ISSN: 2237-826X
ARTIGO | PAPER

Detecção automática de *fake news* em *tweets* em períodos eleitorais

Automatic detection of fake news in tweets during election periods

Rafaela de Amorim Barbosa Silva [1], Eanes Torres Pereira [2], Sylvia Iasulaitis [3]

[1] *Universidade Federal de Campina Grande, Campina Grande-Paraíba, Brasil. ORCID: https://orcid.org/0000-0003-2568-3218*

[2] *Universidade Federal de Campina Grande, Campina Grande-Paraíba, Brasil. ORCID: https://orcid.org/0000-0002-9717-794X*

[3] *Universidade Federal de São Carlos, São Carlos-São Paulo, Brasil. ORCID: https://orcid.org/0000-0002-3526-1003*

Autor para correspondência/*Mail to:* Rafaela de Amorim Barbosa Silva, rafaela.amorim@copin.ufcg.edu.br

Recebido/*Submitted*: 03 de maio de 2024; **Aceito/*Approved*:** 01 de outubro de 2025

Resumo

Introdução: O fenômeno da desinformação consiste na propagação de informações não verídicas e falsificadas com o objetivo de obter ganhos financeiros, manipular a opinião pública, enfraquecer um campo político, alterar as relações de poder, fortalecer grupos de ódio ou alimentar preconceitos por meio de representações deliberadamente distorcidas. Com o advento das mídias sociais, percebe-se o aumento do consumo de notícias online além de alterações no comportamento informacional que levaram a um aumento no acesso e compartilhamento de *fake news*. Tais notícias frequentemente são concebidas para se parecerem com notícias verdadeiras e são criadas e disseminadas muito rapidamente. **Método:** Neste trabalho propomos uma metodologia para detecção de notícias falsas usando redes neurais profundas, com uma amostra de mais de 2 milhões de *tweets* de um conjunto de dados próprio, coletado com a API do Twitter durante as eleições presidenciais brasileiras no pleito de 2022. Os *tweets* foram rotulados automaticamente por um modelo de supervisão fraca. **Resultados:** Os resultados obtidos com o modelo foram F1-score de 98% em *tweets* de notícias não falsas e F1-score de 47% em *tweets* contendo notícias falsas. A área sob a curva ROC foi de 0.848, considerado um valor que mostra potencial. **Conclusão:** São promissores os resultados de modelos de redes neurais artificiais para agilizar o tão necessário trabalho de verificação de veracidade de notícias, especialmente em pleitos eleitorais, que exigem rápida detecção de *fake news*. Contudo, ainda é um desafio trabalhar com grandes volumes de dados não rotulados, como aqueles utilizados nesta pesquisa. Os principais tópicos de *fake news* encontrados estavam relacionados a valores morais e religiosos, pautas econômicas, idoneidade de instituições, endosso de figuras públicas, associação de adversários e partidos ao crime e negacionismo científico.

Palavras-chave: *fake news*; *tweets*; Eleições; Detecção de notícias falsas; Inteligência artificial; Redes neurais convolucionais.

Abstract

Introduction: *The phenomenon of disinformation consists of the propagation of untrue and falsified information to obtain financial gains, manipulate public opinion, weaken a political field, alter power relations, strengthen hate groups, or feed prejudices through representations deliberately distorted. With the advent of social media, online news consumption has increased, and changes in information behavior have led to greater access to and sharing of fake news. Such newsis often designed to look like real newsand is created and disseminated very quickly.* ***Method:*** *In this work, we propose a methodology for detecting fake news using deep neural networks, with a sample of more than 2 million tweets from our own dataset, collected with the Twitter API during the Brazilian presidential elections in 2022. tweets were automatically labeled by a weak supervision model.* ***Results:*** *The results obtained with the model were an F1-score of 98% in tweets with non-fake newsand an F1-score of 47% in tweets containing fake news. The area under the ROC curve was 0.848, considered a value that shows potential.* ***Conclusion:*** *The results of artificial neural network models are promising to speed up the much-needed work of verifying the veracity of news, especially in elections, which require rapid detection of fake news. However, it remains a challenge to work with large volumes of unlabeled data, such as those used in this research. The main fake news topics found were related to moral and religious values, economic agendas, the suitability of institutions, endorsement of public figures, association of opponents, and parties with crime and scientific denialism.*

Keywords: *fake news; tweets; Elections; fake news detection; Artificial Intelligence; Convolutional neural networks.*

INTRODUÇÃO

O fenômeno da desinformação consiste na propagação de conteúdos não verídicos, falsificados com o objetivo de obter ganhos financeiros, influenciar opinião, enfraquecer um campo político, alterar as relações de poder, fortalecer grupos de ódio ou alimentar preconceitos por meio de representações deliberadamente distorcidas (Huyghe, 2016). Em geral, as informações irreais invocam uma reação emocional de quem consome a notícia .

A desinformação desvirtua o conteúdo e consiste no perfeito oposto do real significado de informação. Nos termos de Eugênio Bucci (Rollemberg, 2022), ela anestesia e desativa a razão. Na mesma direção, Santaella (2023) qualifica a desinformação como antônimo de informação.

Apesar de não ser um fenômeno exclusivo das campanhas eleitorais, é predominantemente durante este período que as *fake news* adquirem novas dimensões. A expressão, livremente traduzida como "notícia falsa", ganhou destaque durante as eleições norte-americanas de 2016 (Bovet & Makse, 2019; Allcott & Gentzkow, 2017).

Embora não seja o objetivo deste artigo o aprofundamento teórico a respeito do conceito de 'desinformação' e do termo usual *fake news*, popularizado para designar a disseminação de conteúdos inverídicos em larga escala, este prelúdio mostra-se fundamental para sinalizar que, por trás da adoção do termo *fake news*, há espaço para ampla discussão na teoria da informação, uma vez que se a história é fake, por conseguinte não é *news*, ou seja, se é falsa, não é notícia e muito menos informação (Rizzo & Becker, 2020).

Como este debate excederia em muito o escopo deste artigo, nesta pesquisa será adotada, por questões operacionais, a expressão *fake news*, pelo fato de que tal expressão, nas eleições recentes nos Estados Unidos e no Brasil, passou a ser utilizada como ferramenta de propaganda tanto para rejeitar opiniões críticas (Maci, Demata, Seargeant, & McGlashan, 2023) e realizar campanhas negativas a adversários (Iasulaitis & Vieira, 2022), quanto pelas agências de *fact-checking*, conforme será aprofundado nas próximas seções deste artigo.

O termo *fake news* se popularizou no Brasil, principalmente por conta das eleições presidenciais que ocorreram no final de 2018. Seguindo o modelo de campanha de Donald Trump em 2016, foi observado no pleito brasileiro um amplo uso de redes sociais digitais como veículos de propagação de desinformação sem os filtros dos *gatekeepers*, beneficiando principalmente o então candidato Jair Bolsonaro, que contou com a colaboração do estrategista Steve Bannon, da *Cambridge Analytica* (Ricard & Medeiros, 2020). Estudos identificaram também a existência de *bots* e de contas falsas para propagação de *fake news* (Bradshaw et al., 2021).

Durante a eleição presidencial brasileira de 2022, o cenário de disseminação de *fake news* se repetiu e muitos eleitores, principalmente do presidenciável Jair Bolsonaro , acreditaram em *fake news*. Centenas de informações falsas marcaram este pleito eleitoral, sobre os mais diferentes temas e com foco nos diversos nichos eleitorais.

O pleito presidencial de 2022 foi a disputa mais acirrada no Brasil democrático até então e teve a menor diferença percentual da história entre dois candidatos que levaram a competição ao segundo turno. Luiz Inácio Lula da Silva venceu o então candidato à reeleição, Jair Bolsonaro, com menos de dois pontos percentuais de diferença, pouco mais de 2 milhões de votos. A divisão do Brasil em dois polos políticos, esquerda e direita, tornou-se bastante evidente nas redes sociais e se acentuou com a aproximação do período eleitoral.

Neste contexto, foi ampla a disseminação de conteúdos falsos em plataformas como o Twitter (atualmente denominado X), caracterizado por ser uma rede social de *microblogging*, onde os usuários podem publicar textos de até no máximo 280 caracteres por vez. Em 2022, a plataforma contava com mais de 19 milhões de usuários brasileiros, número que colocava o Brasil na quarta maior base de usuários de Twitter/X do mundo . Uma abundância de informações (e desinformações) circula diariamente nesta rede social digital, com ampla disseminação incentivada pela própria plataforma conforme alguma publicação "viraliza".

Estudos vêm demonstrando que as *fake news* têm maiores chances de viralizar em comparação às notícias verdadeiras (Santana Júnior, Albuquerque, Queiroz, & Lima, 2014). Conforme o estudo empírico de Vosoughi, Roy, e Aral (2018) uma história falsificada tem 70% mais chances de viralizar do que uma verdadeira e o alcance também é maior: enquanto os conteúdos verdadeiros, em geral, alcançaram 1.000 pessoas, as principais mensagens falsas eram lidas por até 100.000 pessoas.

Além dos conteúdos falsos serem espalhados mais rapidamente do que notícias verdadeiras, há o agravante de que são disseminados mais rapidamente do que especialistas conseguem identificar e provar de maneira fidedigna que são irreais.

No Brasil, existem agências e projetos comprometidos a checar a veracidade de notícias duvidosas. Contudo, a quantidade e a velocidade com as quais as *fake news* são produzidas e propagadas não permite que a checagem manual seja realizada antes que a desinformação já tenha causado muitos danos. Isto porque a checagem manual de fatos pode levar um tempo considerável.

Embora existam diversas abordagens para detecção de *fake news* em redes sociais, grande parte dos estudos concentra-se em mídia de língua inglesa e em contextos eleitorais estrangeiros, principalmente dos Estados Unidos. Há uma carência de metodologias específicas para o cenário brasileiro, considerando especialmente a alta polarização e o alto volume de desinformação nas eleições e a dificuldade de trabalhar com grandes bases de dados não rotulados. Este trabalho propõe superar essas limitações por meio da criação de uma metodologia adaptada ao português e capaz de lidar com o desafio de rótulos escassos e grandes volumes de dados.

Assim, mostra-se de fundamental importância automatizar a verificação de veracidade das notícias. Para ajudar jornalistas, instituições democráticas e a população em geral nessa tarefa de identificação de notícias falsas, a inteligência artificial pode se mostrar bastante útil, podendo-se empregar modelos de redes neurais para agilizar este trabalho de verificação de notícias.

As redes neurais podem ser usadas em uma ampla variedade de tarefas, como reconhecimento de padrões, classificação, processamento de linguagem natural, visão computacional e muito mais. Elas são capazes de aprender e generalizar a partir de exemplos fornecidos durante o treinamento. Este trabalho busca contribuir nesta direção, tendo como objetivos: (1) construir uma rede neural profunda que consiga identificar a presença de *fake news* em um *tweet* e (2) criar um método para lidar com grandes bases de dados não rotulados por

meio de um modelo de aprendizado que usa funções ruidosas para aprender padrões em um *tweet*. A proposta diferencia-se dos estudos existentes não apenas pela escolha do idioma e do contexto político, mas também pelo uso combinado de heurísticas temáticas com base em artigos de detecção de notícias enganosas ou tendenciosas, o uso de supervisão fraca para rotular um grande conjunto de dados, e uma arquitetura de rede neural convolucional treinada com textos curtos e informais como os *tweets*.

Serão utilizadas publicações do Twitter/X coletadas durante as eleições presidenciais de 2022 para treinar um modelo de aprendizagem profunda que consiga identificar *fake news* em textos curtos. Trata-se de um *dataset* próprio, composto por 282 milhões de *tweets*, denominado ITED-Br, cujo processo de coleta e características da base são apresentados de forma minuciosa em Iasulaitis et al. (2025).

REVISÃO DA LITERATURA

Mendonça, Freitas, Aggio, e Santos (2023) citam o Fórum Econômico Mundial para enfatizar que a propagação de *fake news* é uma das principais ameaças à sociedade. Os autores agrupam os antídotos contra *fake news* em quatro categorias, sendo uma delas a categoria técnica. Mendonça et al. (2023) enfatizam que, embora soluções técnicas como softwares para detecção de notícias falsas amenizem o problema de sua propagação e possam, até certo ponto, constranger a circulação, tais soluções não atacam a raiz do problema da desinformação que, segundo os autores, é um problema social e político. Outros fatores que dificultam o emprego de algoritmos de inteligência artificial para detectar *fake news* é o fato que a maioria das abordagens usadas são do tipo "aprendizagem supervisionada" e isso requer um rotulamento manual. Apenas a existência de grandes grupos jornalísticos realizando checagem de fatos não será capaz de rotular todos os dados nem solucionar o problema (Mendonça et al., 2023). Uma medida complementar mencionada por Mendonça et al. (2023) para lidar com as *fake news* pela raiz seria a indução de comportamentos dos cidadãos, o que passa pela educação de crianças sobre o tema de *fake news*.

As *fake news* apresentam características ambíguas, uma vez que são escritas intencionalmente para enganar os leitores e fazer com que acreditem em declarações falsas, o que torna difícil e não trivial detectá-las com base no conteúdo da notícia.

Existem esforços que auxiliam na detecção automática de informações falsas para ajudar no controle da praga da desinformação, podendo-se citar os trabalhos de Shu, Sliva, Wang, Tang, e Liu (2017), a ferramenta brasileira de detecção de *fake news* desenvolvida por pesquisadores da USP , bem como estudos sob perspectivas causais da influência das *fake news* no debate sobre as eleições presidenciais estadunidenses de 2016 (Bovet & Makse, 2019). Seguindo uma abordagem similar à proposta do presente trabalho, Lorenceti e Salton (2022) utilizam textos de *tweets* para treinar diferentes modelos de classificação, porém, devido à um volume de dados muito pequeno, não conseguiram resultados satisfatórios para classificar como notícias falsas.

O artigo de Shu et al. (2017) trata da detecção de notícias falsas em redes sociais , fazendo uma revisão abrangente sobre o assunto, a partir de algoritmos existentes sob a perspectiva da mineração de dados e métricas de avaliação e conjuntos de dados representativos. O artigo sugere usar informações como o engajamento social do usuário e das postagens para ajudar na tomada de decisões, o que ainda se mostra uma tarefa desafiadora, pois o engajamento social dos usuários com as *fake news* gera dados volumosos, incompletos, não estruturados e ruidosos.

O estudo de Saleh, Alharbi, e Alsamhi (2021) propõe uma arquitetura de Rede Convolucional otimizada para detecção de *fake news*. É um modelo de aprendizagem supervisionada, denominado OPCNN-Fake, em inglês: *an Optimal CNN model for fake news detection*. Este modelo usa várias camadas para extrair características de alto nível e de baixo nível. A otimização do modelo se fez usando a técnica de otimização *hyperopt* para selecionar os melhores valores para os hiperparâmetros.

Na pesquisa supracitada, foram utilizados para treino e teste quatro conjuntos de dados padrão: *Fake news Net*, FA-KES5, ISOT e um conjunto de dados do *Kaggle*. De cada conjunto de dados foram retiradas amostras que correspondem à fração de 80% para treinamento e 20% para teste, e também foi utilizada validação cruzada 10-fold no treinamento. O modelo foi avaliado usando métricas padrões: acurácia, precisão, revocação e F1-score. Essas métricas foram usadas para compará-lo em relação a quatro outros modelos de aprendizagem profunda (LSTM de uma camada, LSTM de duas camadas, RNN de uma camada, RNN de duas camadas) e Regressão logística, K vizinhos mais próximos (K *Nearest Neighbor* – KNN), Floresta Aleatória, Máquina de Vetores de Suporte e *Naive Bayes*.

A arquitetura OPCNN-Fake teve resultados melhores em comparação com todos esses modelos em todos os quatro *dataset*s utilizados, tanto no conjunto de treinamento com validação cruzada, quanto no conjunto de teste. Por esse motivo, a arquitetura de classificação utilizada nesta pesquisa foi baseada no OPCNN-Fake proposto em Saleh et al. (2021).

No artigo de Ratner et al. (2019) é proposto um *framework* baseado na abordagem de supervisão fraca, que pode rotular grandes *dataset*s rapidamente, fornecendo rótulos mais ruidosos, porém mais baratos que o rotulamento

manual. Essas fontes de supervisão fraca têm acurácias diversas e desconhecidas, além de poderem fornecer rótulos correlacionados, rotular diferentes tarefas ou serem aplicados em diferentes níveis de granularidade.

A framework promete integrar e modelar essas fontes de supervisão fraca, considerando-as como rótulos de diferentes sub-tarefas relacionadas de um problema. O modelo de rotulagem utilizado nesta pesquisa, *LabelModel* , é baseado nesta abordagem. Ele aprende um modelo estatístico que estima as probabilidades de rótulo a partir das informações das fontes ruidosas de rótulos.

Silva et al. (2024) apresentam uma base de dados coletada do Twitter durante as eleições de 2022 no Brasil contendo *tweets* sobre política. Os autores utilizaram a API do Twitter (atual X) para coletar os dados e realizaram pré-processamento antes de disponibilizá-la para download. Para determinar os temas relevantes, fontes de notícias foram monitoradas. Alguns eventos relevantes foram selecionados, os quais foram usados para determinar palavras-chave e hashtags que foram usadas nas queries de busca. Os dados foram coletados no período de 27 de abril de 2022 a 23 de janeiro de 2023 e foram coletados 9,3 milhões de *tweets*. Apenas quatro campos dos *tweets* foram armazenados: data de criação, identificação do autor, identificação do *tweet* que iniciou a conversa e identificador do *tweet* mencionado. A base de dados de Silva et al. (2024) possui o mesmo contexto e objetivo que ITED-Br, porém além de ter um volume de *tweets* muito menor, ela também apresenta bem menos metadados dos *tweets* em questão.

Diferentemente dos trabalhos anteriores, que focam majoritariamente em notícias em inglês ou em artigos jornalísticos, esta pesquisa concentra-se na detecção automática de *fake news* em *tweets* escritos em português durante um contexto eleitoral altamente polarizado. Esses dados são caracterizados por serem muito curtos e por poderem ser escritos por qualquer pessoa, abrindo brechas para usuários falsos cujo único propósito é causar caos, alimentar reações negativas e obter engajamento. Além disso, este estudo inova ao trabalhar com um *dataset* próprio, ruidoso e enorme, usando técnicas de supervisão fraca para viabilizar a rotulagem eficiente em larga escala.

METODOLOGIA

Nesta seção encontram-se: a descrição do conjunto de dados usado nessa pesquisa, que inclui *tweets* e notícias de checagem de *fake news*, o processo de limpeza e exploração dos dados; o agrupamento das notícias para selecionar palavras-chave importantes do contexto político em análise; a rotulagem da base de *tweets* usando um modelo de aprendizado com supervisão fraca; e, por fim, a descrição do modelo de classificação da presença de *fake news* em um *tweet*.

Descrição da base de dados de *tweets*

A base de dados de *tweets* utilizada nesta pesquisa foi coletada pelo grupo de pesquisa do qual os autores fazem parte, o Interfaces. Os *tweets* foram coletados a partir do dia 20 de Junho de 2022 até o dia 24 de Fevereiro de 2023, cobrindo o período pré-eleitoral, eleitoral, pós-eleitoral e os episódios de ataques à sede dos Três Poderes em Brasília em Janeiro de 2023.

O *dataset*, denominado *The Interfaces Twitter Elections dataset*, conhecido pelo acrônimo ITED-Br, é composto por 282 milhões de *tweets*, e é a terceira maior base de dados de *tweets* com propósitos políticos do mundo.

Cada *tweet*, além do texto, contém um identificador único, identificador do autor, data de criação, fonte (mobile, web, etc.), linguagem, contagem de *likes, retweets* e de quote *retweets*, lista de usuários mencionados, lista de *hashtags*, lista de *urls* (podendo as 3 listas serem vazias) e por fim, identificador de conversa e/ou identificador de referência. Cada *tweet* tem seu próprio identificador de conversa, que serve para ser referenciado (por meio do identificador de referência) por um *tweet*-resposta, e o *tweet*-resposta, por sua vez, pode ser respondido do mesmo modo.

Há, ainda, uma base com usuários únicos que são autores dos *tweets* da primeira base, a qual contém o identificador único de usuário, que é referenciado na base de *tweets* como id de autor, nome de usuário, data de criação da conta, indicação da existência de proteção na conta, localização informada na conta, contagem de seguidores, contagem de contas que o usuário segue, contagem de postagens e, por fim, se a conta é verificada ou não. Durante o período de coleta não existia "verificado pago- a assinatura do Twitter Blue , uma mensalidade que o usuário pode pagar para ter alguns benefícios na plataforma -, até 2022 existia apenas o símbolo de "verificado pela plataforma".

O corpus conta com mais de 282 milhões de *tweets* não rotulados, majoritariamente em português do Brasil, e todos os *tweets* estão relacionados às eleições presidenciais de alguma forma, porém em contextos diferentes. Embora o corpus contenha dados sobre os demais candidatos que concorreram no pleito, o foco foram as postagens que mencionavam os protagonistas que levaram a competição ao segundo turno, no caso os presidenciáveis Bolsonaro e Lula, identificadas como as consultas *query_ lula* e *query_ bolsonaro*.

Optou-se por não usar todos os dados da base por alguns motivos. Primeiramente, para manter um escopo melhor definido: pois há vários sub-tópicos, apesar de todos estarem ligados às eleições de 2022 de alguma maneira. Portanto, não serão abordadas as discussões sobre pós-eleições nem sobre os atos golpistas. Em segundo lugar, para reduzir a quantidade de dados a serem processados. Por último, não foi considerado o subgrupo do número dos candidatos, por se tratar de uma consulta extremamente abrangente que inclui uma grande quantidade de *tweets* sem relação com as eleições, o que demandaria uma extensa limpeza dos dados para possibilitar a análise desta consulta.

Esses *tweets* foram coletados por meio da API do Twitter usando como filtro palavras-chaves relacionadas a tópicos relevantes das eleições:

- Os presidenciáveis mais relevantes: Bolsonaro, Lula – Período 20/06/22 a 31/01/23

Visando ampliar a abrangência na coleta de *tweets*, as consultas foram feitas com os nomes dos candidatos, apelidos populares e a hashtag #nome_do_candidato. Além disso, se o *tweet* fosse uma resposta ou um *retweet* com comentário, então também foi incluído o *tweet* que foi referenciado por ele (mesmo se não tiver nenhum apelido de candidato).

1) query_bolsonaro: # bolsonaro OR jair OR bolsonaro OR bozo OR biroliro OR "tchutchuca do centrao "OR bonoro OR capitao OR genocida OR mito OR bolsomito OR bolsolixo OR bolsotrump OR messias OR patriota OR b22 OR b17 OR brocha OR imbrochavel OR ma 'conaro

2) query_lula: # lula OR lula OR "ex presidiario "OR lulalivre OR "9 dedos "OR luladrao OR lulaladrao OR lulinha OR nine OR luis inacio OR cachaceiro OR "sapo barbudo "OR lulao OR l13 OR "faz o L "OR lulindo OR metalurgico OR lulalkimin

- Perfil dos candidatos: foram coletadas todas as postagens dos presidenciáveis mais relevantes com referências às respostas dessas postagens - Período 20/06/22 a 31/01/23

- A API do Twitter disponibiliza um identificador único chamado de conversation_id que corresponde uma postagem às respostas/comentários (*quote retweet*) que ela receber. As respostas referenciam a postagem com a variável reference_id.

A média de *tweets* diários do mês de outubro é claramente maior que o restante do período observado, conforme exibido na Figura 1. O total bruto de *tweets* sobre Lula e sobre Bolsonaro somente no mês de outubro é de quase 27 milhões de *tweets*.

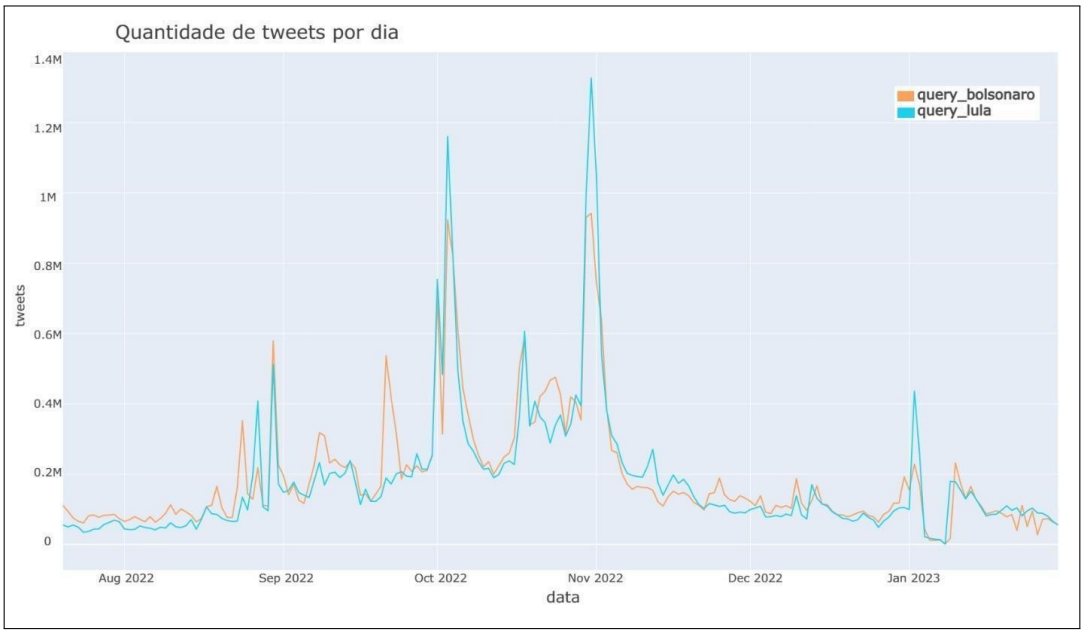

Figura 1. Quantidade de *tweets* coletados por dia de postagem
Nota: A linha laranja indica a quantidade de *tweets* citando o Bolsonaro, e a azul indica a quantidade de *tweets* citando o Lula.

A base utilizada nesta pesquisa continha exatamente 26.900.769 *tweets* antes da limpeza, sendo 13.358.707 *tweets* que mencionam Lula e 13.542.062 que mencionam Bolsonaro. Previa-se que grande parte desses *tweets* tivessem 0 likes, o que de fato corresponde à mediana de 0 likes, *retweets* ou *retweets* com comentário (mais estatísticas são apresentadas na Tabela 1).

Foi realizada uma filtragem por *tweets* com 10 ou mais likes e pelo menos 2 *retweets*, restando um total de 2.100.629 *tweets* de 235.114 autores diferentes (em média 8,9 *tweets* por usuário). Tal escolha foi realizada por

dois motivos: primeiro, como o principal objetivo das *fake news* é a disseminação, se não há engajamento, o objetivo não é plenamente atingido e não atende aos propósitos desta investigação; e em segundo lugar, visando a economia de recursos computacionais. As novas estatísticas após a filtragem encontram-se na Tabela 2.

Estatística	Tipo	Valor
Média	Like	1294
	Re *tweet*	39
	RT c/ coment.	6
Desvio padrão	Like	5511
	Re *tweet*	696
	RT c/ coment.	228
Percentil 25%	Like	0
	Re *tweet*	0
	RT c/ coment.	0
Mediana	Like	0
	Re *tweet*	0
	RT c/ coment.	0
Percentil 75%	Like	2
	Re *tweet*	0
	RT c/ coment.	0

Tabela 1. Estatísticas antes de filtrar os dados

Estatística	Tipo	Valor
Média	Like	3365
	Re *tweet*	517
	RT c/ coment.	65
Desvio padrão	Like	15090
	Re *tweet*	2489
	RT c/ coment.	527
Percentil 25%	Like	37
	Re *tweet*	5
	RT c/ coment.	0
Mediana	Like	162
	Re *tweet*	18
	RT c/ coment.	2
Percentil 75%	Like	1017
	Re *tweet*	128
	RT c/ coment.	11

Tabela 2. Estatísticas após filtrar os dados

Para esta pesquisa, foi necessário dividir o conjunto de dados em várias etapas. Em um primeiro momento foi feita a separação do *dataset* para ser utilizado no classificador e no modelo de rotulagem com o auxílio da função train_test_split() do Scikit-Learn. Dividiu-se o conjunto de dados em 80% (1.680.503) para o classificador e 20% (420.126) para o rotulador. O conjunto do classificador foi novamente dividido na proporção de 80/20, para ser usado nos estágios de treinamento e teste, respectivamente. Já o conjunto do rotulador foi dividido da seguinte forma: 1046 amostras foram rotuladas manualmente (de 1050, 4 amostras foram descartadas por ambiguidade no rótulo), e o restante dos dados destinou-se ao treinamento do modelo. O modelo rotulador adotado é uma abordagem de supervisão fraca, baseada na biblioteca Snorkel para Python3.

Após ser treinado, o modelo rotulador foi utilizado para rotular 100% do conjunto de dados do classificador, e algumas amostras não foram rotuladas, quando o modelo se abstém de atribuir um rótulo (isto é detalhado na seção de rotulagem dos *tweets*).

Quanto à rotulagem manual, foi necessário que os pesquisadores realizassem a rotulagem manual de algumas amostras da base de dados para validar o modelo de rotulagem.

Para determinar se um *tweet* continha uma "notícia falsa"ou não, foram considerados os seguintes aspectos:

- A definição de uma *fake news*: afirmações falsas que, sobretudo, trazem algum tipo de vantagem política a algum dos candidatos;
- Pesquisa: algumas alegações de caráter duvidoso tiveram que ser pesquisadas e só foram descartadas

ou confirmadas como *fake news* quando encontrada uma notícia verificada por agência fidedigna de *fact checking* no *dataset* do grupo de pesquisa Interfaces.

Descrição da base de notícias de checagem de *fake news*

Esta base, igualmente pertencente ao grupo de pesquisas Interfaces, contém 1872 notícias verificadas: do Projeto Comprova, que reúne diversos veículos de comunicação; da AFP Checamos, um departamento da agência francesa Agence France-Presse; do site E-farsas; do Fato ou Fake, serviço de checagem do Grupo Globo; da Lupa, a agência de checagem e educação midiática; da organização Boatos.org, que reúne diversos jornalistas; do site independente Aos Fatos; do UOL Confere e o Fato ou Boato, uma plataforma concebida em 2020 como parte do Programa de Enfrentamento à Desinformação do Tribunal Superior Eleitoral (TSE).

Para este trabalho, foram selecionadas 371 matérias únicas checando a veracidade de notícias de caráter duvidoso que se espalharam por redes sociais. Essas notícias foram de enorme importância para este trabalho, fornecendo valiosos insights para a criação de heurísticas usadas no modelo de rotulagem. A base contém: o título da checagem, data de publicação, veracidade da notícia, o candidato favorecido pela notícia falsa (se houver), agência de publicação, e uma url para a publicação.

A partir disto, foi feita uma raspagem das notícias, ou scraping, com as bibliotecas de Python3: Requests e BeautifulSoup, em seguida foram removidas notícias duplicadas e foi feito um pré-processamento textual: remover tags HTML, remover pontuação, remover stopwords e fazer tokenização do texto (NLTK). Após o processamento dos textos, foram criados os vetores de embeddings com Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013) usando a biblioteca de Python3 Gensim .

Embeddings são representações numéricas densas de palavras em um espaço vetorial contínuo de alta dimensão; durante o treinamento do modelo de embedding, o modelo analisa o contexto em que cada palavra aparece; palavras que são frequentemente usadas em contextos semelhantes tendem a ter representações de embeddings mais próximas umas das outras no espaço vetorial, e dessa forma, o espaço vetorial gerado consegue representar numericamente informações semânticas e sintáticas dessas palavras. Os vetores foram criados com 100 dimensões. Verificando assimetria dataframe.skew() e curtose dataframe.kurt(), as distribuições dos vetores de embeddings aparentam ser similar à normal padrão dado que ambas as estatísticas foram próximas a 0 em todos os vetores – segundo a definição do Pandas.

Decidiu-se usar um algoritmo de clustering para agrupar as notícias e identificar automaticamente tópicos que poderiam ser utilizados na criação das funções de rotulagem. Seguindo a tabela do Scikit-Learn na Figura 3, utilizou-se o Kernel PCA para reduzir a dimensionalidade dos vetores de embeddings das notícias para três dimensões, a fim de visualizar os dados em um gráfico, Figura 2. Pelo gráfico, foi possível descartar o DBSCAN e o OPTICS, pois os dados se assemelham mais à terceira linha da tabela da Figura 3. O algoritmo BIRCH é mais adequado para *dataset*s grandes, enquanto o MeanShift não lida bem com dados de alta dimensão, sendo assim, foram descartados. O AgglomerativeClustering, o WardClustering e o SpectralClustering realizaram, respectivamente, 3, 2 e 3 agrupamentos, como é característico desses algoritmos fazerem poucos clusters.

No entanto, isso não estava alinhado com o objetivo de buscar vários tópicos distintos para a criação das funções de rotulagem. Essa necessidade ficou clara quando comparados os resultados desses algoritmos com o K-means e o AffinityPropagation, que formaram bem mais que 3 grupos e se tornaram as duas melhores opções. A formação de 3 ou 2 grupos resultava em clusters genéricos, sem representar de forma adequada os diferentes tópicos.

Em resumo, o K-means funciona da seguinte forma: escolhe-se um valor para K; o algoritmo seleciona K centroides e atribui os dados ao centroide mais próximo; os centroides são atualizados iterativamente até a convergência, movendo-se para a média dos pontos atribuídos a cada grupo.

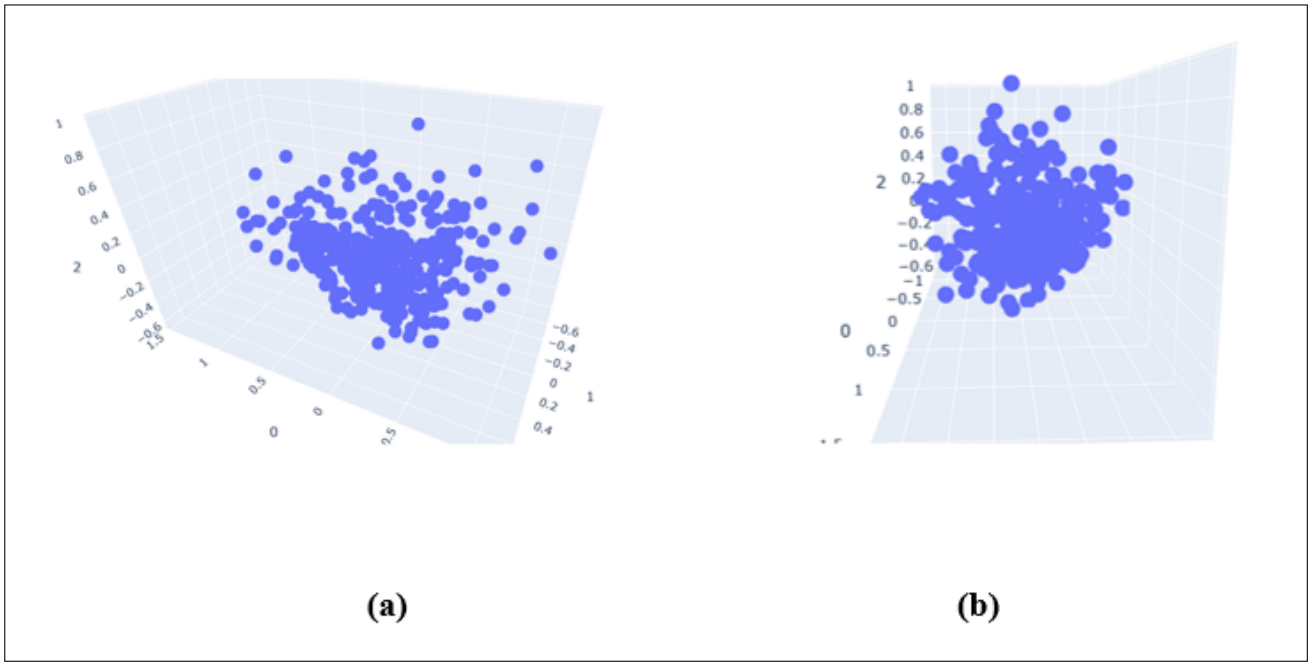

Figura 2. (a) Visão frontal dos dados reduzidos com KPCA. (b) Visão lateral dos dados reduzidos com KPCA

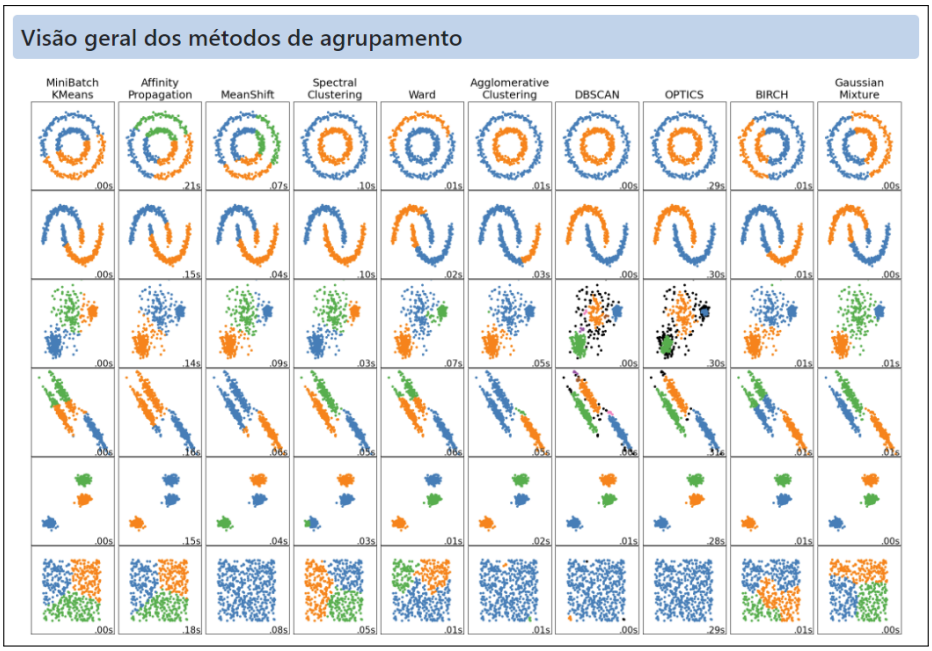

Figura 3. Comparação dos algoritmos de agrupamento do scikit-lean

Já o algoritmo Affinity Propagation funciona da seguinte forma: é um método de agrupamento que identifica automaticamente os pontos mais representativos em um conjunto de dados para formar clusters. Ele utiliza medidas de afinidade entre os pontos para propagar a informação de similaridade e selecionar o exemplar mais representativo como centro do cluster.

Usando o método do cotovelo, aliado ao Silhouette score para cada k testado (de 2 a 30), o agrupamento ótimo foi obtido com $k = 11$ grupos, localizado no "cotovelo" do gráfico da Figura 4 e com o maior silhouette score na Figura 5 entre os valores de K próximos à dobra. Os agrupamentos do K-means apresentaram resultados satisfatórios, embora tenham ficado um pouco desbalanceados; isso, no entanto, não representou um problema para o objetivo final com os agrupamentos.

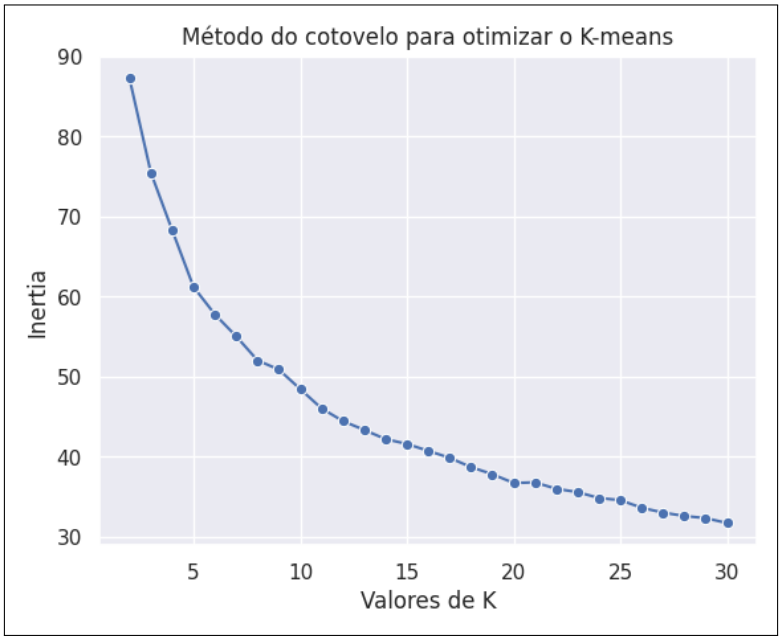

Figura 4. Método do cotovelo para o K-means

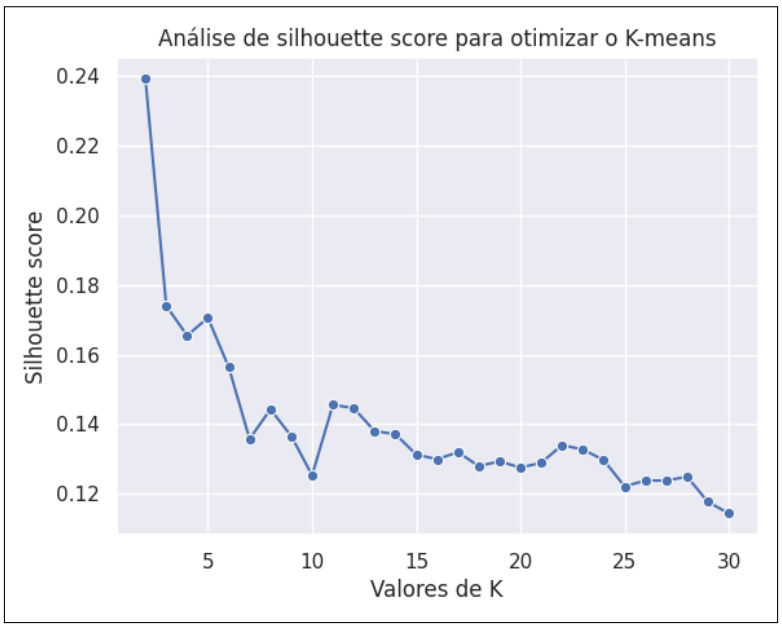

Figura 5. Silhouette score do K-means

O K-means criou um cluster de tamanho máximo com 60 notícias e um cluster de tamanho mínimo com 18 notícias. Por outro lado, os agrupamentos do Affinity Propagation mostraram-se mais promissores. Empiricamente, eles apresentaram uma distribuição mais uniforme e distinta. O Affinity Propagation gerou 29 clusters, dos quais 3 clusters possuíam o tamanho máximo de 31 notícias e 1 cluster possuía o tamanho mínimo de 1 notícia.

O algoritmo Affinity Propagation possui as seguintes características:

1) Não requer a especificação prévia do número de clusters desejados.
2) Pode lidar bem com dados que possuem estruturas complexas ou desconhecidas.
3) Cria muitos clusters e é capaz de lidar com clusters de tamanhos variáveis.
4) É computacionalmente eficiente em conjuntos de dados relativamente pequenos a moderados, pois exige recursos computacionais significativos.

Tais características atenderam muito bem ao que era esperado, pois, avaliando os gráficos gerados usando Kernel PCA, na Figura 2, não havia clusters bem definidos. Além disso, havia poucas notícias, o que tornou a complexidade do algoritmo irrelevante. Foi possível até mesmo utilizar o Grid Search para escolher o melhor

valor de damping. A característica do algoritmo de identificar vários clusters foi exatamente o que o tornou superior aos outros métodos avaliados.

Com os agrupamentos definidos, agora era necessário encontrar uma maneira de extrair os tópicos. Inicialmente, procurou-se obter os 10 termos mais representativos usando o Word2Vec e a função word2vec.wv.most_similar, utilizando o vetor que representava o centro de cada cluster. No entanto, os termos resultantes não foram satisfatórios em termos de fornecer ideias abrangentes.

Uma alternativa simples, porém com resultados muito melhores, foi criar nuvens de palavras (Word Clouds) a partir de todas as notícias de um determinado agrupamento. Os resultados serão apresentados no próximo item deste artigo.

Sobre a rotulagem dos *tweets*

O conjunto de *tweets* utilizado neste estudo não possui rótulos, o que apresenta um desafio para a aplicação de abordagens supervisionadas. No entanto, é importante ressaltar que a base de dados utilizada é extensa, contando com mais de 2 milhões de registros. Para contornar essa limitação, optamos por uma abordagem de supervisão fraca, a fim de rotular esses dados antes de empregarmos um classificador supervisionado.

A aprendizagem com supervisão fraca, também conhecida como weak supervision em inglês, é comumente utilizada quando é difícil ou custoso obter rótulos precisos para todo o conjunto de dados. Essa abordagem consiste em utilizar heurísticas, regras simples e potencialmente ruidosas para realizar a rotulagem dos dados.

Nesse contexto, a biblioteca Snorkel , em Python3, tem como principal objetivo auxiliar desenvolvedores a criar um conjunto de treinamento que pode ser usado para treinar modelos de aprendizado. Embora seja mencionado que o Snorkel pode ser utilizado para criar um conjunto de treinamento sem a necessidade de qualquer rotulagem manual, no tutorial das funções de rotulagem é recomendado ter um conjunto pequeno de dados rotulados manualmente para fins de avaliação final do modelo. Ressalta-se que esse conjunto não deve ser utilizado para análise de erros, mas sim para avaliação padrão de métricas, como acurácia e precisão, por exemplo.

Tendo como base os termos relevantes selecionados das notícias de checagem de *fake news*, foram criadas dezessete funções de rotulagem heurísticas para identificar as principais notícias falsas que marcaram o segundo turno das eleições. Os resultados temáticos serão apresentados na próxima seção deste artigo.Outras heurísticas para rotular notícias falsas envolveram:

- Casos em que o autor do *tweet* é conhecido por publicar desinformação com frequência, nos quais foram separados os seguintes usuários: @gazetadopovo, @revistaoeste e @PastorMalafaia;
- Casos que aparecem palavras imperativas para o leitor compartilhar a informação, como por exemplo "espalhe", "compartilhe".
- Enquanto as heurísticas para encontrar *tweets* sem notícias falsas incluem:
- *tweets* declarando voto;
- Ausência da menção do candidato por nome ou apelido amigável;
- Ausência da menção do candidato por apelido pejorativo ou depreciativo

Apesar da busca na API ter sido especificamente por *tweets* que mencionem os candidatos, foi detectado durante a rotulagem dos *tweets* a existência de vários destes que não os mencionam de forma alguma, no caso de algumas opiniões que falam somente sobre o contexto das eleições. Possivelmente, trata-se dos comentários das publicações dos candidatos.

Tais heurísticas onde não se encontra o candidato no texto foram observadas com expressões regulares. Foram testadas duas implementações de reconhecimento de entidades, uma do Snorkel, e outra do Spacy 23, mas ambas não geraram um resultado satisfatório, possivelmente porque os textos são no idioma português.

- Se o autor do *tweet* é um usuário dito confiável, seriam estes, portais de notícia ou jornalistas de renome;
- Portais de notícia: @g1, @NexoJornal, @agencialupa, @exame, @aosfatos, @UOLNoticias, @Globo *news*, @jornalnacional, @bbcbrasil, @portalR7, @CNNBrasil;
- Jornalistas: @miriamleitao, @evaristocosta, @cristilobo, @monicabergamo, @PVC, @PatriciaKogut, @LilianPacce, @gcamarotti, @AndreiaSadi, @JoycePascowitc

As funções de rotulagem são declaradas usando a anotação @labeling_function(), devendo receber como entrada um texto $\bar{x}$, onde é feita a verificação conforme a heurística, a qual retorna um valor inteiro: 1 para *fake news*, 0 não há *fake news*, -1 em casos em que o modelo não sabe o que rotular. Por exemplo:

```
FAKE_ \textit{news} = 1
NOT_FAKE = 0
```

```
ABSTAIN = -1

@labeling_function()
def check_canibalismo(x):
    if re.search(r"(canibal|carne humana)",
                 x.text.lower(), flags=re.I):
        return FAKE_ \textit{news}
    else:
        return ABSTAIN
```

Usando um objeto do tipo LFApplier do Snorkel, aplicamos as funções; para cada entrada de dados há uma saída de cada função, que retorna em uma matriz relacionando todas as funções com todos os dados. Essa matriz deve ser dada como entrada para um modelo de rotulagem. O modelo principal do Snorkel é o LabelModel (Ratner et al., 2019), que foi utilizado neste trabalho.

Resumidamente, o LabelModel usa um framework probabilístico para combinar os rótulos gerados pelas funções heurísticas definidas, e assim, gerar rótulos probabilísticos para os dados não rotulados.

Na Tabela 3 é apresentada a distribuição dos rótulos atribuídos pelo snorkel aos dados do modelo classificador. Na Tabela 4 tem-se a distribuição dos rótulos atribuídos pelo snorkel aos dados durante o treinamento do modelo rotulador e também a distribuição dos rótulos que foram atribuídos manualmente para testar o modelo rotulador.

	Treinamento		Teste	
Rótulo	**Freq.**	**Porc.**	**Freq.**	**Porc.**
Fake	64488	5,134%	16414	5,230%
Não Fake	1256208	93,440%	313822	93,371%
Abstenção	23706	1,763%	5865	1,745%
Total	1344402	100%	336101	100%

Tabela 3. Distribuição dos dados de treinamento e do teste classificador

	Treinamento		Rótulo Manual	
Rótulo	**Freq.**	**Porc.**	**Freq.**	**Porc.**
Fake	20394	5,212%	69	7,062%
Não Fake	391273	93,366%	977	93,048%
Abstenção	7409	1,768%	4	0,381%
Total	419076	100%	1050	100%

Tabela 4. Distribuição dos dados de treinamento e de teste (rotulamento manual) do modelo rotulador

Quatro *tweets* eram ambíguos em relação à sua veracidade, portanto foram descartados antes de ser feita a avaliação do modelo de rotulagem. Pela maneira como foram definidas as funções de rotulagem, se o modelo não identificar se há *fake news* ou não, ele poderia abster-se de dar um rótulo. Percebe-se que a proporção de abstenções do modelo nos dados não vistos (o conjunto do classificador) é a mesma que a proporção do conjunto de teste, por volta de 5%.

Percebe-se também que os dados sofrem de sub-amostragem severa de notícias falsas. Por estarem presentes em quantidade muito pequena os modelos de aprendizagem podem interpretar essa categoria (notícias falsas) como sendo anomalia. Dos dados que foram rotulados pelo modelo do Snorkel, todos tem entre 1% e 2% de notícias falsas em relação ao total. Mesmo no conjunto rotulado manualmente, na Tabela 4, a proporção de notícias falsas é pequena.

Sobre o modelo de classificação

O modelo de classificação utilizado teve sua arquitetura inspirada na arquitetura proposta por Saleh et al. (2021). Foi projetada uma rede neural convolucional de treze camadas para processamento dos *tweets*, implementada com a API de Deep Learning para Python3 Keras (Chollet et al., 2015), baseada no TensorFlow. Esta deve receber como entrada matrizes de números com tamanho fixo, razão pela qual os *tweets* passaram por um pré-processamento antes de treinar o classificador.

Utilizando o Tokenizer do Keras, cada *tweet* foi transformado numa matriz de números. Para fixar o tamanho das matrizes, encontra-se a maior matriz, e faz-se o padding com zeros nas outras matrizes para que todas tenham as mesmas dimensões $m \times m$.

O modelo tem a seguinte estrutura: uma camada de embeddings, uma camada de Reshape, 3 sequências seguidas de: camada convolucional 2D, seguida por camada de MaxPooling2D, seguida por camada de SpatialDropout 2D, e depois das três sequências, uma camada Flatten e a Camada Densa de saída.

1) Camada de Embeddings: Como parâmetro de pesos (weights) foi usada uma matriz de embeddings criada com glove.6B.200d (Pennington et al., 2014), que cria vetores de 200 dimensões para cada palavra. A dimensão de entrada é o tamanho do vocabulário do conjunto de dados, a dimensão de saída é 200, que é a dimensionalidade do Glove, e o tamanho da entrada é a quantidade de palavras do maior *tweet*.

2) Camada de Reshape: Redimensiona a saída da camada de embeddings, é necessário para que a saída seja aceita como entrada pela camada convolucional.

3) Camada Convolucional: Camada Conv2D do Keras, recebe como entrada uma matriz de "palavras", camada que é o componente fundamental desta rede neural porque é usada para extrair características relevantes das entradas. Todas as camadas usadas nessa rede foram definidas com 128 filtros, e tamanho de kernel 2, como descrito no artigo de Saleh et al. (2021).

A função de ativação utilizada foi a ReLU (Rectified Linear Unit), que retorna somente valores positivos ou 0 se a entrada for negativa: $f(x) = \max(0, x)$. Essa função de ativação é amplamente utilizada em redes neurais porque ajuda a resolver o problema de desvanecimento do gradiente, que pode ocorrer em funções de ativação saturadas, como a função sigmoide, evitando, assim, a desaceleração do aprendizado em camadas mais profundas da rede.

4) Camada de Max Pooling: A camada de pooling é uma janela que percorre o mapa de características; por ser max pooling, ela pega o maior valor dentro da janela. Essa camada é usada para reduzir o mapa de características. Saleh et al. (2021) usaram Max Pooling porque escolher o maior valor resulta na captura das características mais significativas do mapa.

5) Camada de Dropout: Usar uma camada de dropout é uma conhecida técnica de regularização, servindo para reduzir a complexidade do modelo e também diminuir o overfitting, o super ajuste dos pesos da rede neural, que impede a capacidade de generalização de um modelo. Uma camada de dropout tradicional desativa neurônios de forma independente a cada atualização, enquanto o spatial dropout desativa regiões de neurônios, ambos agem de forma aleatória. O uso do spatial dropout pode ser especialmente eficaz em redes neurais convolucionais, onde a estrutura espacial dos dados é importante para a extração de características.

6) Camada Flatten: Serve para transformar a matriz em um vetor de uma dimensão, para dar como entrada à camada de saída.

7) Camada Densa: Camada de saída com ativação sigmóide, recebe como entrada o vetor unidimensional da camada anterior Flatten para produzir a resposta final com seu único neurônio, se o *tweet* contém *fake news* (1) ou não (0).

1) Camada de Embeddings: Como parâmetro de pesos (weights) foi usada uma matriz de embeddings criada com glove.6B.200d (Pennington, Socher, & Manning, 2014), que cria vetores de 200 dimensões para cada palavra. A dimensão de entrada é o tamanho do vocabulário do conjunto de dados, a dimensão de saída é 200, que é a dimensionalidade do Glove, e o tamanho da entrada é a quantidade de palavras do maior *tweet*.

2) Camada de Reshape: Redimensiona a saída da camada de embeddings, é necessário para que a saída seja aceita como entrada pela camada convolucional.

3) Camada Convolucional: Camada Conv2D do Keras, recebe como entrada uma matriz de "palavras", camada que é o componente fundamental desta rede neural porque é usada para extrair características relevantes das entradas. Todas as camadas usadas nessa rede foram definidas com 128 filtros, e tamanho de kernel 2, como descrito no artigo de Saleh et al. (2021). A função de ativação utilizada foi a ReLU (Rectified Linear Unit), que retorna somente valores positivos ou 0 se a entrada for negativa: $f(x) = \max(0, x)$. Essa função de ativação é amplamente utilizada em redes neurais porque ajuda a resolver o problema de desvanecimento do gradiente, que pode ocorrer em funções de ativação saturadas, como a função sigmoide, evitando, assim, a desaceleração do aprendizado em camadas mais profundas da rede.

4) Camada de Max Pooling: A camada de pooling é uma janela que percorre o mapa de características; por ser max pooling, ela pega o maior valor dentro da janela. Essa camada é usada para reduzir o mapa de características. Saleh et al. (2021) usaram Max Pooling porque escolher o maior valor resulta na captura das características mais significativas do mapa.

5) Camada de Dropout: Usar uma camada de dropout é uma conhecida técnica de regularização, servindo para reduzir a complexidade do modelo e também diminuir o overfitting, o super ajuste dos pesos da rede neural, que impede a capacidade de generalização de um modelo. Uma camada de dropout tradicional

desativa neurônios de forma independente a cada atualização, enquanto o spatial dropout desativa regiões de neurônios, ambos agem de forma aleatória. O uso do spatial dropout pode ser especialmente eficaz em redes neurais convolucionais, onde a estrutura espacial dos dados é importante para a extração de características.

6) Camada Flatten: Serve para transformar a matriz em um vetor de uma dimensão, para dar como entrada à camada de saída.

7) Camada Densa: Camada de saída com ativação sigmóide, recebe como entrada o vetor unidimensional da camada anterior Flatten para produzir a resposta final com seu único neurônio, se o *tweet* contém *fake news* (1) ou não (0).

Configurações utilizadas

Com exceção da coleta de notícias verificadas do Uol Confere e Projeto Comprova, todos os experimentos foram conduzidos utilizando notebooks no ambiente do Google Colab , utilizando a linguagem de programação Python3. A versão gratuita do Google Colab possui uma limitação em relação à duração contínua da execução na mesma célula, uma vez que isso é considerado inatividade e resulta no encerramento automático do ambiente. Como resultado, não foi possível realizar o treinamento da rede neural utilizando a GPU Padrão, devido à duração prolongada do processo que excederia o limite de tempo considerado como inatividade. O treinamento do modelo de rotulagem teve uma duração aproximada de 3 horas, utilizando o ambiente com a GPU T4 e memória RAM de alta capacidade do Colab, enquanto o treinamento da rede neural de classificação levou cerca de 2 horas, utilizando o ambiente com a GPU A100 e memória RAM também de alta capacidade.

RESULTADOS

Tópicos de *fake news* na campanha eleitoral

Com os agrupamentos definidos, foi possível extrair os principais tópicos das *fake news* que circularam durante a campanha presidencial em 2022. Assim, foram criadas nuvens de palavras (Word Clouds) a partir de todas as notícias de um determinado agrupamento. Um exemplo pode ser observado na Figura 6.

Figura 6. Exemplo de Word Cloud feita com um dos clusters

Foram criadas nuvens de palavras para todos os agrupamentos que continham pelo menos dez notícias, o que totalizou em dezessete gráficos, dentre vinte e nove, para análise visual.

Word Clouds, também conhecidas como nuvens de tags ou nuvens de palavras, são representações visuais em que o tamanho de cada palavra é proporcional à sua frequência de ocorrência em um texto específico ou em um conjunto de textos. Ou seja, palavras maiores representam maior frequência. Apesar da maior frequência, as palavras mais importantes, na realidade, são as palavras menores presentes nas nuvens. Isso ocorre porque as palavras mais frequentes, em quase todos os textos, tendem a ser as mesmas, como "vídeo", "presidente", "comprova", "lula", "bolsonaro"e "jair", conforme demonstrado na Figura 6. Por outro lado, as palavras menores

forneceram insights mais relevantes para a criação de funções heurísticas de rotulagem, como "vacina", "covid19", "fraude", "urna eletrônica", "militares", entre outras.

É importante observar também que os quatro maiores agrupamentos apresentaram palavras mais genéricas e clusters pequenos demais, que foram os últimos a serem analisados e já não traziam termos novos, e por isso, foram desconsiderados. Abaixo estão listadas as palavras extraídas dos word clouds, por cluster:

- CLUSTER 1: china, aborto, vídeo, motociata, controle natalidade, facada, pílula, coelho
- CLUSTER 2: laudo, empresa, Petrobras, terceirizada, processo eleitoral, serviço, urna, fraude eleitoral, maçonaria, carteiro/correio, direitos humanos
- CLUSTER 3: fábio assunção, ivermectina, militar, forças armadas, covid19, bndes, financiamento, mst
- CLUSTER 4: jogador, richarlison bolsonarista, vacina, tiktok e kwai, lava jato, collor
- CLUSTER 5: urnas, forças armadas, tiktok e kwai, indígena (terras indígenas), manifestações, luciferianismo, satanismo, igreja, professora
- CLUSTER 6: urna, indígena, ciro gomes, pesquisador(a/es), tiktok, polícia, londres
- CLUSTER 7: campanha, seções eleitorais, comício, boletins, urna, "lula xinga eleitores", ônibus, tiktok kwai facebook instagram youtube, alexandre moraes e xandão
- CLUSTER 8: bndes, jovem pan, gusttavo lima, ministério e ministro, heineken, jbs, moradia social, poliomielite, marine club, carne, frigorífico, seção eleitoral, urna, mato grosso, empresa, brasnorte, comício, vacinação
- CLUSTER 9: leite, carne, urna, boletins, conta, carros usados, aborto, china, pílula, portugal, pandemia, recontagem, ato, orçamento secreto, controle natalidade, imposto renda, pobre
- CLUSTER 10: igreja, estádio, hospitais, pistola, torcida, vasco, enfermeiros, fraude, jovem pan, coronel tadeu, torcedores, youtube tiktok kwai, live
- CLUSTER 11: férias remuneradas, medida provisória mp, whatsapp tiktok kwai facebook, tv cultura, câmara, alexandre moraes (xandão), militares, benefício, forças armadas, jornalista, vera magalhães, jovem pan
- CLUSTER 12: lulinha, filho, fazenda, ludmilla, argentina, google, caminhões, advogados, herança, condenado, tiktok kwai facebook, seção eleitoral, militar, manifestação
- CLUSTER 13: bahia, guerrilha, nikolas ferreira, militar, fidel castro, colômbia, rio de janeiro, salário mínimo, urna, complexo alemão, vila joão, fraude, medina, comando vermelho cv, sigla cpx cupinxa cupincha complexo, traficante
- CLUSTER 14: militares, eleitores, aposentadoria, benefício, forças armadas, roberto jefferson, andré, constituição, gleisi hoffmann, banheiro unissex, criança, xuxa, gazeta, segurança pública, previdência, facebook tiktok kwai, salário mínimo, grávida, bilionário, karina, gritos, mito, teto de gastos, elon musk
- CLUSTER 15: g1, imigrante, pará, calçada, namorar, guarapari, instituto, emprego, estrangeiro, igreja, neymar, religioso, google, drogas, MEI, pandemia, debate, teto de gastos
- CLUSTER 16: vacinação, mislav kolakusic, lacração, carga, terceirizado, sindicato, covid19, apoio, importações, parlamento europeu, pobre, morte, braga netto, janine small, seções eleitorais, criança, zona eleitoral, facebook tiktok kwai, 53 zona, vacina, itapeva, fraude, google
- CLUSTER 17: imunizante, janja, IPEC, ministro da educação, jean wyllys, canadá, mato grosso, pfizer, ianomâmi, miami, boca de urna, pesquisa boca, primeira universidade

Exemplificamos uma das centenas de *fake news* verificadas de um dos 17 clusters acima descritos, com a tag "fábio assunção", que a princípio não parece remeter a uma *fake news*, mas que na realidade se refere à seguinte notícia falsa já verificada: "É falso que Fábio Assunção gravou vídeo imaginando reeleição de Bolsonaro".

Heurísticas para identificação de *fake news*

Tendo como base os termos relevantes selecionados das notícias de checagem de *fake news*, foi possível criar dezessete funções de rotulagem heurísticas para identificar as principais notícias falsas que marcaram o segundo turno das eleições, a saber:

- *tweets* que relacionassem algum dos candidatos à maçonaria, satanismo ou canibalismo;
- Alguma apologia ao "tratamento precoce" contra o Covid19 e o "kit covid" de remédios sem comprovação científica, envolvendo palavras-chave como melhora, cura e afins;

- *tweets* afirmando fraude nas eleições ou que insinuavam que as urnas eletrônicas estevam beneficiando algum candidato com contagem de votos enviesada;
- *tweets* mencionando aborto e técnicas abortivas, uma vez que nenhum dos dois candidatos mencionou apoiar o procedimento como método contraceptivo durante o período de campanha eleitoral;
- Tentativas de relacionar Lula, seu filho, ou o Partido dos Trabalhadores a: traficantes, propostas de "kit gay" ou ao ímpeto de "forçar" a ideologia de gênero nas escolas, aumento do desmatamento na Amazônia, fechar igrejas, entre outras notícias falsas conhecidas;
- Censura do TSE por motivos de disputa política, como se um candidato tivesse influência sobre o TSE para ganhar vantagem em campanha política;
- Que o número de votar no candidato Bolsonaro seria 17, sendo que o correto no pleito de 2022 era o número 22;
- Que algum jogador ou time de futebol fez campanha ou homenagem a algum candidato;
- Financiamento ilegal de construções pelo BNDES.

Observações e Métricas da Classificação

Nas Figuras 7 e 8 estão as métricas avaliadas durante o treinamento do modelo. A acurácia do conjunto de treinamento e de validação ficaram bem próximas, e a acurácia de validação oscila um pouco, como visto na Figura 8.

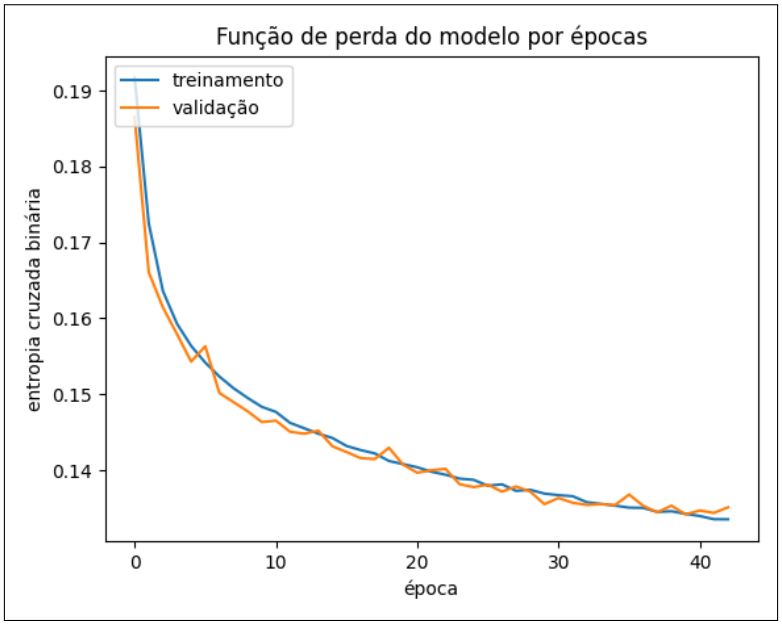

Figura 7. Função de perda (entropia cruzada binária) do modelo por épocas

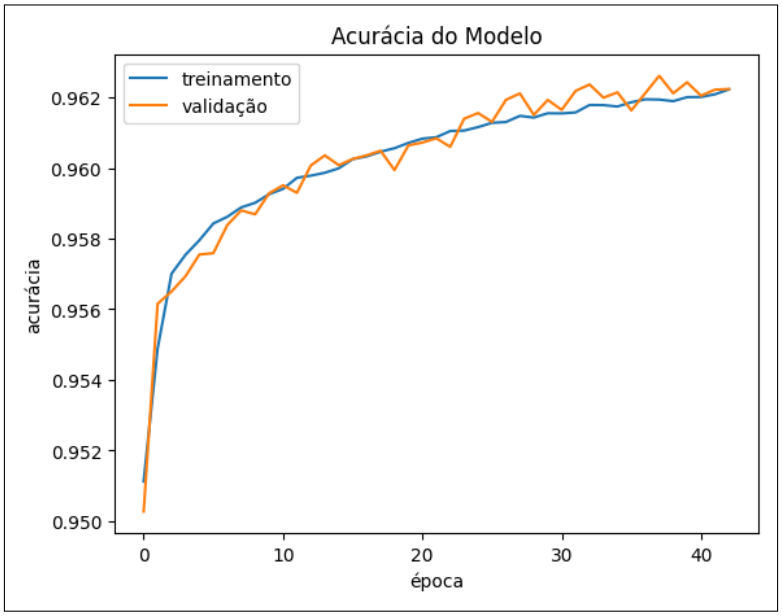

Figura 8. Acurácia do modelo por épocas

Pelo gráfico da função de perda na Figura 7, percebe-se que o mesmo ainda está em tendência de queda e apesar da perda no conjunto de validação aparentar uma subida, talvez voltasse a cair se a paciência do callback de interrupção de treinamento estivesse acima de 3. Para não ultrapassar o limite de unidades computacionais disponíveis, a melhor opção foi utilizar um valor baixo para a paciência do callback de interrupção. A acurácia do modelo no conjunto de teste foi de 96%.

Na Tabela 6, pode-se observar os valores das outras métricas de classificação da rede neural em relação aos dados de teste. Como mostrado na seção de rotulagem dos *tweets*, os dados sofrem de grave sub-amostragem, visto que cerca de 5% dos dados (após eliminação das abstenções) têm rótulo positivo (i.e., *fake news*). O limiar de classificação escolhido foi 0,3, e foi levado em consideração que a rede neural aprendeu a classificar muito bem os rótulos negativos, mas não tão bem os rótulos positivos, com limiar = 0,5. A revocação de *tweets* com *fake news* ficou por volta de 0,25, e o uso do limiar = 0,2 levava a uma diminuição significativa da precisão. Nesse sentido, esse foi o limiar que obteve o melhor equilíbrio entre precisão e revocação, observando principalmente as médias macro e ponderada do F1-score.

Ao observar os valores das métricas separados por classe na Tabela 6, é possível argumentar que o classificador é capaz de distinguir com uma precisão razoavelmente boa as duas classes, visto que obteve 97% para a classe "Não" e 78% para a classe "Sim". O principal problema do classificador esteve em recuperar uma maior quantidade de amostras da classe "Sim". Neste aspecto, evidencia-se um caso típico de desbalanceamento entre classes, no entanto, onde a aplicação de métodos populares como aumento de dados não é possível. Seria possível uma investigação da utilização de heurísticas de rotulamento com aprendizagem fraca que fossem mais representativas para permitir a recuperação de uma maior quantidade de amostras da classe "Sim". Essas heurísticas poderiam ser obtidas por meio da consulta a profissionais da área ou da exploração de dados rotulados por fact checking. Cabe ressaltar, que a baixa recuperação de amostras da classe "Sim" pode decorrer também do fato de que as *fake news* contém dubiedade.

Hiperparâmetro	**Valor**
Otimizador	ADAM
Função de perda	entropia cruzada binária
Tamanho do batch	512
Fração de validação	20%
Épocas	43 épocas (terminou com *early stopping*)
Callbacks	EarlyStopping (paciência = 3) ModelCheckpoint

Tabela 5. Hiperparâmetros do Classificador

A curva ROC – do inglês Receiver Operating Characteristic – é construída traçando as taxas de verdadeiros positivos e de falsos positivos. A área sob a curva ROC (AUC-ROC) é uma medida comumente usada para resumir o desempenho do classificador. Quanto maior for o valor da AUC-ROC, maior é a capacidade de distinguir entre as classes positiva e negativa. A área sob a curva ROC foi de 0.848, o que é considerado um bom

fake news?	Precisão	Revocação	F1-score	Qt. de Amostras
Não	0.97	0.99	0.98	313822
Sim	0.78	0.34	0.47	16414

Tabela 6. Precisão, Revocação, F1-score nos dados de teste. Limiar de classificação = 0,3

Médias	Precisão	Revocação	*F1-score*
macro	0.87	0.67	0.73
ponderada	0.96	0.96	0.96

Tabela 7. Médias macro e ponderada: Precisão, Revocação e F1-score nos dados de teste. Limiar de classificação = 0,3

valor bom, que mostra potencial. A Figura 9 mostra a área sob a curva ROC da rede neural de classificação, que ficou acima da linha reta (AUC = 0.5) indicando que o modelo é melhor que um classificador aleatório.

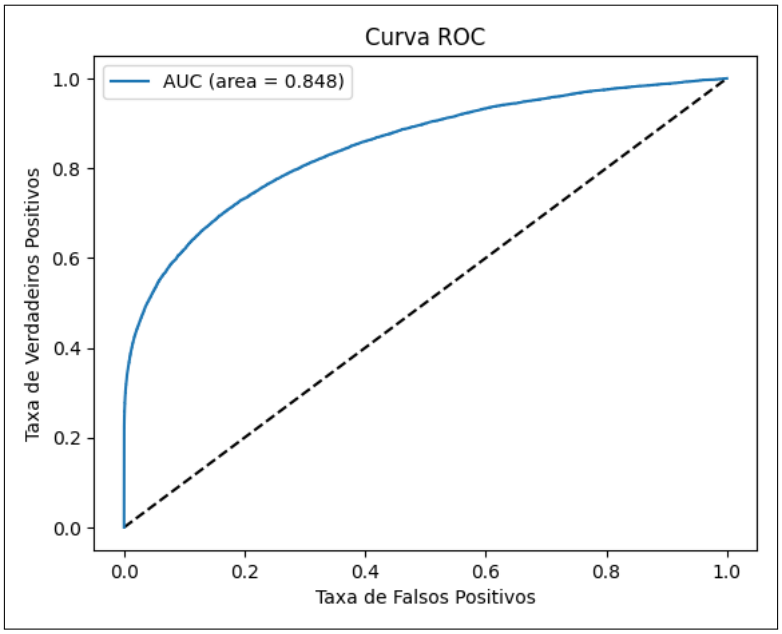

Figura 9. Área sob a curva ROC de 0.848

Discussão sobre as classificações

Analisando uma pequena amostra aleatória dos dados (500 amostras), comparando o rótulo atribuído pelo modelo rotulador com a classificação predita e lendo o texto para avaliá-los manualmente, é possível verificar que grande parte das postagens não contém propriamente notícias falsas, e que muitas vezes, contém apenas uma afirmação exagerada de uma opinião.

Considerando a pequena amostra, a rede neural parece ter aprendido bem a identificar padrões subjacentes em uma notícia verdadeira. Alguns casos de falsos negativos aparentam ter sido mal rotulados pelo modelo de supervisão fraca, apesar de terem sido criadas poucas heurísticas para a rotulagem de postagens sem *fake news*.

Nessa amostra, não foi encontrado nenhum falso positivo em que a postagem aparentava ter notícia falsa, onde o classificador rotulou como falso e o rotulador não. Um problema observado encontrou-se nos dados em que o rotulador e o classificador concordaram que a postagem não continha *fake news*, quando na realidade continha. Foram encontrados alguns exemplos disso na amostra. Da mesma forma, foram encontrados alguns exemplos do rotulador e classificador, concordando que uma postagem continha *fake news*, quando na realidade não as continha, o que acaba por mascarar a performance do modelo quando se observa métricas como precisão e revocação. Além disso, é provável que a proporção de falsos "verdadeiros positivos/negativos"seja similar no restante dos dados que não foram observados. Abaixo estão listados alguns exemplos:

Exemplo de verdadeiro positivo (rótulo: 1, predição: 1), que devia ser "não *fake news*", mas ambos erraram:

- "@taoquei1 MEI não é emprego, é um CNPJ entendeu? MEI não tem salário, não tem férias, 13º, repouso remunerado e nem FGTS. Ser MEI não é garantia de TRABALHO. Foi isso que Lula quis dizer e entendi, pois sou MEI. Te atualizei?"

Exemplo de Falso Positivo (rótulo: 0, predição: 1), que deveria ter sido considerado não *fake news*, onde o rotulador acertou e o classificador errou.

- "PARA DEPUTADO FEDERAL É NENÉM DO TAPETE! PARA PRESIDENTE É LULA 13 Por São João de Meriti e pelo Brasil! https://t.co/0W4iKqVFaR"

Exemplo de Falso Negativo (rótulo: 1, predição: 0), que devia ser não *fake news*, onde o classificador acertou e o rotulador errou.

- "Sou PcD e estou com Lula 13 https://t.co/dWdc4CvI9e"

Exemplo de Verdadeiro Negativo (rótulo: 0, predição: 0), mas que é *fake news*, onde ambos erraram.

- "Domingo Espetacular: "LULA é um dos MANDANTES do ASSASSINATO do prefeito Celso Daniel."Aponta investigação. Alexandre de Moraes já censurou a Liberdade da Record? @DomEspetacular @MPF_PGR @PoliciaFedeeral @STF_oficial @LulaOficial https://t.co/Fx1BBvRJAx"

CONCLUSÕES E TRABALHOS FUTUROS

Considerando que as *fake news* apresentam características ambíguas e são escritas com o intuito de enganar os leitores, é um desafio detectá-las com base no conteúdo da notícia. Este trabalho buscou contribuir com a detecção automática de notícias falsas por meio da construção de uma rede neural profunda que pudesse identificar a presença de *fake news* em *tweets* de uma base de dados coletada durante as eleições.

Com esta pesquisa, foi possível concluir que são promissores os resultados de modelos de redes neurais artificiais para agilizar o tão necessário trabalho de verificação de veracidade de notícias, especialmente em pleitos eleitorais, que podem ter consequências severas para a democracia. Contudo, vale ressaltar que ainda é um desafio trabalhar com grandes volumes de dados não rotulados, como aqueles utilizados nesta pesquisa. Ao adaptar e expandir técnicas de detecção de *fake news* para o ambiente brasileiro do X/Twitter e para o idioma português, esta pesquisa não apenas avança o estado da arte na área de detecção automática de desinformação, como também oferece uma base sólida para o desenvolvimento de ferramentas práticas de apoio à integridade eleitoral e ao combate à desinformação em democracias vulneráveis.

Foi possível identificar que os principais tópicos que retratam a agenda das campanhas estavam relacionados a valores morais e religiosos (aborto, perseguição a igrejas, ideologia de gênero), pautas econômicas (Pix, privatização, salário mínimo, políticas redistributivas), questionamento à idoneidade de instituições, fraude nas urnas eletrônicas, endosso de figuras públicas, associação de adversários e partidos ao crime, negacionismo científico.

Os resultados obtidos nesta pesquisa possuem como principal contribuição a metodologia que pode ser aplicada a outros contextos semelhantes. Pretende-se expandir o uso da metodologia proposta neste trabalho para futuras pesquisas, utilizando o mesmo conjunto de dados. Para lidar com a baixa revocação para amostras da classe "Sim", poderão ser investigadas abordagens que extrapolem os métodos populares, como o aumento de dados. Para isso, devem ser investigadas abordagens de criação de heurísticas com auxílio de especialistas humanos e detecção de dubiedade em textos de redes sociais.

AGRADECIMENTOS

À FAPESP, processo 2022/03090-0, pelo financiamento da pesquisa.

Ao Interfaces – Núcleo de Estudos Sociopolíticos dos Algoritmos e da Inteligência Artificial do qual os pesquisadores fazem parte.

REFERÊNCIAS

Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, *31*(2), 211–236. doi: 10.1257/jep.31.2.211

Bovet, A., & Makse, H. A. (2019). Influence of fake news in twitter during the 2016 us presidential election. *Nature Communications*, *10*(1), 7. doi: 10.1038/s41467-018-07761-2

Bradshaw, S., Campbell-Smith, U., Henle, A., Perini, A., Shalev, S., Bailey, H., & Howard, P. N. (2021). *Country case studies: industrialized disinformation. 2020 global inventory of organized social media manipulation.*

Chollet, F., et al. (2015). *Keras.* Keras Project. Recuperado de https://keras.io

Huyghe, F. B. (2016). *La désinformation: Les armes du faux.* Colin.

Iasulaitis, S., Valejo, A. D. B., Greco, B. C., Perillo, V. G., Messias, G. H., Vicari, I., ... Intelligence, A. (2025). The interfaces twitter elections dataset: Construction process and characteristics of big social data during the 2022 presidential elections in brazil. *PLOS ONE*, *20*(2), e0316626. doi: 10.1371/journal.pone.0316626

Iasulaitis, S., & Vieira, A. O. (2022). Quando o ataque é o programa: As estratégias de campanha de donald trump e de jair bolsonaro no twitter. *Comunicação & Sociedade*, *44*(2), 5–46. doi: 10.15603/2176-0985/cs.v44n2p5-46

Lorenceti, A. D., & Salton, G. D. (2022). Detecção de fake news em um tweet utilizando machine learning e processamento de linguagem natural. *Brazilian Journal of Development*, *8*(6), 43581–43599. doi: 10.34117/bjdv8n6-071

Maci, S. M., Demata, M., Seargeant, P., & McGlashan, M. (2023). The various dimensions of disinformation: an introduction. In *The routledge handbook of discourse and disinformation* (p. 1–13). Routledge.

Mendonça, R. F., Freitas, V. G., Aggio, C. d. O., & Santos, N. F. d. (2023). Fake news e o repertório contemporâneo de ação política. *Dados: Revista de Ciências Sociais*, *66*(2). doi: 10.1590/dados.2023.66.2.301

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint*. doi: 10.48550/arXiv.1301.3781

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (p. 1532–1543).

Ratner, A., Hancock, B., Dunnmon, J., Sala, F., Pandey, S., & Ré, C. (2019). Training complex models with multi-task weak supervision. In *Proceedings of the aaai conference on artificial intelligence* (v. 33, p. 4763–4771).

Ricard, J., & Medeiros, J. (2020). Using misinformation as a political weapon: Covid-19 and bolsonaro in brazil. *Harvard Kennedy School Misinformation Review*, *1*(3).

Rizzo, A., & Becker, C. (2020). *Se é 'fake', não é 'news'.* El País Brasil. Recuperado de https://brasil.elpais.com/opiniao/2020-08-28/se-e-fake-nao-e-news.html

Rollemberg, M. (2022). *Olhares atentos sobre os males da desinformação.* Jornal da USP. Recuperado de https://jornal.usp.br/cultura/olhares-atentos-sobre-os-males-da-desinformacao/

Saleh, H., Alharbi, A., & Alsamhi, S. H. (2021). Opcnn-fake: Optimized convolutional neural network for fake news detection. *IEEE Access*, *9*, 129471–129489. Recuperado de https://www.researchgate.net/publication/354589045_OPCNN-FAKE_Optimized_Convolutional_Neural_Network_for_Fake_News_Detection

Santaella, L. (2023). Definir desinformação é preciso. In *Flagelos da desinformação.* Educ.

Santana Júnior, C. A., Albuquerque, J. P. S., Queiroz, F. S., & Lima, S. R. (2014). A disseminação da informação no twitter: uma análise exploratória do fluxo informacional de retweets. *AtoZ: novas práticas em informação e conhecimento*, *3*(1), 50–59. doi: 10.5380/atoz.v3i1.41334

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, *19*(1), 22–36. doi: 10.1145/3137597.3137600

Silva, L. J. d., Santos, L. A. d., Araujo, R., Coelho, O. B., Correa, A. G. D., & Oliveira, I. C. A. (2024). Tweet_eleições_2022: Um dataset de tweets durante as eleições presidenciais brasileiras de 2022. In *Xiii brazilian workshop on social network analysis and mining.* Brasília, DF. doi: 10.5753/brasnam.2024.1940

Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. *Science*, *359*(6380), 1146–1151. doi: 10.1126/science.aap9559

Como citar este artigo (APA):
Silva, R. A. B.; Pereira, E. T. & Iasulaitis, S. (2025). Detecção automática de *fake news* em *tweets* em períodos eleitorais. *AtoZ: novas práticas em informação e conhecimento, 14*, 1 – 20. Recuperado de: http://dx.doi.org/10.5380/atoz.v14.95677

NOTAS DA OBRA E CONFORMIDADE COM A CIÊNCIA ABERTA

CONTRIBUIÇÃO DE AUTORIA

Papéis e contribuições	Rafaela de Amorim Barbosa Silva	Eanes Torres Pereira	Sylvia Iasulaitis
Concepção do manuscrito	X	X	X
Escrita do manuscrito	X	X	X
Metodologia	X	X	
Curadoria dos dados	X	X	X
Discussão dos resultados	X	X	X
Análise dos dados	X	X	

FINANCIAMENTO

O(s) autor(es) declara(m) que esta pesquisa recebeu financiamento conforme dados indicados a seguir e o documento comprobatório foi anexado como documento suplementar: **Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) - processo 2022/03090-0**

EQUIPE EDITORIAL