT classificou 1% dos comentários a menos como positivos, 0,4% a menos como negativos e 1,7% a mais como neutros, se mostrando muito mais próximo dos resultados obtidos por Cristiani et al. (2020) utilizando SVM.

Figura 7: Porcentagem de *tweets* classificados em cada classe utilizando os métodos BERT e SVM.



Fonte: Elaborado pelo autor.

O *corpus* anotado automaticamente com polaridade foi, então, dividido em quatro conjuntos distintos de *tweets* para realizar a análise de tópicos: Bolsonaro+, Bolsonaro-, Haddad+ e Haddad-. Para cada um desses conjuntos foram gerados 20 tópicos com 5 palavras cada. Destes 20 tópicos, foram selecionados os 9 primeiros, numerados de 0 a 8, uma vez que quanto maior o número do tópico, menos frequente ele é. A medida de frequência de um tópico é calculada por meio da contagem do número de vezes em que as palavras presentes no tópico aparecem juntas no documento. As palavras dentro de cada tópico são ordenadas pela relevância gerada pelo c-TF-IDF. A Tabela 3 resume os principais resultados da análise de tópicos nestes quatro conjuntos.

Tabela 3: Análise dos principais tópicos citados positivamente e negativamente para cada

candidato.

Conjunto	Tópicos	Análise
Bolsonaro+	carvalho, hasselmann, paschoal acabouapiranhagempt, vitória, imparcial	Partes dos nomes de alguns dos principais aliados de Bolsonaro em 2018: Olavo de Carvalho, Joyce Hasselmann e Janaína Paschoal. Partes de slogans ou frases feitas de campanha, além de características mencionadas por possíveis apoiadores de Bolsonaro.
Bolsonaro-	homofóbicos, fascistas, desrespeitarem, ódio	Termos encontrados em <i>tweets</i> com polaridade (argumentos) negativos em relação a Bolsonaro.
Haddad+	haddadpresidente, trabalhando, vença, democracia, respeito, imparcial indecisos	Partes de slogans ou frases feitas de campanha, além de características mencionadas por possíveis apoiadores de Haddad A palavra “indecisos” também apareceu, uma vez que naquela época uma das chances de virada de Haddad era na conversão dos votos de pessoas indecisas.
Haddad-	xingando, meme	Poucas palavras que levam à conotação negativa apareceram relacionadas a Haddad, enquanto o restante foram palavras neutras, como “campanha”, “votando” e “falando”.

Fonte: Elaborado pelo autor.

Pode-se notar que o BERTopic se mostrou uma ferramenta bastante eficaz para a leitura de tópicos nas diferentes bases, mostrando tópicos que faziam sentido em todos os cenários. Entretanto, houveram diversos ruídos, havendo palavras de pouco significado, palavras ruidosas e de sentido oposto à base, como palavras com conotação positiva nas bases com tópicos negativos, como por exemplo as palavras “deus”, “gentil”, “parabens” e a *hashtag* “#acabouapiranhagempt” aparecendo entre os 10 principais tópicos negativos do candidato Jair Bolsonaro.

Os tópicos negativos do candidato Fernando Haddad foram os menos representativos. Poucas palavras que levam à conotação negativa apareceram, como “xingando” e “meme”, enquanto o restante foram palavras neutras, como “campanha”, “votando” e “falando” ou ruídos, como “pro”, “sorri” e “oci”.

É importante mencionar que uma das possíveis causas dos ruídos na geração de tópicos são *tweets* classificados incorretamente pelo modelo BERT, além de uma associação incorreta de *tweets* que continham duas ou mais *hashtags* ao candidato incorreto. Na Tabela 4 é possível ver casos de falsos positivos e

negativos para os dois candidatos, além de um *tweet* positivo que foi classificado como negativo, possivelmente devido à palavra “tiro”. Podemos ver também, na primeira linha, que o candidato correto a ser atrelado ao *tweet* seria o Fernando Haddad, e não o Jair Bolsonaro.

Tabela 4: Classificações incorretas de *tweets* pelo modelo BERT.

<i>Tweet</i>	Candidato	Rótulo
#ViraVirouHaddad fiz a minha parte #Elenão	bolsonaro	Positiva
Fogos de artifícios sendo já estourados na minha cidade! Fora PT! #MudaBrasil17 #AcabouaPiranhagemPT #OBrasilVota17 #FiscalizaBrasil	bolsonaro	Negativa
Panelaço contra Haddad - Mas o ibope jurava que Haddad vencia Bolsonaro em São Paulo #OBrasilBota17 #FiscalizaBrasil	haddad	Positiva
VOTEM, votar em branco ou nulo é dar um tiro no pé da democracia! #ViraVirouHaddad	haddad	Negativa

Fonte: Elaborado pelo autor.

Porém, vale ressaltar que a análise de tópicos foi a ponta final do experimento, carregando erros tanto de associação do *tweet* ao candidato correto quanto de falsos positivos e falsos negativos na classificação pelo BERT.

Dessa forma, é possível verificar que o algoritmo se mostrou uma ferramenta interessante de extração de tópicos na língua portuguesa, cenário que aparentou ser pouco explorado em pesquisas relacionadas.

6 CONCLUSÃO

O presente trabalho apresentou uma abordagem de análise de sentimentos (polaridade) utilizando o BERT, que utiliza redes neurais para a classificação de textos. O modelo treinado foi aplicado a mensagens obtidas da rede social *Twitter* referentes às eleições de 2018. Foram testadas diversas técnicas de pré-processamento para buscar a melhor F1, e posteriormente o melhor modelo foi aplicado um *corpus* maior para a classificação das mensagens. Feito isto, foi feita uma análise dos principais tópicos positivos e negativos envolvendo cada um dos candidatos.

Analisando os resultados obtidos para a classificação de polaridade, é possível afirmar que o modelo BERT apresentou um salto considerável de F1 (96%) em comparação a métodos mais tradicionais como o SVM (56%) e o Naive Bayes (54%), mostrando-se como um método excelente para classificação e análise de sentimentos em textos da língua portuguesa. Com relação à análise de tópicos, o algoritmo BERTopic conseguiu extrair diversos tópicos relevantes, porém gerou alguns ruídos.

Considerando os trabalhos futuros, há uma margem para melhorias nos métodos de pré-processamento, visto que o uso do Enelvo se tornou computacionalmente inviável para grandes quantidades de texto. Pode-se estudar métodos de melhoria de desempenho deste algoritmo utilizando técnicas de paralelização, por exemplo. Também existe uma margem para melhorias na extração de tópicos, buscando métodos de identificação de entidades e removendo as entidades neutras ou com baixo significado.

Vale ressaltar que a metodologia e as ferramentas empregadas neste trabalho podem ser utilizadas em outros tipos de pesquisa relacionadas a conteúdos textuais em língua portuguesa. Assim, espera-se que o modelo desenvolvido possa sustentar novas pesquisas na área de processamento de linguagem natural em português, principalmente no tocante ao tema da política.

Esse trabalho poderá servir como base teórica para o estudo dos algoritmos e técnicas envolvidos na construção do código, buscando influenciar pesquisas futuras da comunidade científica. Visando que o código e os resultados aqui apresentados sirvam de base para a validação de estudos em PLN, análise de sentimentos e de tópicos em textos, o código produzido neste trabalho encontra-se

disponível em: <https://github.com/LALIC-UFSCar/tcc_leonardo_capellaro>.

REFERÊNCIAS

- BECKER, K.; TUMITAN, D. **Introdução à Mineração de Opiniões: Conceitos, Aplicações e Desafios**. Anais do 28th Brazilian Symposium on Databases. p. 27–52. 2013.
- BERTAGLIA, T. F. C.; NUNES, M. d. G. V. **Exploring word embeddings for unsupervised textual user-generated content normalization**. Anais do 2nd Workshop on Noisy User-generated Text (WNUT). p. 112–120. 2016.
- BOUAZIZI, M.; OHTSUKI, T.. **A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter**. 2017.
- CAMPELLO, R. J. G. B.; MOULAVI, D.; SANDER, J. **Advances in Knowledge Discovery and Data Mining**. Berlin: Springer, 2013.
- CHRISTHIE W. et al. **Deteccão de posicionamento em tweets sobre política no contexto brasileiro**. Anais do VII Brazilian Workshop on Social Network Analysis and Mining. Porto Alegre, RS, Brasil. SBC. 2018.
- CIGANOVIC I.; PLUSKOSKI A.; JOVANOVIC, M.D. **Autonomous car driving - one possible implementation using machine learning algorithm**. Anais do 5th international conference IcETRAN. 2018.
- CRISTIANI, A.; LIEIRA, D.; CAMARGO, H. **A sentiment analysis of brazilian elections tweets**. Anais do VIII Symposium on Knowledge Discovery, Mining and Learning. p. 153–160. Porto Alegre, RS, Brasil. SBC. 2020.
- DEVLIN, J. et al. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. Anais do NAACL - HLT 2019. p. 4171–4186. 2019.
- DOU, W.; LIU, S. **Topic and time-oriented visual text analysis**. IEEE ComputerGraphics and Applications. p.8–13. 2016.
- FEIGENBAUM, A. **Expert Systems: Principles and Practice**. 1992.
- GROOTENDORST, M. (2020). **BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics**. Disponível em <<https://zenodo.org/record/5574296>>. Acesso em: 09 set. 2021
- GROSSI, E.; BUSCEMA, M. **Introduction to artificial neural networks. European journal of gastroenterology & hepatology**. 2008.
- HUANG, T.; SMITH, C. **The History of Artificial Intelligence**. Disponível em: <<https://courses.cs.washington.edu/courses/csep590/06au/projects/history-ai.pdf>>. Acesso em: 8 ago. 2021
- McCULLOCH, W. S.; PITTS, W. **A logical calculus of the ideas immanent in nervous activity**. Bulletin of Mathematical Biophysics. p. 115–133. 1943.
- MOREIRA, R.; VAZ DE MELO, P; PAPPA, G. **Elite versus mass polarization on the brazilian impeachment proceedings of 2016**. Anais do Social Network Analysis and Mining. 2020.
- MURPHY, K. P. **Machine learning: a probabilistic perspective**. Cambridge, MA. MIT Press. 2012.
- PEREIRA, J. G. **Análise de sentimentos da população brasileira em relação a eleição presidencial de 2018 através da rede social twitter**. Trabalho de Conclusão de Curso (Bacharelado) - Universidade Federal do Rio Grande do

Norte. Departamento de Computação e Tecnologia. 2019.

PINTO, M. A. S. et al. **Relacionando modelagem de tópicos e classificação de sentimentos para análise de mensagens do twitter durante a pandemia da covid-19**. Anais Estendidos do XXVI Simpósio Brasileiro de Sistemas Multimídia e Web. p. 61–64, Porto Alegre, RS, Brasil. SBC. 2020.

POURJALALI, S. **THE AI Future: Alexa, Siri, Google Assistant, Cortana, Samantha and Their Cousins**. Disponível em: <https://www.linkedin.com/pulse/future-alex-google-assistant-siri-cortana-samantha-pourjalali/>. Acesso em: 5 ago. 2021.

ROSENBLATT, F. **The perceptron: A probabilistic model for information storage and organization in the brain**. Psychological Review, 65(6). p. 386–408. 1958.

SADOUGHI, F. et al. **Artificial intelligence methods for the diagnosis of breast cancer by image processing: A review**. Breast Cancer: Targets and Therapy. Volume 10. p. 219-230. 2018.

SAH, S. **Machine Learning: A Review of Learning Types**. 2020

SALES, M. L.; BARBOSA, M. W. **Uma avaliação do potencial de uso dos dados do Twitter para a predição do resultado de eleições: O caso das eleições presidenciais brasileiras de 2018**. Revista de Informática Aplicada - RIA. p. 30–43. 2019.

SILVEIRA, R. et al. **Topic modelling of legal documents via legal-bert1**. São Paulo, SP.. 2021.

SAMUEL, A. L. **Some Studies in Machine Learning Using the Game of Checkers**. IBM Journal of Research and Development, vol. 3, no. 3. p. 210-229. Julho 1959.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. **BERTimbau: pretrained BERT models for Brazilian Portuguese**. In **Lecture Notes in Computer Science**. Anais do 9th Brazilian Conference on Intelligent Systems (BRACIS). Volume 12319. p. 403–417. 2020.

SURYA, M. **The Decade of Artificial Intelligence**. Disponível em: <<https://towardsdatascience.com/the-decade-of-artificial-intelligence-6fcaf2fae473>>. Acesso em: 07 set. 2021

PAIVA, F. **A história do nascimento da BIA**. Disponível em: <<https://www.mobiletime.com.br/noticias/04/05/2018/a-historia-do-nascimento-da-bia/>>. Acesso em: 25 ago. 2021