Universidade Católica do Salvador
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas

Wilton O. Júnior[1], Mauricio S. da Cruz[2]

O uso de Data Augmentation como técnica para o aprimoramento de redes neurais a fim de detectar notícias falsas sobre a COVID-19

SALVADOR
2021

Wilton O. Júnior[1], Mauricio S. da Cruz[2]

O uso de *data augmentation* como técnica para o aprimoramento de redes neurais a fim de detectar notícias falsas sobre a COVID-19

Trabalho de Conclusão de Curso apresentado à Universidade Católica do Salvador como parte dos requisitos necessários para a obtenção do Título de Tecnólogo em Analise e Desenvolvimento de Sistemas.Orientador: Prof André Brasil Wyzykowski

Universidade Católica do Salvador

SALVADOR
2021

O uso de *data augmentation* como técnica para o aprimoramento de redes neurais a fim de detectar notícias falsas sobre a COVID-19

Wilton O. Júnior[1], **Mauricio S. da Cruz**[2]

[1]Instituto de Informática – Universidade Católica do Salvador (UCSAL)
Av. Prof. Pinto de Aguiar, 2589 - Pituaçu, Salvador - BA, 41740-090

Abstract. *This paper aims to present how the application of Natural Language Processing (NLP) and data augmentation techniques can improve the performance of a neural network for better detection of fake news in the Portuguese language. Fake news is one of the main controversies during the growth of the internet in the last decade. Verifying what is fact and what is false has proven to be a difficult task, while the dissemination of false news is much faster, which leads to the need for the creation of tools that, automated, assist in the process of verification of what is fact and what is false. In order to bring a solution, an experiment was developed with neural network using news, real and fake, which were never seen by artificial intelligence (AI). There was a significant performance in the news classification after the application of the mentioned techniques.*

Resumo. *Esse trabalho tem como proposta apresentar como a aplicação de técnicas de Processamento de Linguagem Natural (NLP) e data augmentation podem melhorar a desempenho de uma rede neural para melhor detecção de notícias falsas na língua portuguesa. Notícias falsas são uma das principais polêmicas durante o crescimento da internet na última década. Averiguar o que é fato e o que é falso mostrou-se uma difícil tarefa, ao passo que a disseminação de notícias falsas é muito mais rápida, o que leva à necessidade da criação de ferramentas que, automatizadas, auxiliem no processo de averiguação sobre o que é fato e o que é falso. De modo a trazer uma solução, foi desenvolvido um experimento com a rede neural utilizando notícias, reais e falsas, as quais nunca foram vistas pela inteligência artificial (IA). Houve um desempenho significativo na classificação das notícias após a aplicação das técnicas mencionadas.*

1. Introdução

A pandemia da COVID-19 surgiu inserida em um contexto em que as pessoas se informam mais por manchetes nas redes sociais do que na matéria de fato, abrindo brechas para o fenômeno das notícias falsas. A descoberta de um novo vírus demanda notícias que tratam dos sintomas, gravidade, formas de transmissão. E como tudo é novo e volátil, configura-se um terreno fértil para a inserção da desinformação, principalmente em um panorama em que não se observa o hábito de checar a veracidade das informações. A produção de notícias falsas acontece a uma velocidade humanamente impossível de ser combatida por um pequeno grupo de jornalistas e esses também devem cobrir as novidades vindas de fontes confiáveis. Dado esse

contexto, a automação dessa atividade parece ser a única forma efetiva do combate às notícias falsas através da Inteligência Artificial (IA).

Vale ressaltar que o combate a notícias falsas possui uma complexidade em relação ao peso de determinar algo como sendo falso ou verdadeiro. Tendo em vista que existe a capacidade de se inovar na própria produção de novas fraudes. Por isso trabalhos como esse e outros citados nesse artigo possuem bastante relevância nesse contexto de pandemia.

Diversos autores como Madani, Erritali and Bouikhalene 2021, Patwa et al. 2020 e Mookdarsanit and Mookdarsanit 2021 já trouxeram propostas de algoritmos, muitos usando técnicas da IA moderna como Linguagem de Processamento Neural (NLP) ou Perceptron multi-camadas (MLP). A maior parte desses trabalhos está relacionada a *contenders* promovidos na plataforma Kaggle[1]. Alguns autores como Ding et al. 2020 e Patwa et al. 2020 ultilizam o termo *infodemic* para descrever esse fenômeno das notícias falsas sobre o coronavírus.

Algumas redes sociais como o Twitter e o Facebook, divulgaram ferramentas ao combate de notícias falsas, no caso do Facebook, uma série de autenticações para garantir que o responsável pela postagem é de fato o dono da conta (CanalTech 2021) e o Twitter, traz o Birdwatch[2], uma ferramenta a qual o usuário pode denunciar uma postagem como uma notícia falsa. Porém, há diversas razões para suspeitar do interesse dessas corporações no combate a essa prática já que, ao averiguar a fonte, o usuário sai da plataforma, indo ao encontro do modelo de negócio de uma rede social que focado em engajamento.

2. Fundamentação Teórica

Com o avanço da pandemia da COVID-19 a discussão se o acesso à *internet* é um dos direitos fundamentais para a sobrevivência (assim como água, luz e saneamento básico) deixou de ser polêmica e passou a ser consenso, dado o fato que a dinâmica da sociedade moderna exige que, para um cidadão estar incluído, ele deve estar conectado. Almeida 2020, fala como a democratização da internet é fundamental para a manutenção do direito básico a educação previsto na constituição brasileira de 1988. Segundo o Collins Dictionary, *lockdown* é a palavra do ano de 2020, tendo em vista que o dicionário registrou, cerca de, 250 mil usos contra apenas 4 mil do ano anterior (Collins 2020).

Durante muitos anos, a credibilidade da mídia fosse ela impressa, radiofônica ou televisiva, era praticamente irrefutável. Hoje, com a democratização do acesso, e principalmente, da produção de informação e conteúdo, novos veículos, sites, *blogs*, canais no YouTube e outras plataformas, passaram atuar também como imprensa (Kalsnes 2018). Kalsnes 2018 continua dizendo que, mesmo após a notícia falsa ser desmentida, ela ainda causa efeito de influência na população, dado que ela ataca não apenas os fatos, mas também a credibilidade de importantes instituições como os veículos de imprensa. Além disso, as redes sociais possuem um impacto tremendo na forma como as pessoas pensam e mostram-se armas poderosas de manipulação,

[1]Kaggle: comunidade on-line de cientistas de dados e profissionais de aprendizado de máquina subsidiária da Google.

[2]Fonte: https://tinyurl.com/ferramentamessenger

levando grupos políticos a criar notícias falsas ou disseminar meias verdades, de modo a criar ou influenciar um movimento.

Em 2020, alguns autores realizaram trabalhos analisando o impacto e o perigo das notícias falsas durante a pandemia. Kalsnes cita em seu trabalho que as principais razões por trás de uma notícia falsa incluem questões políticas, financeiras e sociais (Kalsnes 2018). Neto et al. 2020 Mostra como uma série de notícias falsas pode estar a serviço de uma ideologia como a privatização do Sistema Único de Saúde brasileiro (SUS), visto que esses falsos artigos tentam descredibilizar um órgão de tamanha tradição e importância.

Ainda em sua análise Neto et al. 2020 conta que as notícias falsas no contexto brasileiro sobre a COVID-19 se dividem em 5 categorias. São elas: informações relacionadas a autoridades de saúde, terapêutica, medidas de prevenção, prognóstico da doença e vacinação. Esses são os 5 pontos principais e o seus principais veículos são as redes sociais, em especial o *WhatsApp*.

O poder de notícias falsas foi fortemente subestimado no passado. Das eleições de 2016 nos Estados Unidos às eleições no Brasil em 2018 e agora durante a pandemia do coronavírus, onde o presidente Bolsonaro faz uma acusação contra a Rede Globo e o jornal Folha de SP em 2018 (Pereira 2021). Comportamento este similar ao do presidente dos Estados Unidos, Donald Trump, que diversas vezes classificou informações provenientes da CNN e *New York Times* como notícias falsas (Kalsnes 2018).

Os esforços para combater as notícias falsas se mostraram bastante desafiadores para os métodos tradicionais já que a produção de uma notícia falsa exige menos tempo e esforço do que desmenti-la. Um desses esforços é a Lupa, um projeto feito pelo jornal Folha de São Paulo, que é uma agência de *Fact-Checking* (Checagem dos Fatos, tradução livre), a primeira do Brasil (Folha 2021). Entretanto, tem-se observado que existe um padrão nas notícias falsas criado pelo contexto da pandemia e, através, dele é possível automatizar essa validação.

3. Trabalhos Correlatos

Diversos pesquisadores ao redor do mundo tem se dedicado a criar soluções para a detecção das notícias falsas. Ainda há muito debate com relação a melhor arquitetura de uma IA com esse objetivo. Singh 2020 Fez um trabalho comparando diversas técnicas da inteligência artificial moderna como Rede Neural Convolucional (CNN), Rede Neural Artificial (ANN) e Rede Neural Recorrente (RNN) e a acurácia variava bastante conforme o *dataset* e o preprocessamento dos dados antes do treinamento.

Por se tratar de uma classificação, é necessário retratar a similaridade dos dados através de uma representação vetorial. Isso consiste basicamente entre analisar a curva entre, os resultados esperados com os resultados obtidos e assim obter a acurácia e o *LOSS*, ou seja, o quanto se perde, de um treinamento.

Singh 2020 em seu trabalho, fez um comparativo entre as técnicas *Word2Vet*, *One-Hot Encoding*, *Doc2Vec* e TF-IDF, utilizadas vetorização de texto, utilizados baseando-se nas arquiteturas CNN, ANN e RNN. Este concluiu que os dados não variam muito conforme a arquitetura e sim, graças ao *dataset*, que utilizando a

técnica TF-IDF com o *dataset* Kaggle, obteve-se resultados satisfatórios na faixa de 0,96.

Em 2021, Shushkevich and Cardiff 2021 participou do *TUDublin team at Constraint@AAAI2021 - COVID19 Fake News Detection*, uma competição na qual pesquisadores com o mesmo *dataset* procuram obter a melhor acurácia possível, onde, em 2021, o tema foi justamente a pandemia do novo coronavírus. Ao final do projeto, eles obtiveram entre 0,91 e 0,95 de acurácia, o que são resultados bastante animadores dada a urgência dessa categoria de trabalho.

Para isso, os pesquisadores analisaram os termos mais frequentes utilizando TF-IDF, removeram os afixos, terminações morfológicas e inflexionais mais comuns para manter apenas a ideia central da palavra com o algoritmo de *PorterStemmer* e, por fim, alteraram todo o texto para letras minúsculas. Após o pré-processamento, seu modelo usou a seguinte arquitetura: *logistic regression*[3], SVM[4], *Naive Bayes*[5] e uma combinação de *Naive Bayer* com *logistic regression*.

Em outro trabalho correlato, pesquisadores da Acenture, (Paka et al. 2021) criaram o *Cross-SEAN*, que tem como base a análise de mensagens vindas das redes sociais, em especial o Twitter. Para isso eles dividiram o processo em quatro etapas. A primeira foi separar os *tweets* relacionados ao coronavírus, tendo como base as postagens disponíveis no *dataset Kaggle*, foram pegos os IDs desses *tweets* e pesquisados na API[6] do próprio Twitter. Depois coletaram informações confiáveis sobre a pandemia utilizando ferramentas já existentes para checagem de fatos (*Snopes, PolitiFact, FactCheck e TruthOrFiction*). De lá, retiram URLs[7] com tópicos relacionados a COVID-19, classificando os *tweets* entre genuínos e falsos com relação à pandemia utilizando técnicas de manipulação de textos como BERT[8] e RoBERTa[9]. A última etapa realizada foi a validação manual de uma amostra dos resultados obtidos. Após todo o processo, foi constatado uma acurácia de 92% entre as falsas postagens detectadas automaticamente.

Apesar de excelentes iniciativas, sentimos falta em todos esses trabalhos da aplicabilidade deles em notícias, já que são por meio delas que a opinião pública se baseia dado ao fato de confiar na credibilidade do órgão de imprensa.

4. Materiais e métodos

Essa pesquisa foi adaptada com base em um trabalho escrito por outro autor (singularity014 2021). O seu repositório pode ser encontrado nas referências deste

[3]Logistic Regression: técnica estatística que tem como objetivo produzir um modelo que permita a predição de valores tomados por uma variável categórica.

[4]SVM: conceito na ciência da computação para um conjunto de métodos de aprendizado supervisionado usado para classificação e análise de regressão.

[5]Naive Bayes: família de "classificadores probabilísticos"simples baseados na aplicação do teorema de Bayes.

[6]API: conjunto de rotinas e padrões de programação para acesso a *software* ou plataforma baseado na Web.

[7]URL: endereço virtual de uma página ou website.

[8]Modelo BERT: técnica utilizada para pré-treinamento e processamento de linguagem natural desenvolvida pelo Google para aprendizado de máquina.

[9]RoBERTa: modelo de pré-treinamento baseado em BERT altamente otimizado

artigo. Nossa proposta é aumentar a acurácia desse algoritmo aplicando técnicas de *data augmentation* em textos, além de atuar na tradução dos *dataset* de modo a tentar trazer a aplicação da pesquisa para território nacional devido à escassez de conjuntos de dados a serem utilizados sobre o tema proposto.

Apesar de haver bons trabalhos como o Fake Br Corpus (Santos 2021) e (Dias 2019) em português brasileiro para as notícias falsas, eles não contemplam o contexto da pandemia do novo coronavírus. Existe uma lacuna entre os trabalhos disponíveis em português e as informações da COVID 19 e isso acaba se tornando um desafio nessa área. A única alternativa encontrada foi a LATAM Coronavírus (Chequeado 2021), uma base de dados constantemente atualizada, utilizada para checar, validar e explicar informações sobre o novo coronavírus. Entretanto, essa base de dados não se encontra bem estruturada para a análise de notícias. Todo o código está disponível no Google Colab[10] e o link se encontra nas referências desse artigo, bem como o link para o Github[11] com o projeto do autor o qual utilizamos como base.

5. Tecnologias utilizadas

Neste tópico, abordaremos todas as tecnologias aplicadas durante o processo de desenvolvimento deste trabalho.

5.1. *Python*

Lançada por Guido Van Rossum em 1991, Python é uma linguagem de programação de alto nível, que pode ser utilizada tanto para programação orientada a objetos (POO) quanto para programação estruturada. A linguagem foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional (PYTHON 2021).

Há várias razões pelas quais os cientistas de dados usam Python. Estes precisam criar visualizações de dados para comunicar de forma clara os resultados e as previsões em qualquer nível de um negócio. Pensando nisso, a linguagem Python detém grande vantagem por conter bibliotecas, *frameworks*[12] e pacotes exclusivos para se utilizar na área de dados, como o *sci-kit learn* para *Machine Learning*, *Numpy*, *Pandas* para análise de dados, dentre outros (UFC 2021).

5.2. *Dataset*

Dataset (ou conjunto de dados) é o termo utilizado para uma coleção de informações (dados) tabulados, onde cada coluna representa um valor variável e particular de cada linha corresponde a um conjunto de dados em questão.

Os *datasets* podem conter informações, como registros médicos ou registros de seguros, para serem usados por um programa em execução no sistema. Estes

[10]Google Colab: O Colaboratory ou "Colab"permite escrever código Python no seu navegador sem nenhuma configuração necessária e com acesso gratuito a GPUs.

[11]Github: plataforma de hospedagem de código-fonte e arquivos que permite qualquer usuário cadastrado na plataforma contribuir em projetos privados e/ou Open Source.

[12]Framework: conjunto de códigos prontos que podem ser utilizadas no desenvolvimento de aplicações.

também são usados para armazenar informações necessárias aos aplicativos ou ao próprio sistema operacional, como programas de origem, bibliotecas de macros[13] ou variáveis ou parâmetros do sistema. Estes podem ser catalogados, o que permite que o *dataset* seja referido pelo nome sem especificar onde está armazenado (IBM 2021).

5.3. *Natural Language Processing* (NLP)

É uma subárea da Inteligência Artificial que dá ao computador a habilidade de entender, analisar, manipular e reproduzir a fala / escrita humana com uma precisão considerável. Ela também é utilizada para estudar e compreender a linguagem humana natural em sua totalidade (NLP 2021).

Atualmente o próprio buscador da gigante da tecnologia, a Google, utiliza NLP para encontrar e devolver resultados similares as palavras-chave digitadas no campo de busca. Além disso, ela utiliza a NLP no Google Translate, fazendo com que a tradução de uma língua para outra seja muito mais eficiente, não deixando o texto perder o sentido do contexto (Google 2021).

5.4. Tokenização

A tokenização é a técnica utilizada para quebrar a sequência de caracteres em um texto localizando o limite de cada palavra, separando cada termo de uma frase em tokens, termo utilizado para fins computacionais. A tokenização pode normalizar um texto, por exemplo, mapeando suas palavras para versões apenas com letra minúscula, expandindo contrações e até mesmo extraíndo o radical de cada palavra (NLTK 2021).

5.5. *SpaCy*

SpaCy é uma biblioteca de código aberto utilizada na linguagem de programação Python para NLP (Processamento de Linguagem Natural). Essa biblioteca já possui modelos pré-moldados que atuam como "cérebros treinados" para cada língua, facilitando o seu uso para diversos fins (PyPi 2021).

Além de ser utilizada em NLP, essa biblioteca contém ferramentas de processamento *POS-Tagging* (*Part-Of-Speeching Tagging* (ou simplesmente entender como "Análise Gramatical") para detecção de adjetivos, substantivos entre outras classes gramaticais no texto, além de também poder ser utilizada como NER (*Named Entity Recognition*) para determinar se um termo se refere a uma pessoa, data, lugar, organização, entre outros, de modo a melhorar a acurácia na identificação correta dos termos.

5.6. PyDictionary

PyDictionary é uma biblioteca para obter significados, traduções, sinônimos e antônimos de palavras. Ele usa WordNet para obter significados, Google para traduções e o site synonym.com para obter sinônimos e antônimos (PyPi 2021).

[13]Macro: sequências de eventos programados (como pressionamentos de tecla, cliques e atrasos) que auxiliam em tarefas repetitivas.

5.7. Rede Neural (*Neural Network*)

Estes são modelos computacionais inspirados em neurônios do cérebro humano. Com o devido treinamento, podem reconhecer padrões e correlações em dados brutos, além de conseguir agrupá-los e classificá-los e, com o tempo, poder aprender de forma contínua. Sua primeira aparição foi em 1943 por Warren McCulloch e Walter Pitts. A partir de um artigo criado sobre como os neurônios funcionavam, estes modelaram uma rede neural simples utilizando circuitos eletrônicos (Stanford 2021).

5.8. *Data Augmentation*

A técnica de *data augmentation* consiste, basicamente, em manipular o *dataset* de forma a criar dados semelhantes para o treinamento de modo a aumentar a quantidade de dados, adicionando cópias ligeiramente modificadas de dados já existentes ou sintéticos recém-criados a partir de dados existentes. Todavia, para que um *data augmentation* seja eficiente, a manipulação não pode descaracterizar o *dataset*. Ou seja: uma recomendação da Organização Mundial de Saúde (OMS) ainda deve ser verdadeira e corresponder com o contexto da COVID-19 (TensorFlow 2021).

6. Desenvolvimento

O ponto-chave dessa pesquisa é aprimorar um conjunto de dados pré-formatados de modo a aumentar a quantidade de informações contidas nele através da técnica de *data augmentation*, para que, após o treinamento, a rede neural possua uma porcentagem de acurácia superior a anterior, aumentando sua eficácia. Porém, ainda há outro fator que precisamos considerar: o *dataset* escolhido está na língua inglesa. O que significa que toda a essa abordagem teria que ser feita obedecendo às regras gramaticais da língua. Houve a tentativa de buscar um *dataset* na língua nativa (português). No entanto, o Brasil carece de conjunto de dados pré-definidos para poderem ser livremente analisados neste contexto. Por conta disso, a escolha do *dataset* estrangeiro foi a opção mais rápida e próxima encontrada para o desenvolvimento da pesquisa.

No total, separou-se 04 *datasets*: um para o treino real (*Real Train*), um contendo notícias reais que serão usadas para testar (*Real Test*), um para o treino *fake* (*Fake Train*) e o último contendo notícias fake (*Fake Test*), sem a necessidade de serem utilizados nessa ordem, onde todos os *datasets* "*Train*" possuem 800 notícias e os *datasets* "*Test*" possuem 200.

O desafio então foi traduzir todos os *dataset* para o português de modo a manter toda a estrutura gramatical inteligível para que a rede neural mantivesse a acurácia ou tivesse a menor perda possível, pois, como a validação ocorrerá com notícias nacionais (reais ou falsas), a translação de uma língua para outra poderá impactar significativamente na precisão de análise da rede neural. Para isso, foi utilizada a biblioteca *Google Translate* para fazer a tradução de maneira adequada. O processo de tradução levou 36 horas para ser concluído. Esse fator se agravou por conta que a biblioteca escolhida não traduzia textos que continham mais que 5.000 (cinco mil) caracteres. O que levou a seguir-se instruções específicas:

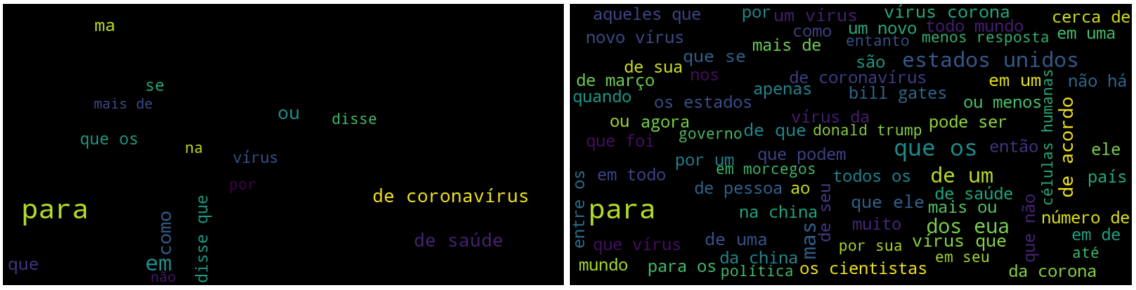
1. Criar um loop para percorrer todo o *dataset*
2. Verificar se o texto atual possui mais que 5.000 caracteres

3. Caso sim, traduzimos o texto frase por frase, detectando através de um ponto final (.) no texto.
4. Caso não, a biblioteca traduz todo o texto sem tratamento.

```
def translate_text(word):
  translator = google_translator()
  tword = translator.translate(word, lang_src=''en'',lang_tgt=''pt'')
  time.sleep(1)
  return tword
```

Listing 1. Código fonte em Python. Fonte: O Autor

Após o processo de tradução feito pela função mostrada no Listing I, era preciso visualizar as palavras mais comuns em notícias reais ou falsas, com base nos *datasets* utilizados, para descobrir algum padrão que os diferenciasse. Para isso, foi utilizada a biblioteca *WordCloud*, eficiente para visualizar as palavras mais utilizadas em ambas as situações. Foi gerado uma *word cloud* para o *dataset* traduzido "*Real Train*" e outra *word cloud* para o *dataset* traduzido "*Fake Train*". Os *datasets* "*Test*" são apenas para fins de validação.



(a) Palavras mais comuns em notícias reais (b) Palavras mais comuns em notícias falsas

Figura 1. *Word clouds* após processamento.
Fonte: O Autor

A Figura 1 (a) indica as palavras mais comuns em notícias reais baseado no *dataset* "*Real Train*". Porém, ela mostra uma falha crítica: ao traduzir o *dataset* , a *word cloud* aparenta uma escassez de palavras que, em relação ao tamanho do conjunto de dados escolhido, não reflete a quantidade ideal de palavras, enquanto a *word cloud* mostrada na Figura 1 (b), do *dataset* " "*Fake Train*", apresenta uma constância de palavras muito maior. É notado que, quando os textos são traduzidos, as palavras com contextos neutros da língua portuguesa tendem a se destacar como palavras mais usadas em notícias reais. No caso do *dataset fake*, as palavras em comum tendem a ter mais diversidades, como nomes próprios, de países, contextos políticos, entre outros. Em suma, o que consegue-se destacar é:

1. A *word cloud* do *dataset* "*Real Train*" indica que as palavras mais comuns são aquelas com contextos neutros ("para", "que", "como", "mais de"...)
2. Já a *word cloud* do *dataset* "*Fake Train*" já abrange palavras chaves contextualizadas, como "Bill Gates", "Donald Trump", "Morcegos", "células humanas", dentre outras palavra.

No entanto, é necessário levar em consideração que a *word cloud* do *dataset "Real Train"* mantém, ao que parece, uma falsa representatividade do conjunto de dados utilizado. Isso porque, dado o tamanho de ambos os conjuntos de dados, as figuras precisariam apresentar uma quantidade de palavras o mais próximo possível uma da outra. Todavia, a Figura 1 (a) mostra uma exacerbada falta de palavras em relação à Figura 1 (b).

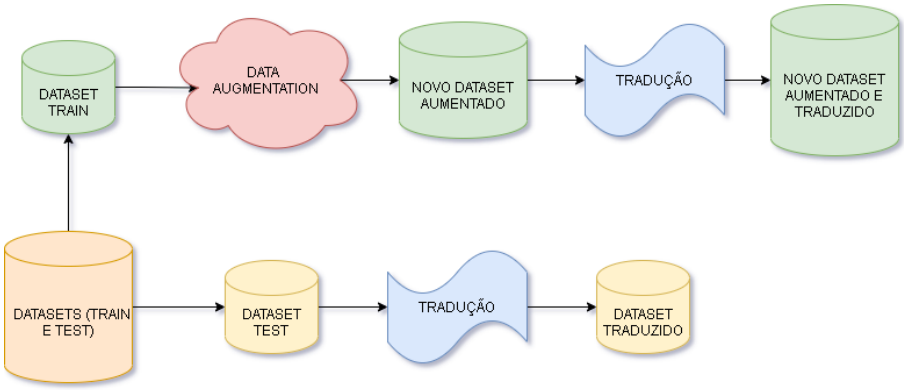


Figura 2. Processo de *data augmentation* e tradução de *dataset*.
Fonte: O Autor

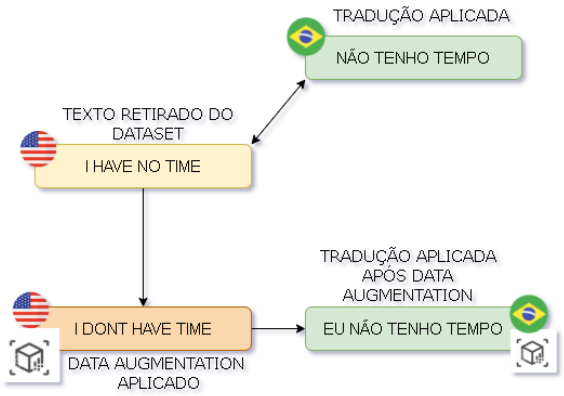


Figura 3. Exemplo de texto traduzido sem o *data augmentation* e traduzido com a técnica aplicada.
Fonte: O Autor

Pensando nisso, foi construído um fluxo, como é mostrado na Figura 2, onde o *data augmentation* é aplicado antes mesmo da tradução ocorrer. Com esse processo, ambos os *datasets* ficariam muito mais abundantes e bastante contextualizados. Para aplicação dessa técnica, serão utilizadas duas bibliotecas: *SpaCy* e *PyDictionary*, para aplicar o NLP e buscar por sinônimos das palavras escolhidas, respectivamente. Esse exemplo também pode ser notado na Figura 3, a qual mostra como ficaria uma frase se traduzida em sua íntegra e como ficaria a mesma frase se aplicássemos primeiro o *data augmentation*.

Com o objetivo de não descaracterizar as notícias, a primeira regra que aplicamos foi: todas as palavras trocadas serão substantivos. Verbos, pronomes e adjetivos podem facilmente sair do contexto caso sejam substituídos por sinônimos.

A segunda regra é: o substantivo só é trocado caso a sua semelhança seja de pelo menos 40 por cento em relação ao sinônimo escolhido. Esse número foi tirado com base dos parâmetros da biblioteca *SpaCy*, conhecida como "índice de similaridade". Ainda que esse número possa parecer baixo, a biblioteca é bastante exigente com relação a sua semelhança. Caso o valor seja muito alto, o *data augmentation* não ficará robusto. Portanto, não será impactante durante o treinamento.

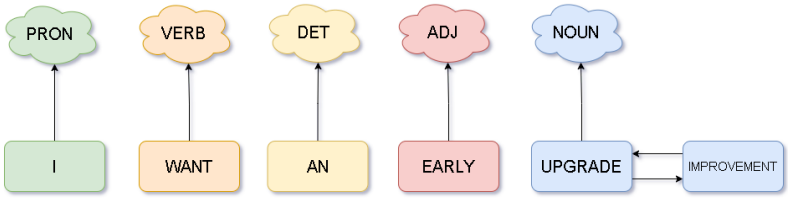


Figura 4. Exemplo de como é feito o processo de tokenização, detecção das classes gramaticais e alteração de palavra por sinônimo.
Fonte: O Autor

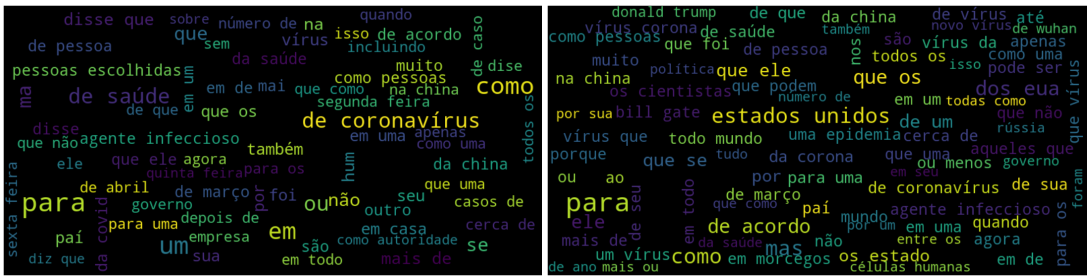
Para a primeira regra, utilizou-se a técnica POS-Tagging (Part Of Speech Tagging), como mostrado no exemplo da Figura 4, a qual a biblioteca já está possui um modelo pré-treinado para ser utilizado, separando cada palavra em um *token*[14] para ser analisado de forma unitária. A técnica de POS-Tagging serve para identificar qual classe gramatical aquela palavra pertence (pronome, substantivo, verbo, adjetivo, entre outros). Com isso, serão somente identificadas palavras na classe "substantivo".

Para a segunda regra, aplicamos a biblioteca PyDictionary para buscar sinônimos para o substantivo tokenizado, como também é mostrado na Figura 4. Como essa biblioteca pode retornar uma lista contendo vários substantivos, a depender da palavra, utilizamos o índice de similaridade da biblioteca *SpaCy*[15] para identificar se esta possui similaridade acima de 40 por cento em relação ao *token*, substituindo-o pelo sinônimo com maior percentual caso sim. Existe também a possibilidade de um substantivo não possuir sinônimos. Neste caso, seguimos sem alterar a palavra.

Esse processo é repetido para cada notícia em um *dataset*. Ao todo, o processo levou 60 horas para ser concluído. Após a conclusão do *data augmentation* nos *datasets* "*Train*" e o processo de tradução, gerou-se novamente a *word cloud* de ambos os *datasets* (*Real* e *Fake*) para saber se houve alguma alteração significativa nas palavras mais comuns de cada uma.

Em ambas as Figuras 5 (a) e (b), é possível identificar que houve um crescimento bastante significativo nos termos mais comuns apresentados na *word cloud* dos *datasets*. Também é possível notar que a Figura 5 (b) em específico, *word cloud* do "*Fake Train*", está um pouco mais preenchida de palavras em relação à Figura 5 (a). A partir dessas informações, pode-se indicar que o processo de *data augmentation* impactou em mudanças significativas em ambos os conjuntos de dados.

[14]Token: Palavra de um texto separada em uma unidade independente.
[15]SpaCy: spacy.io



(a) Palavras mais comuns em notícias reais após *data augmentation.*

(b) Palavras mais comuns em notícias falsas após *data augmentation.*

Figura 5. *Word clouds* após *data augmentation.*
Fonte: O Autor

6.1. Modelagem e Treinamento

Para a fase de treinamento usou-se uma arquitetura baseada no modelo de classificação BERT. Esse modelo tem sido bastante usado para classificação de textos por conta do seu sucesso em diversas categorias de arquitetura. Segundo os próprios criadores do BERT (Devlin et al. 2018) este modelo é pré-treinado em um grande corpo de texto e, em seguida, ajustado para ser utilizado em tarefas específicas.

Para esse trabalho, além do BERT, foram usadas técnicas de *Droupout*[16] e *Dense Layer*[17] com 5 camadas. Todo o treinamento acontece em 10 épocas[18] e 3 épocas de paciência. Paciência é um indicador referente ao número de épocas sem melhora da acurácia antes do treinamento parar. Ou seja, caso não haja uma melhora em uma época, se a próxima, tiver um desempenho negativo, o treinamento para. Isso evita o crescimento do *LOSS* e mantém a acurácia em seu melhor momento.

A Figura 6 representa como é arquitetada a estrutura dos neurônios. Como o algoritmo BERT só pode aceitar sentenças de até 512 palavras, é necessário pré-processar os dados (notícias longas) para alimentar o algoritmo. Para fazer isso, o autor segue essa ideia e segmenta cada parte do texto em subtexto múltiplo de no máximo 150 palavras. Os subtextos sofrerão uma sobreposição onde, especificamente, as últimas 30 palavras do primeiro subtexto serão as primeiras 30 palavras do segundo subtexto.

Em seguida, o texto original é pré-processado em recursos de entrada que o BERT possa ler. O processo consiste basicamente em tokenizar e converter o texto original em ID's de token que podem ser lidos pelo algoritmo. As palavras são tokenizadas com base no dicionário de vocabulário em que foram pré-treinadas (cerca de 30.000 palavras), e as palavras desconhecidas são divididas em palavras

[16] *Dropout*: Técnica para reduzir co-adaptações complexas de neurônios, forçando um neurônio a aprender recursos mais robustos que são úteis em conjunto com muitos subconjuntos aleatórios diferentes dos outros neurônios.

[17] *Dense Layer*: É uma camada na qual cada neurônio de entrada está conectado ao neurônio de saída, sendo que as unidades de parâmetros apenas informam a dimensão de sua saída

[18] Época de treino: refere-se a um ciclo de treinamento com todo o *dataset*. Normalmente, o treinamento de uma rede neural leva mais do que algumas épocas.

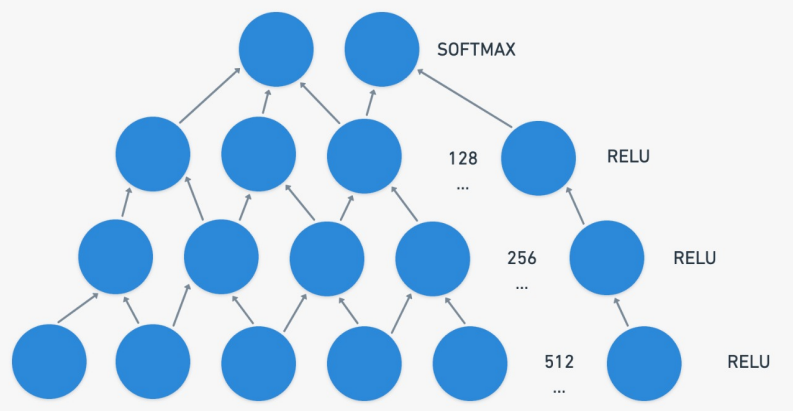


Figura 6. Arquitetura dos neurônios da rede neural.
Fonte: O Autor

menores contidas no dicionário. O comprimento máximo da sequência também é especificado para preencher todas as sequências com o mesmo comprimento [19].

Após toda a preparação dos textos, avanço para o treinamento da rede neural utilizando os *datasets* traduzidos e aumentados. A estrutura do treinamento feita pelo autor utilizava 5 épocas. Como os *datasets* tiveram o seu tamanho quase dobrado, o treinamento foi alterado para 10 épocas para que não haja perda de pro eficiência em caso de um curto limite de tempo de treinamento.

	ANTES DO DATA AUGMENTATION	DEPOIS DO DATA AUGMENTATION
LOSS DA VALIDAÇÃO	0.3638	0.3297
ACURÁCIA DA VALIDAÇÃO	0.9487	0.9172
ACURÁCIA GERAL	0.9516	0.9913
LOSS	0.3611	0.0293

Tabela 1. Tabela comparativa indicando os valores antes e depois da aplicação da técnica.
Fonte: O Autor

Os dados plotados na Tabela 1 indicam que houve um aumento significativo na precisão do treinamento e diminuição do percentual de *loss*, apresentando também uma queda significativa em relação à acurácia da validação, um dos indicadores que corresponde aos dados reais em comparação com as predições.

6.2. Validação e Resultados

Após o treinamento, utilizou-se a precisão como métrica principal para a validação do modelo, além da comparação dos valores anteriores ao processo de *data augmentation*, para analisar se houve alguma mudança em três parâmetros importantes: “Acurácia do Treino”, “Acurácia da Validação” e “Acurácia do Teste”, onde cada item representa a precisão de cada passo (“treino”, “teste” e “validação”).

Pode-se notar na Tabela 2 que houve uma queda nos valores da ”Acurácia de Validação”e da ”Acurácia de Teste”após a aplicação do *data augmentation*. Apesar

[19]O comprimento final da sequência seria maior que o especificado, pois o tokenizer BERT dividirá palavras desconhecidas em várias palavras pequenas conhecidas.

	ANTES DO DATA AUGMENTATION	DEPOIS DO DATA AUGMENTATION
ACURÁCIA DO TREINO	0.9596157073974609	0.9944672584533691
ACURÁCIA DA VALIDAÇÃO	0.954023003578186	0.9166180491447449
ACURÁCIA DO TESTE	0.9438775777816772	0.9254571199417114

Tabela 2. Tabela comparativa indicando os valores antes e depois da aplicação da técnica

Fonte: O Autor

do aumento considerável no indicador "Acurácia de Treino", a queda dos outros indicadores pode estar associada a tradução das notícias para o português. De qualquer forma, os valores pós *data augmentation* se mantiveram firmemente acima de 91 por cento.

Para validação e de modo a verificar as taxas de acerto em notícias em que a rede neural nunca viu antes, foi abordado um estudo de caso com as seguintes métricas:

1. Analisar 20 notícias reais e 20 notícias fakes, ambas retiradas dos *datasets* "*Real Test*" e "*Fake Test*".
2. Analisar 20 notícias reais e 20 notícias fakes, ambas retiradas de sites e postagens reais em território brasileiro.

Primeira validação: no teste com o *dataset* "*Fake Test*", a rede neural obteve 95 por cento de acerto (19 das 20 notícias), além de melhorias significativas na acurácia. No teste com as notícias reais, a rede neural obteve 95 por cento de acurácia (19 das 20 notícias), também apresentando melhoria significativa nos valores da acurácia.

Segunda validação: no teste com notícias reais, a rede neural obteve 95 por cento de acerto (19 das 20 notícias), apresentando uma boa estabilidade na acurácia entre as notícias. Já no teste com notícias falsas, a rede neural obteve 70 por cento de acerto (14 das 20 notícias). Observando o último teste dessa etapa de validação, foi decidido gerar uma *word cloud* com as palavras mais comuns para entender porque a rede neural errou em certas notícias.

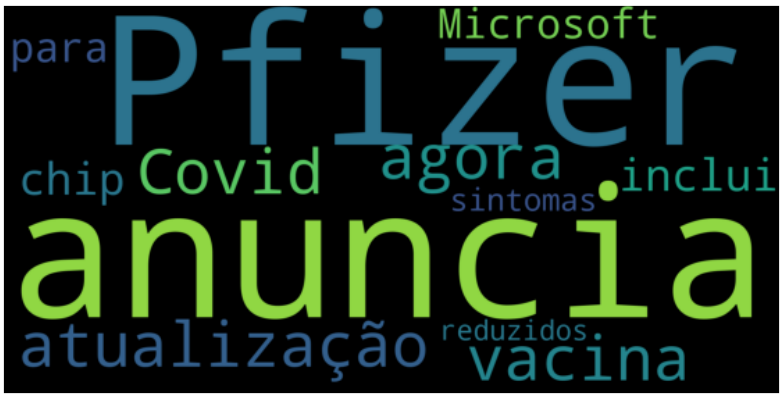


Figura 7. Palavras mais comuns nas notícias com classificação equivocada.
Fonte: O Autor

Na Figura 7 é possível notar que palavras como "Pfizer", "*chip*" e "Microsoft" aparecem como palavras-chave mais comuns entre as frases onde a rede neural aplicou uma classificação equivocada. Isso ocorre por conta que, no *dataset*, não há informações consistentes contendo estas palavras (o mais próximo de "*Microsoft*" seria "Bill Gates", mas a rede neural não foi treinada para relacionar termos). Como o *dataset* é de origem estrangeira, é possível que, em outros países, as notícias falsas sejam criadas com características um tanto quanto diferentes das quais estamos habituados, sendo a adaptação para o contexto brasileiro um verdadeiro desafio.

Também é possível explicar o caso da primeira validação: ambas as notícias analisadas foram provenientes de *datasets* que também foram traduzidos. Isso pode explicar a diferença do percentual entre as duas etapas de validação. Além disso, antes mesmo da tradução dos *datasets* a IA já classificava falsos positivos em testes, informação que pode ser considerada através dos percentuais de acurácia mostrados na Tabela 1 e 2

Todos esses dados podem ser consultados no Github[20]

7. Trabalhos futuros

Ao decorrer deste trabalho, surgiram algumas funções que poderiam auxiliar no processo de treinamento da IA, bem como na acurácia final de classificação:

1. Análise correlacional de palavras: como foi possível notar na segunda etapa de validação de classificação de notícias, a rede neural não conseguiu relacionar a palavra "Microsoft" com "Bill Gates" porque ela não foi treinada para isso. Essa prática pode ser implementada durante o processo de *data augmentation* como uma forma de buscar relações entre palavras do mesmo contexto (não somente por sinônimos) e construir um *dataset* alternativo para que a rede neural possa classificar notícias com maior acurácia.
2. Adição de notícias brasileiras aos *datasets*: reais ou falsas, adicionar mais mensagens no contexto brasileiro, não somente traduzir notícias estrangeiras, pode ter um impacto significativo na rede neural durante a classificação de notícias nacionais.
3. Classificação do tipo de notícia falsa: implementar uma estrutura onde a rede neural possa identificar além de somente notícias reais ou falsas, podendo subclassificar notícias falsas como enganosas, parcialmente falsas, insustentável (quando não há evidências estatísticas que comprovem, por exemplo), dentre outros tipos.

.

8. Conclusões

Como os testes demonstram, o *data augmentation* mostrou eficácia no aumento da acurácia das análises. Foi possível detectar que a completa tradução de um *dataset* para outra língua implica em perda de precisão quando aplicada no contexto do país. Porém, com a tradução aliada a técnica do *data augmentation*, foi possível

[20]Teste de validação: https://github.com/mauriciosena/tccdataaugmentation/tree/main/Tests

utilizar a rede neural para analisar notícias verdadeiras e falsas na língua brasileira, apresentando um percentual bastante significativo.

Por outro lado, é compreensível notar que houve uma queda considerável de detecção de notícias falsas na língua brasileira, as quais a rede neural nunca teve contato em relação às notícias do conjunto de dados. Isso ocorre porque todo o treinamento foi feito com base em um *dataset* traduzido, onde as frases e contextos, por mais que fossem traduzidas de maneira eficiente, geralmente não correspondiam a reais situações do contexto brasileiro.

Contudo, este projeto de pesquisa mostrou ser possível, além de aprimorar funcionalidades existentes, também implementar novas funções em um projeto já desenvolvido, apresentando um significativo aumento na margem de precisão. Todo o projeto está disponível no Github[21].

Referências

Almeida 2020 Almeida, T. F. de. A pandemia de covid-19: Reflexos na garantia do direito à educação. *Pensar Acadêmico*, v. 18, n. 5, p. 881–894, 2020.

CanalTech 2021 CanalTech. *"Facebook Messenger ganha novas ferramentas para combater assédio e fake news"*. 2021. ⟨https://tinyurl.com/ferramentatwitterfacebook⟩. Accessed: 2021-05-23.

Chequeado 2021 Chequeado. *LATAM Coronavírus*. 2021. ⟨https://chequeado.com/latamcoronavirusportugues/⟩. Accessed: 2021-05-22.

Collins 2020 Collins. *Collins Dictionary 2020*. 2020. ⟨https://www.collinsdictionary.com/woty⟩. Accessed: 2021-05-15.

Devlin et al. 2018 Devlin, J. et al. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. Available on: ⟨http://arxiv.org/abs/1810.04805⟩.

Dias 2019 Dias, C. R. M. Towards fake news detection in portuguese: New dataset and a claim-based approach for automated detection. 2019.

Ding et al. 2020 Ding, K. et al. Challenges in combating covid-19 infodemic–data, tools, and ethics. *arXiv preprint arXiv:2005.13691*, 2020.

Folha 2021 Folha. *Lupa, a primeira agência fact-checking do Brasil*. 2021. ⟨https://piaui.folha.uol.com.br/lupa/?utm_source=Search&utm_medium=cpc&utm_campaign=lupa5anos⟩. Accessed: 2021-05-16.

Google 2021 Google. *Google Research*. 2021. Disponível em: ⟨https://research.google/teams/language/⟩.

IBM 2021 IBM. *Dataset*. 2021. Disponível em: ⟨https://www.ibm.com/docs/en/zos-basic-skills?topic=more-what-is-data-set⟩.

Kalsnes 2018 Kalsnes, B. Fake news. In: *Oxford Research Encyclopedia of Communication*. 2018.

Madani, Erritali and Bouikhalene 2021 Madani, Y.; Erritali, M.; Bouikhalene, B. Using artificial intelligence techniques for detecting covid-19 epidemic fake news in moroccan tweets. *Results in Physics*, Elsevier, p. 104266, 2021.

Mookdarsanit and Mookdarsanit 2021 Mookdarsanit, P.; Mookdarsanit, L. The covid-19 fake news detection in thai social texts. *Bulletin of Electrical Engineering and Informatics*, v. 10, n. 2, p. 988–998, 2021.

Neto et al. 2020 Neto, M. et al. Fake news no cenário da pandemia de covid-19. *Cogitare Enfermagem*, v. 25, 2020.

NLP 2021 NLP. *NLP*. 2021. Disponível em: ⟨https://www.nlp.com/what-is-nlp/⟩.

NLTK 2021 NLTK. *Tokenize*. 2021. Disponível em: ⟨http://nltk.sourceforge.net/doc/pt-br/tokenize.html⟩.

Paka et al. 2021 Paka, W. S. et al. Cross-sean: A cross-stitch semi-supervised neural attention model for covid-19 fake news detection. *Applied Soft Computing*, Elsevier, p. 107393, 2021.

Patwa et al. 2020 Patwa, P. et al. Fighting an infodemic: Covid-19 fake news dataset. *arXiv preprint arXiv:2011.03327*, 2020.

Pereira 2021 Pereira, M. R. A desinformação como estratégia política: uma análise dos tweets de ataque à imprensa postados por jair messias bolsonaro no ano de 2019. *Aquila*, n. 24, p. 97–110, 2021.

PyPi 2021 PyPi. *PyDictionary*. 2021. Disponível em: ⟨https://pypi.org/project/PyDictionary/⟩.

PyPi 2021 PyPi. *SpaCy*. 2021. Disponível em: ⟨https://pypi.org/project/SpaCy/⟩.

PYTHON 2021 PYTHON. *PYTHON*. 2021. Available on: ⟨www.python.org/about⟩.

Santos 2021 Santos, R. *Fake.Br Corpus*. 2021. ⟨https://github.com/roneysco/Fake.br-Corpus⟩. Accessed: 2021-05-19.

[21] Github do projeto: https://github.com/mauriciosena/

Shushkevich and Cardiff 2021 Shushkevich, E.; Cardiff, J. Tudublin team at constraint@ aaai2021–covid19 fake news detection. *arXiv preprint arXiv:2101.05701*, 2021.

Singh 2020 Singh, L. Fake news detection: a comparison between available deep learning techniques in vector space. In: IEEE. *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*. 2020. p. 1–4.

singularity014 2021 singularity014. *Singularity014 - Autor do Projeto Base*. 2021. Disponível em: ⟨https://github.com/singularity014/BERT_FakeNews_Detection_Challenge⟩.

Stanford 2021 Stanford. *História da Rede Neural*. 2021. Disponível em: ⟨https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html⟩.

TensorFlow 2021 TensorFlow. *TensorFlow*. 2021. Disponível em: ⟨https://www.tensorflow.org/tutorials/images/data_augmentation⟩.

UFC 2021 UFC, I. *Pythonx na área de Data Science*. 2021. Disponível em: ⟨https://tinyurl.com/linguagemdatascience⟩.