UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"

INSTITUTO DE BIOCIÊNCIAS, LETRAS E CIÊNCIAS EXATAS

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DETECÇÃO DE FAKE NEWS UTILIZANDO APRENDIZADO DE MÁQUINA

BAURU
2023

Gabriel Lino Garcia

Detecção de Fake News utilizando Aprendizado de Máquina

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de Bauru, como requisito para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. João Paulo Papa

G216d

Garcia, Gabriel Lino
Detecção de Fake News Utilizando Aprendizado de Máquina / Gabriel Lino Garcia. -- Bauru, 2023
79 p.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Faculdade de Ciências, Bauru
Orientador: João Paulo Papa

1. Processamento de Linguagem Natural. 2. Fake News. 3. Inteligência Artificial. 4. Português Brasileiro. 5. Novos Corpus. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Ciências, Bauru. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Bauru

ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE GABRIEL LINO GARCIA, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO, DA FACULDADE DE CIÊNCIAS - CÂMPUS DE BAURU.

Aos 04 dias do mês de janeiro do ano de 2023, às 14:00 horas, por meio de Videoconferência, realizou-se a defesa de DISSERTAÇÃO DE MESTRADO de GABRIEL LINO GARCIA, intitulada **Detecção de Fake News Utilizando Aprendizado de Máquina**. A Comissão Examinadora foi constituida pelos seguintes membros: Prof. Dr. JOAO PAULO PAPA (Orientador(a) - Participação Virtual) do(a) Departamento de Computacao / Faculdade de Ciencias Unesp Bauru, Prof. Dr. DIEGO FURTADO SILVA (Participação Virtual) do(a) Departamento de Computação / Universidade Federal de São Carlos, Prof. Dr. KELTON AUGUSTO PONTARA DA COSTA (Participação Virtual) do(a) FC / UNESP/Bauru (SP). Após a exposição pelo mestrando e arguição pelos membros da Comissão Examinadora que participaram do ato, de forma presencial e/ou virtual, o discente recebeu o conceito final:_ _ APROVADO _ _ _ . Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelo(a) Presidente(a) da Comissão Examinadora.

João Paulo Papa

Prof. Dr. JOAO PAULO PAPA

Dedico este trabalho à minha mãe, mulher guerreira que me ensinou sempre a batalhar e nunca desistir dos meus sonhos. Mesmo não estando presente fisicamente, sei que ilumina meus passos e me protege ao longo da vida, espero, um dia, sentir seu abraço novamente.
Obrigado por tudo, Gédera Delfino Lino (in memoriam).

Agradecimentos

Primeiramente, agradeço por todos os obstáculos que Deus coloca em meu caminho, pois quando chego ao topo da montanha, reconheço na paisagem o que ele queria me ensinar. Sou grato aos meus pais por sempre me incentivarem e acreditarem que eu seria capaz de superar os obstáculos que a vida me apresentou. Em especial, a minha MÃE, por todo ensinamento, por cada momento, por ser a melhor mãe do mundo, saiba que sou eternamente grato pela oportunidade de ser seu filho! **Essa é para você, MÃE!**

Agradeço meu orientador, Prof. João Paulo Papa por todos os conselhos, pela ajuda e pela paciência com a qual guiaram o meu aprendizado, serei eternamente grato.

A minha futura esposa e meus amigos, que sempre estiveram ao meu lado, por todo amor e amizade incondicional, pelo apoio demonstrado ao longo de todo o período de tempo em que me dediquei a este trabalho.

Por fim, agradeço a Deus pelo que conquistei até agora, mas peço a Ele para me dar sabedoria para conquistar muito mais.

Resumo

Nos últimos anos, as mídias sociais, redes sociais e aplicativos de mensagens se tornaram os principais canais para as pessoas acessarem e consumirem notícias, devido à rapidez e facilidade de divulgação de informações. No entanto, essas propriedades tornaram esses canais um foco de disseminação de notícias falsas, trazendo impactos negativos para os indivíduos e para a sociedade. Para resolver este problema, nos últimos anos, a utilização de inteligência artificial vem ganhando espaço e obtendo ótimos resultados na classificação de textos. O presente trabalho aborda o assunto de classificação de notícias falsas na língua portuguesa, possuindo o objetivo principal de estudar e avaliar procedimentos de processamento de linguagem natural, juntamente com artifícios de aprendizado de máquina, além de elaborar a criação de novos corpus de notícias falsas e verdadeiras para fomentar pesquisas de detecção automática de notícias falsas em português.

Palavras-chave: Informação, Notícias Falsas, Processamento de Linguagem Natural, Aprendizado de Máquina, Novos Corpus.

Abstract

Social media, social networks and messaging applications have become the main channels for people to access and consume news, due to the speed and ease of disseminating information. However, these properties made these channels a focus for disseminating false news, bringing negative impacts to individuals and society. To solve this problem, in recent years, the use of artificial intelligence has been gaining ground and obtaining excellent results in text classification. The present work addresses the subject of false news classification in the Portuguese language, with the main objective of studying and evaluating natural language processing procedures, along with machine learning artifices, elaborating the creation of new corpus of true and false news to foster research on automatic false news detection in Portuguese.

Key-Words: Information, Fake News, Natural Language Processing, Machine Learning, New datasets.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AM	Aprendizado de Máquina
BERT	*Bidirectional Encoder Representations from Transformers*
BoW	*Bag-of-Words*
CNN	*Convolutional Neural Network*
DCDistance	*Document-Class Distance*
EUA	Estados Unidos da América
GRU	*Gated Recurrent Unit*
IFT	*Image Foresting Transform*
LSTM	*Long Short-Term Memory*
MLP	*Multi Layer Perceptrons*
NFM	*Non-negative Matrix Factorization*
OCC	*One-Class Classification*
OPF	*Optimum-Path Forest*
PCA	*Principal Component Analysis*
PLN	Processamento de Linguagem Natural
POS	*Part-of-Speech*
SP	São Paulo
SVM	*Support Vector Machine*
TF-IDF	*Term Frequency - Inverse Document Frequency*
TV	Televisão

Sumário

1 Introdução

Diante do advento das mídias sociais, a facilidade de criação e compartilhamento de informações tornaram-se muito práticas e rápidas, alcançando milhões de usuários ao redor do mundo. Com esta facilidade para atingir uma grande quantidade de pessoas, outros meios de comunicação como canais de televisão (TV), jornais e rádios tornaram-se menos eficientes na troca de informação e no alcance de pessoas ao redor do mundo. Entretanto, várias destas informações podem ser disseminadas com seus dados distorcidos, contendo opiniões forjadas sobre produtos e serviços, manchetes sensacionalistas, textos satíricos, com conteúdo enganoso e, muitas vezes, sem possuir uma base teórica sobre o assunto tratado. Estas notícias são denominadas *fake news* (notícias falsas) e causam grandes transtornos para a sociedade atual, pois, afetam a formação de opinião, a tomada de decisões e os padrões de votação das pessoas em eleições nacionais, estaduais e municipais (ALLCOTT; GENTZKOW, 2017).

A maioria das notícias falsas é inicialmente distribuída por canais de mídia social como *Facebook* e *Twitter* e, mais tarde, encontra seu caminho para as plataformas de mídia convencional, como notícias tradicionais de rádio e televisão (TRAYLOR et al., 2019). O primeiro grande impacto mundial com as notícias falsas deste século ocorreram em 2016, durante as eleições norte americanas que influenciaram na escolha dos candidatos a presidência dos Estados Unidos da América (EUA). Vale ressaltar, que as *fake news* foram popularizadas neste século, entretanto há relatos que elas são circuladas na comunidade desde o século VI (ALLCOTT; GENTZKOW, 2017).

Embora esse fenômeno não seja novo, as notícias falsas ganharam uma grande popularidade conforme o acesso a *internet* e a polarização de discussões entre internautas sobre assuntos públicos, como por exemplo, as eleições norte americanas, tornando-se um ambiente favorável para a disseminação de conteúdos que visam beneficiar um lado destas discussões, onde o compartilhamento destas notícias podem ser realizadas por um humano ou por uma grande quantidade de *bots* (SHAO et al., 2017). As notícias podem conter não apenas o seu texto falsificado, mas em muitos casos, imagens falsas, legendas incorretas e manchetes enganosas são utilizadas para maximizar o impacto da notícia criada. Após as eleições norte americanas, as mídias sociais apenas alavancaram todo o conteúdo gerado, por meio de postagens e compartilhamentos, assim, as notícias falsas começaram a ser classificadas de acordo com suas características de escrita ou conteúdo, sendo elas: sátira ou paródia, falsa conexão, conteúdo enganoso, falso contexto, conteúdo impostor, conteúdo manipulado, conteúdo fabricado (WARDLE, 2017).

A detecção de conteúdo enganoso na *internet* tornou-se preocupante, pois segundo Bond e DeoPaulo (2008), a capacidade humana de diferenciar o que é verídifico ou falso

fica na média de 54%, o que é muito próximo de uma escolha aleatória. Diante disso, em um esforço para combater a crescente desinformação, vários sites foram elaborados para checar essas informações, desempenhando um papel importante no combate as notícias falsas (MIHALCEA; STRAPPARAVA, 2009), tornando a identificação correta de notícias falsas um problema desafiador.

Várias pesquisas relacionadas a esse tema também foram desenvolvidas com o intuito de identificar corretamente se uma notícia é verdadeira ou falsa. Entretanto, de acordo com Wang (2017), a pequena coleção de dados contendo notícias falsas e sua rotulação manual, são os primeiros problemas encontrados para a realização de pesquisas. Um segundo problema, está relacionado a escrita das notícias, pois, a maioria dos escritores tendem a utilizar uma linguagem estratégica para evitar ser capturado e exposto (YANG et al., 2018). Porém, levando em consideração pesquisas realizadas no idioma português, o principal desafio é a pequena quantidade de dados públicos em relação a outros idiomas, assim, dificultando a identificação e classificação de notícias.

A fim de que o leitor se familiarize com o cenário deste trabalho, os objetivos e a estrutura desta monografia são elencados a seguir.

1.1 Objetivos

O objetivo geral e os objetivos específicos deste trabalho são apresentados a seguir.

1.1.1 Objetivo Geral

Com base em todo o contexto discutido anteriormente, o trabalho visa mostrar modelos preditivos em conjunto com algoritmos de aprendizado de máquina (AM) e técnicas de processamento de linguagem natural (PLN), para a classificação de notícias falsas. Propor a criação, desenvolvimento e experimentação de novas abordagens, além de avaliar e comparar a eficácia destas abordagens por meio dos corpus brasileiros: Fake.Br Corpus (MONTEIRO et al., 2018), FakeRecogna (GARCIA; AFONSO; PAPA, 2022) e FakeRecogna Anomaly.

1.1.2 Objetivos Específicos

Nesta seção serão apresentados os objetivos específicos do presente trabalho.

- Identificar as características de *fake news* circuladas na *internet*;
- Analisar as principais abordagens utilizadas no cenário de classificação de *fake news*;
- Encontrar e analisar aplicações com base de dados em português;

- Aplicar abordagens utilizadas com outros métodos de processamento de linguagem natural;
- Comparar o desempenho de algoritmos por meio da acurácia utilizando a base de dados Fake.Br Corpus;
- Desenvolver uma nova base de dados balanceada para detecção de fake news em português;
- Testar e validar a base de dados a partir de algoritmos de classificação;
- Desenvolver uma nova base de dados desbalanceada para detecção de fake news em português para contextualizar com o mundo real;
- Testar e validar o corpus com experimentos envolvendo detecção de anomalias.

1.2 Contribuições do Trabalho

As principais contribuições do trabalho são:

- Criação de uma base de dados atualizada para identificação de fake news[1];
- Identificação de fake news como sendo um problema de detecção de anomalias[2].

1.3 Hipóteses de Pesquisa

As contribuições do trabalho já mencionadas foram direcionadas pelas seguintes hipóteses de pesquisa:

1. As bases de dados para detecção de fake news disponíveis na literatura estavam desatualizadas e poderiam não corresponder à realidade atual;
2. Fake news ocorrem com muito menos frequência do que as notícias reais, gerando bases de dados que são desbalanceadas. Muito embora possamos endereçar este problema por meio de técnicas superamostragem ou subamostragem, ainda assim não corresponderíamos a realidade. Desta forma, acreditamos que modelar o problema de identificação de fake news como sendo uma tarefa de detecção de anomalias resulte em uma abordagem mais eficaz.

[1] https://link.springer.com/chapter/10.1007/978-3-030-98305-5_6

[2] Garcia, Gabriel L and Afonso, Luis and Passos, Leandro and Jodas, Danilo and Costa, Kelton and Papa, João P, "FakeRecogna Anomaly: Fake News Detection in a New Brazilian Corpus.", in International Conference on Computer Vision Theory and Application, No prelo.

1.4 Estrutura da Monografia

O restante dessa dissertação está organizada da seguinte forma:

- Capítulo 2: Apresenta conceitos e definições sobre as *fake news*, com foco nas características e classificações desenvolvidas;
- Capítulo 3: Apresenta os conceitos, definições e técnicas que envolvem o processamento de linguagem natural;
- Capítulo 4: Apresenta conceitos, definições e técnicas que contextualizam o aprendizado de máquina;
- Capítulo 5: Apresenta as bases de dados envolvidas no projeto, o processo de criação, a modelagem dos dados, além de características de cada uma delas;
- Capítulo 6: Apresenta os resultados obtidos por meio de algoritmos com técnicas já utilizadas e com novas abordagens, a comparação com outros trabalhos relacionados a base de dados Fake.Br Corpus, além da apresentação dos experimentos envolvendo os corpus FakeRecogna e FakeRecogna Anomaly;
- Capítulo 7: Apresenta a consideração final desta dissertação.

2 Fake News

No presente trabalho, as notícias falsas serão abordados em conjunto com suas nomenclaturas, uma vez que sua definição é peça importante para que as categorias possam enfatizar ainda mais como elas são empenhadas e disseminadas conforme sua classe. Primeiramente, será apresentado um breve resumo sobre as definições e a história por trás das *fake news*. No segundo momento, são descritas suas classificações, juntamente com exemplos para maior compreensão do assunto.

2.1 Definição

De acordo com Muigai (2019), *fake news* é qualquer informação falsa que é deliberadamente destinada a ser total ou amplamente falsa ou enganosa, espalhada através da mídia social *online*, mas ocasionalmente encontrando seu caminho para a mídia impressa e de transmissão de notícias tradicionais. Além disso, o termo pode ser utilizado para lançar dúvidas sobre notícias legítimas de um ponto de vista político oposto, uma tática conhecida como imprensa mentirosa. A informação falsa é muitas vezes sensacionalista, desonesta e totalmente fabricada, onde muita das vezes, é repercutida nas redes sociais, pois, as mídias sociais estão se tornando as principais fontes de notícias para um número crescente de indivíduos ao redor do mundo. Assim, a desinformação parece ter encontrado um novo canal muito eficaz para sua propagação.

Segundo Muigai (2019), uma definição mais ampla de notícia falsa pode ser informação falsa ou enganosa publicada como notícia autêntica, geralmente entendida como deliberada, embora possivelmente acidental. Essas histórias não são encontradas apenas na política, esporte, medicina, celebridades, etc, mas em todas as áreas de interesse humano.

As notícias falsas também foram definidas por Allcott e Gentzkow (2017) como artigos que são intencionalmente e verificadamente falsos e podem enganar os leitores, no qual, duas motivações principais estão por trás da produção de notícias falsas: financeiras e ideológicas. Pelo lado financeiro, histórias ultrajantes e falsas que se tornam virais fornecem aos produtores de conteúdo cliques que podem ser convertidos em receita de publicidade. E, por outro lado, outros fornecedores de notícias falsas produzem notícias falsas para promover ideias ou pessoas específicas que eles favorecem, muitas vezes desacreditando outras pessoas, serviços e, diversos assuntos tratados.

Para Tandoc, Lim e Ling (2017), as *fake news* são notícias com conteúdo falso e sem embasamento, criadas com o intuito de enganar e manipular o leitor. Esta categorização de *fake news* é relativamente nova, mas a desinformação afetando a sociedade é algo que acontece há

muito tempo. Um exemplo clássico de desinformação é datada em 1938, quando a transmissão de uma adaptação para o rádio do drama de H. G. Well, A Guerra dos Mundos, amedrontou cerca de um milhão de pessoas nos EUA (CANTRIL, 2005). Ao adotar um formato de notícias de rádio por meio da tecnologia relativamente nova do rádio, com atores desempenhando os papéis de repórteres, residentes, especialistas e funcionários do governo, o diretor de drama de rádio Orson Welles encontrou uma maneira inteligente de narrar a história da invasão marciana. Embora sua intenção fosse entreter os ouvintes, a adaptação para o rádio assumiu a forma de uma reportagem, ao vivo, em um período que o rádio era a principal fonte de informação americana. Embora a intenção de Wells fosse produzir uma peça de drama para o rádio, os ouvintes interpretaram-na como notícias reais (TANDOC; LIM; LING, 2017). Um outro exemplo, foram as principais armas nazistas, antes e durante a segunda guerra mundial, desenvolvida pelo Ministro de Propaganda, Joseph Goebbels, que disseminava o nazismo e a intensa propagação de notícias falsas que levavam a população a acreditar que o partido autoritário era o melhor caminho a se seguir.

No Brasil, mais especificamente na cidade de Guarujá, São Paulo (SP), houve um caso em 2014 que ficou muito famoso na época. Uma página de portal de notícias da cidade, no *Facebook*, realizou uma publicação afirmando que uma cidadã sequestrava crianças e praticava magia negra. Logo, com todo o boato gerado, criou-se um retrato falado da mulher que, na verdade, era um retrato que havia sido feito por agentes da Polícia Civil do Rio de Janeiro por conta de um crime que havia acontecido dois anos antes desta publicação. Porém, este retrato foi rapidamente espalhado pelas redes, juntamente com histórias e relatos falsos. A mulher, Fabiane Maria de Jesus, de 33 anos, ofereceu uma fruta para uma criança e, a mãe, mobilizada pelas publicações realizadas no *Facebook*, suspeitou da atitude da mulher e reagiu aos gritos, algo que mobilizou os populares a sua volta. O final desta triste história foi o linchamento e morte, da cidadã, pelos populares (RIBEIRO, 2014).

Em resumo, as notícias falsas abrangem tudo o que é falso, errôneo, duvidoso e contos da mídia que circulam amplamente e servem para corromper ou distorcer o popular. Sendo assim, torna-se evidente que as notícias falsas, são um grande entrave do bem-estar da humanidade. Diante desse contexto, é evidente a atenção à passividade das redes sociais e à alienação da sociedade (MUIGAI, 2019).

A seguir, serão dispostos os tipos de notícias falsas, que de acordo com Wardle (2017), são classificadas como: sátira ou paródia, falsa conexão, conteúdo enganoso, falso contexto, conteúdo impostor, conteúdo manipulado e conteúdo fabricado. O objetivo é determinar as conceituações para oferecer um panorama de como o fenômeno vem sendo percebido em diferentes pesquisas, além de demonstrar que possuem propriedades e atributos diferentes.

2.2 Sátira ou Paródia

Segundo Tandoc, Lim e Ling (2017), é a categoria mais comum, referindo-se a programas de notícias simuladas, que utilizam-se do humor ou exagero para apresentar ao público atualizações de notícias. Esses programas são apresentados por comediantes ou artistas e são comuns na mídia televisiva mundial. Esse tipo de linguagem não fica apenas no canal televisivo de comunicação, podendo ser encontrado em diversos canais: internet, mídias sociais, quadrinhos e desenhos animados (BURFOOT; BALDWIN, 2009). Mesmo se baseando em notícias reais, esses programas normalmente deixam perceptível que a produção de conteúdo é de caráter humorístico. Em resumo, segundo Wardle (2017), a linguagem sátira ou paródia não possui intenção de causar danos, mas tem potencial para enganar.

No Brasil, o exemplo mais famoso é o site *Sensacionalista* que com tamanha repercussão, acabou ganhando um programa televiso no canal *MultiShow* (CARVALHO, 2009). Na paródia, o formato jornalístico é mais acentuado que na sátira, o que faz com que seja mais facilmente confundida com acontecimentos reais, o que pode frustar a intenção dos seus formuladores. Segundo Thu e New (2017), a manipulação da sátira é atualmente uma das tarefas desafiadoras na linguística computacional, processamento de linguagem natural e análise de sentimentos de mídias sociais.

2.3 Falsa Conexão

Segundo Wardle (2017), falsa conexão é um tipo de notícia que apresenta divergências entre a chamada da notícia e o conteúdo apresentado. Esse tipo de notícia é usado para atrair a atenção do leitor com manchetes, imagens ou legendas chamativas, o conteúdo atrativo da chamada da notícia não condiz com o conteúdo apresentado. Em suma, são quando manchetes, imagens ou legendas não condizem com o conteúdo da notícia.

Para exemplificar este tipo de notícia falsa, Lopes (2020), a partir do sítio eletrônico criado em 2002, que procura desmascarar notícias falsas, denominado E-farsas, demonstra que houve um *post* circulado nas mídias sociais a respeito de que "*o atual presidente do Brasil, Jair Messias Bolsonaro, havia decretado estado de emergência* 18 *dias antes do carnaval*". Entretanto, o que realmente ocorreu foi que o presidente decretou estado de emergência em saúde pública no dia 03 de fevereiro de 2020, no entanto, tratava-se de uma portaria que permitia à Secretaria de Vigilância em Saúde a contratação temporária de profissionais de saúde, além da aquisição de equipamentos e a contratação de serviços. Não há nenhuma menção, nessa portaria, a respeito de quarentena ou de isolamento social.

2.4 Conteúdo Enganoso

De acordo com Wardle (2017), conteúdo enganoso é um tipo de notícia que utiliza informações falsas para enquadrar um problema ou um indivíduo. Esse tipo de notícia é usado com intuito de denegrir a imagem de um conteúdo ou pessoa atribuindo dados ofensivos à sua fama, honra ou dignidade.

Segundo Gomes (2020), montagens que circulam nas redes sociais associadas ao ex-Ministro da Saúde, Nelson Teich, onde repercutem uma visita ao estado do Amazonas e, a partir da sua visita, mortes por Covid-19 diminuíram de forma expressiva, porém, um levantamento realizado pelo portal G1 e Portal (2020), junto à Secretaria da Saúde do Amazonas apontaram que a situação foi ao contrário, após a passagem de Teich pelo Amazonas, houve um recorde em mortes em apenas 24 horas.

2.5 Falso Contexto

Em conformidade com Wardle (2017) falso contexto é um tipo de notícia onde o conteúdo é verdadeiro, mas é compartilhado com um contexto falso, ou seja, o usuário utiliza a notícia com uma abordagem de denegrir ou difamar o assunto. Por exemplo, um usuário cria uma legenda falsa do conteúdo e compartilha com intuito de difamar o conteúdo verdadeiro da notícia, manipulando o foco da notícia, assim, outros usuários podem acreditar apenas na legenda da notícia, sem conferir-la na íntegra.

Conforme Gomes (2020), um exemplo pode ser visualizado com a seguinte manchete: "*Polícia Federal desenterra 26 caixões cheios de pedras no Amazonas*". Porém as imagens demontradas na notícia referem-se há um fato que realmente aconteceu, porém em 2007, na cidade de São Carlos (SP), no contexto de uma investigação que envolvia fraudes contra uma seguradora.

2.6 Conteúdo impostor

De acordo com Wardle (2017), conteúdo impostor é um tipo de notícia que usa o nome de uma pessoa ou marca, mas com afirmações irreais. Esse tipo de notícia tenta enganar os leitores usando nomes de pessoas ou marcas que são normalmente confiáveis, já que uma das formas usadas para verificar a credibilidade da notícias é verificar a fonte ou autor.

Nesta categoria, pode-se exemplificar com uma notícia que circulou pelas mídias sociais, onde referia-se a uma notícia publicada pela Folha de São Paulo. A notícia demonstrava certo sarcasmo e ironia possuindo a seguinte manchete: "*Segundo especialistas da Folha, com a demissão do ministro Moro, ninguém mais morrerá de Covid-19, apenas de desgoto*". Fica claro que existe uma sátira ou paródia nesta notícia, entretanto quando analisado a fundo,

descobriu-se um perfil falso com temas humorísticos que compartilhava informações possuindo a mesma logo e nome de uma das grandes referencias jornalísticas brasileiras (GOMES, 2020).

2.7 Conteúdo Manipulado

Segundo Wardle (2017), o conteúdo verdadeiro é manipulado para enganar o público, ou seja, o escritor da notícia distorce as informações criando novas afirmações falsas para enganar e manipular os leitores que irão visualizar a notícia.

Conforme o portal de notícias G1 e Portal (2020), a partir do serviço de monitoramento e checagem de conteúdos duvidosos denominado "FATO OU FAKE", um claro exemplo nessa categoria fica por conta de uma notícia manipulada que discorria com a seguinte manchete: "*Traficante armado com fuzil morre por covid-19 após confronto com a polícia, entretanto a foto desta notícia envolvia uma reportagem realizada pela GloboNews, onde o repórter falava sobre as fortes chuvas no Rio de Janeiro que deixaram a cidade em estágio de atenção e duas pessoas mortas*".

2.8 Conteúdo Fabricado

De acordo com Wardle (2017), nesta categoria, a informação da notícia é totalmente falsa, construída para causar algum mal e espalhar um boato, ou seja, este tipo de notícia não apresenta nenhuma informação verídica, sendo utilizada com o único objetivo de desinformar e manipular os leitores.

Segundo Matsuki (2020), do portal BOARTOS.ORG, uma notícia que foi publicada no *Youtube* e teve repercussão, possuía o seguinte título: "*Água tônica tem quinino, base da cloroquina. Isso a Globo não te conta!!!*". No entanto, a água tônica é feita de um extrato da casca da árvore de cinchana (hidrocloreto de quinina) que garante o gosto amargo e característico do produto. A quinina não é a mesma molécula do hidroxicloroquina e cloroquina.

3 Processamento de Linguagem Natural

3.1 Definição

A linguagem é um dos aspectos fundamentais do comportamento humano, pois permite a interação entre os indivíduos e a perpetuação dos conhecimentos. Segundo Bird, Klein e Loper (2009), o termo "linguagem natural", significa uma linguagem que é usada para comunicação diária pelos humanos, por exemplo, idiomas como inglês, português, espanhol, mandarim, etc. Em contraste com as linguagens artificiais, como as linguagens de programação e notações matemáticas, as linguagens naturais evoluem conforme passam de geração para geração e são difíceis de definir com regras explícitas. Assim, conforme Indurkhya e Damerau (2010), o processamento de linguagem natural (PLN) é uma área da computação que tem como objetivo extrair representações e significados mais completos de textos escritos em linguagem natural.

De acordo com Gasperin e Lima (2016), o processamento automático da linguagem natural, visa aproximar o computador da realidade do homem, através do desenvolvimento de ferramentas que possibilitem uma comunicação mais natural entre homem e máquina, além de ferramentas para a extração de informações de grandes bases textuais, tradução automática de textos, etc. A grosso modo, o PLN pode ser definido como uma forma de descobrir quem fez o quê, a quem, quando, onde, como e porquê. Para o desenvolvimento de ferramentas utilizando PLN, é considerado uma estrutura hierárquica da linguagem, possuindo várias palavras que formam uma frase, várias frases que formam uma sentença e, sentenças que transmitem ideias (BARBOSA et al., 2017).

Segundo Bird, Klein e Loper (2009), as tecnologias baseadas em PLN se tornaram cada vez mais difundidas ao longo do tempo. Por exemplo, telefones e computadores portáteis suportam texto preditivo e reconhecimento de escrita; motores de pesquisa na *web* fornecem acesso a informações encerradas em texto não estruturado; a tradução automática permite recuperar textos escritos em chinês e lê-los em espanhol. Além destes exemplos, os sistemas de PLN podem ser utilizados em diversas outras gamas, como na conversão de fala para texto, análise de sentimentos/mineração de opiniões, sumarização automática de texto, extração de tópicos, mineração de dados, *stemming* entre outras (LIU; ZHANG, 2012). Ao fornecer interfaces homem-máquina mais naturais e um acesso mais sofisticado às informações armazenadas, o processamento de linguagem passou a desempenhar um papel central na sociedade da informação multilíngue.

Segundo Aho et al. (2006), a linguagem humana é raramente precisa, pois, diariamente as palavras e orações são abreviadas, distorcidas ou utilizadas com diversos significados e situações diferentes. Assim, entender a linguagem humana é entender não apenas as palavras,

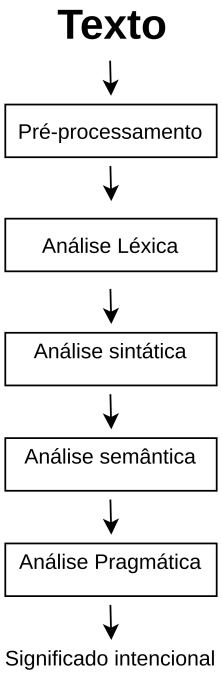
mas os conceitos e como eles estão ligados entre si para criarem significados. Apesar de a linguagem ser uma das coisas fáceis para os humanos aprenderem, a ambiguidade da linguagem é o que torna o processamento de linguagem natural um problema díficil para computadores.

Conforme Pinto (2015), a ambiguidade existe quando não é possível atribuir apenas um significado a uma determinada expressão. Normalmente, uma pessoa consegue facilmente interpretar corretamente a ambiguidade na comunicação devido à sua experiência, às regras sintáticas, ao contexto ou à cultura. No entanto, um computador não tem a capacidade que lhe permite discernir todos esses pontos. A ambiguidade pode ocorrer nos diferentes níveis de conhecimento: fonético, morfológico, sintático, semântico, pragmático e de discurso.

O processamento de linguagem natural é importante por razões científicas, econômicas, sociais e culturais. As técnicas de PLN estão experimentando um rápido crescimento à medida que suas teorias e métodos são implantados em uma variedade de novas tecnologias de linguagem. Na indústria, isso inclui pessoas na interação homem-computador, análise de informações de negócios e desenvolvimento de software da web. Dentro da academia, inclui pessoas em áreas de computação em ciências humanas e linguística de corpus até ciência da computação e inteligência artificial (BIRD; KLEIN; LOPER, 2009). Por esses motivos, é importante conhecer e desenvolver pesquisas para minimizar tarefas, evoluir desempenhos e transformar sistemas com o uso destas técnicas, trazendo consigo um maior benefício para a sociedade atual.

Conforme Jurafsky e Martin (2019), pode-se definir o processamento de linguagem natural como um campo da inteligência artificial, cujo objetivo é compreender, analisar e gerar a língua natural para os humanos, de forma que, eventualmente, seja possível dirigir-se a um computador da mesma forma que se comunica com uma pessoa. Segundo Barbosa et al. (2017), o processamento de linguagem natural é decomposto em várias etapas, refletindo as distinções linguísticas teóricas traçadas entre sintaxe, semântica e pragmática. Essa separação serve como base para modelos arquitetônicos que tornam a tarefa de análise de linguagem natural mais gerenciável do ponto de vista de engenharia de software (DALE, 2010). Esta decomposição em mais estágios é útil ao se levar em conta o atual estado da arte em combinação com os dados de linguagem reais, como descritos na Figura 1.

Figura 1 – Estágios de análise em PLN.



Fonte: Adaptado de (DALE, 2010).

Assim, conforme Machado et al. (2010), PLN visa promover um nível mais alto de compreensão da linguagem natural através do uso de recursos computacionais, com o emprego de técnicas para o rápido processamento de texto. O processamento de linguagem natural surgiu devido à necessidade de compreensão automática e comunicação em geral do ser humano com o computador. Trata-se de um mecanismo criado não somente para extrair as informações de textos, mas também para facilitar a entrada de dados nos sistemas e a estruturação desses dados (BULEGON; MORO, 2010).

Em suma, segundo Aranha (2007), PLN é o campo da Ciência da Computação e da Linguística que abrange um conjunto de métodos formais para analisar textos, gerar frases e realizar comunicação em um idioma humano através do uso de programas computacionais. A partir disso, a seguir, serão abordados mais detalhadamente cada uma das cinco fases de análise no PLN.

3.2 Pré-Processamento

Conforme Guimarães, Meireles e Almeida (2019), o objetivo do pré-processamento é extrair de textos uma representação estruturada e manipulável por algoritmos de classificação, para que assim, identifique o subconjunto mais significativo para a padronização da coleção de textos, ou seja, a seleção reduz o conjunto de palavras que representará o documento no processo de classificação. São exemplos das etapas de seleção a remoção de palavras

irrelevantes, como preposições ou artigos, a formatação dos textos e o cálculo de relevância dos termos para identificar os mais significativos.

De acordo com Junior (2013), o pré-processamento pode ser dividido, de maneira geral, em dois processos. O primeiro é a triagem documental, que é a conversão de um conjunto de arquivos digitais em um documento de texto bem formado. Assim, a conclusão desse estágio é um corpus bem definido e pronto para ser utilizado para análises mais profundas. O segundo processo, é a segmentação do texto, consistindo na conversão do corpus em unidades lexicais e sentenças, onde cada unidade lexical é denominada *token*, e esse processo denomina-se tokenização (CHAUDIRON, 2007).

Conforme Indurkhya e Damerau (2010), a tokenização também conhecida como segmentação de palavras, quebra a sequência de caracteres em um texto localizando o limite de cada palavra, ou seja, os pontos onde uma palavra termina e outra começa, assim, as palavras são frequentemente chamadas de *tokens*. A tokenização é bem estabelecida e bem entendida para linguagens artificias tais como linguagens de programação (AHO et al., 2006). No entanto, tais linguagens artificiais podem ser estritamente definidas para eliminar ambiguidades léxicas e estruturais. Assim, essa aptidão não existe na linguagem natural, em que os mesmos caracteres podem adequar-se para várias finalidades e que a sintaxe não é exclusivamente definida (BARBOSA et al., 2017).

Para a realização da tokenização, tem-se que diferenciar duas variantes no que tange a delimitação dos *tokens*, dependente da linguagem no qual o texto ou sentença tiver sido escrito. São elas: linguagem delimitadas por espaço e as linguagens não segmentadas. Nas linguagens delimitadas por espaço, como a grande maioria das linguagens ocidentais, os limites das palavras são geralmente definidos por espaços em branco ou alguma forma de pontuação. Entretanto, em linguagens não-segmentadas, tais como as orientais, as palavras são escritas sucessivamente sem nenhuma indicação dos limites correspondentes a uma palavra. De acordo com Aranha (2007), a tokenização é o primeiro estágio do pré-processamento de um texto. Nesta etapa, o texto representado por uma sequência de caracteres é agrupado em um primeiro nível, possuindo fronteiras delimitadas por caracteres primitivos como espaço em branco, vírgula, ponto, etc. Neste primeiro nível, cada grupo de caracteres estabelecidos são os *tokens* e, a sequência desses grupos, por sua vez, é chamada de *tokenstream*. Para uma melhor comprensão, observe o exemplo abaixo.

Frase: Meu irmão ganhou na loteria, ficou milionário.

Tokenização: [Meu] [irmão] [ganhou] [na] [loteria] [,] [ficou] [milionário] [.]

Segundo Aranha (2007), em português, o processo de tokenização é mais fácil de se realizar, pois consiste em separar as palavras sempre que houver um espaço entre elas e, quando houver sinais de pontuação, elas serão como tokens, assim sendo, a pontuação também tem significado. Entretanto, existem casos a serem considerados, tais como, problemas com

armazenamento de tokens desnecessários, por exemplo, símbolos e sinais (“(-):,/”), problemas de excesso de precisão, onde os verbos e nominalizações devem aparecer exatamente da forma digitada, problema relacionado a letras maiúsculas e minúsculas, entre outros problemas.

Conforme o trabalho de Junior (2013), a normalização do texto é exatamente o processo de certificar diferentes tokens para o seu lema original, com o objetivo de diminuir o espaço de armazenamento e acelerar o processamento. De acordo com Junior (2013), após o reconhecimento de palavras, passa-se ao reconhecimento das orações. Assim, precisa-se delimitar o final da oração, entretanto a utilização de ponto no decorrer de orações podem acontecer, seja na parte fracionária de um número, em uma abreviação ou quando uma frase com um único ponto final ao término pode possuir várias orações, separas por vírgulas ou semivírgulas, e cada uma dessas sentenças apresentar uma ideia diferente.

Neste contexto, um dos principais desafios de pré-processamento textual pode ser descrito pelo tipo do sistema de escrita na linguagem, ou seja, existem diversos tipos de escrita como o iconográfico, simbolizado pelos japoneses, nas quais um grande conjunto de símbolos individuais representam as palavras. Nas línguas silábicas, os símbolos individuais simulam sílabas, além da língua alfabética, como o português, que utilizam os símbolos para representação do som. A maioria das linguagens usa modelos silábicos ou alfabéticos, entretanto qualquer linguagem, em algum momento, pode utilizar os três sistemas. Por exemplo, o símbolo $\%$, na língua portuguesa, é muito utilizado em frases, porém é um iconográfico e certamente estará ligado com outro tipo de linguagem. Assim sendo, um problema que deve ser tratado na tokenização.

Outro problema é a questão da dependência, pois um sistema de PLN depende do conjunto de caracteres utilizado na codificação. Entretanto, o maior problema está propriamente na linguagem. Um sistema de PLN para língua portuguesa não obterá bons resultados se aplicado na língua inglesa ou alemã, por exemplo. Assim, fica claro que existe uma dependência do corpus. Um sistema de PLN treinado para documentos de certo formato não apresenta bons resultados em textos livres, com seus erros e quebras gramaticais.

3.3 Análise Léxica

A subseção anterior tratou sobre pré-processamento, ou seja, trabalhar com os dados para conseguir um melhor desempenho e melhor tratamento para futuros processos subsequentes. O próximo processo em questão é a análise em nível de palavras, mais conhecida como análise léxica ou morfológica.

Segundo Bulegon e Moro (2010), análise morfológica ou léxica, é responsável por definir artigos, substantivos, verbos e adjetivos, os armazenando em um tipo de dicionário. Depois de construído o dicionário, a análise sintática faz-se uso, procurando mostrar relacionamento entre as palavras e, num segundo momento, verificar sujeito, predicado, complementos nominais e

verbais, adjuntos e apostos. Na análise semântica, ocorre o encontro de termos ambíguos, de sufixos e afixos, ou seja, questões de significado associados aos morfemas componentes de uma palavra, o sentido real da frase ou palavra.

Na análise léxica, o objetivo é estudar a morfologia das unidades lexicais ou palavras, e recuperar informação que será útil em níveis mais profundos de análise. Segundo Dale (2010), a decomposição das palavras, assim como a detecção de regras de formação, permite a redução de espaço de armazenamento e aumenta a velocidade de processamento, considerando o armazenamento de cada unidade lexical encontrado em um repositório.

De acordo com Hippisley (2010), as palavras são atribuídas a tijolos para construção de textos em linguagem natural. Acrescenta que há normalmente duas abordagens em PLN para tratamento de uma unidade lexical. A primeira é tratar simplesmente como uma *string* de texto: uma cadeia de caracteres. A segunda é considerar a palavra como um objeto mais abstrato, que é um termo derivado de um lema canônico com um conjunto de regras de formação. Assim, segundo Barbosa et al. (2017), uma palavra pode ser pensada de duas maneiras: como uma sequência de caracteres no texto em execução, como por exemplo, o verbo ENTREGAR, ou como um objeto mais abstrato que é o termo principal para um conjunto de sequência de caracteres (palavras), sendo o verbo ENTREGAR o objeto abstrato, ou *lemma*, do conjunto entrega, entregador, entregando, entregue.

Conforme Erjavec e Dzeroski (2004), a linguagem possui uma grandeza de inflexões, com substantivos flexionando para gênero, número e uma configuração complexa de terminações e modificações do lema original. Por exemplo, o plural de certas palavras no português podem ser simples, adicionando apenas o caractere "s" ao final da palavra, como "letra" se torna "letras", entretanto, existem casos que podem ser irregulares, como na palavra "informação" que no plural se torna "informações", onde a adição do caractere "s" ao final há alteração de vogal no lema.

Assim, pode-se definir que a análise léxica possui dois lados: o mapeamento da palavra até o seu *lemma*, denominado *parsing side* e, o outro é o mapeamento do *lemma* para a palavra, chamado de geração morfológica. O *parsing side*, segundo Junior (2013), substitui as diversas formas de representação da palavra pela forma primitiva. Essa estratégia tem a vantagem de captar mais as intenções de busca do usuário, já que a forma da lematização é de fácil interpretação. Outro processo que ocorre é denominado stemização, uma operação que leva em consideração as flexões verbais, realizando truncamentos que tornam as palavras, na maioria das vezes de difícil compreensão. Segundo Barbosa et al. (2017), a stemização é chamado de radical da palavra. Por exemplo, o verbo ENTREGAR, com o processo de stemização resultaria em ENTREG, pois a partir do *stem* poderiam ser criadas outras palavras.

Vale ressaltar que na análise léxica, uma importante ferramenta é o etiquetador *part-of-speech (POS)*. Segundo Junior (2013), a abordagem consiste em partir de uma sentença completa, com seu conjunto de palavras, etiquetar cada unidade lexical encontrada com sua

respectiva categoria morfológica ou classe. Um etiquetador é responsável pelo processo de definição da classe gramatical das palavras, de acordo com as funções sintáticas. O processo de classificação das palavras em sua classe gramatical e rotulação são conhecidas como *part-of-speech tagging* (BARBOSA et al., 2017). De acordo com GÜNGÖR (2010), esse é um subprocesso ou processo complementar da análise morfológica, pois enquanto a análise léxica propriamente dita envolve encontrar a estrutura interna de uma palavra, seu lema, derivações e regras de formação, o etiquetador faz uma análise superficial, associando uma categorização à unidade lexical.

De acordo com Jurafsky e Martin (2019), existem muitas dificuldades no processo de etiquetagem POS. O primeiro deles é a ambiguidade, onde palavras iguais podem assumir diferentes funções morfológicas dentro da sentença. Complementando ainda mais as dificuldades encontradas na POS, conforme GÜNGÖR (2010), demonstra exemplos desse problema, além de acrescentar o problema do encontro de palavras desconhecidas.

A aplicação de etiquetagem POS pode ser visualizada no trabalho de Ribeiro, Oliveira e Trancoso (2003), onde foi proposto uma arquitetura Transformer baseada em POS, na qual, este modelo utiliza o mecanismo de auto-atenção para aprender a expressão característica do texto, além da incorporação do POS-Atenção que funciona na captação de informações sentimentais contidas na parte do discurso visando a classificação de sentimentos. Já Garcia e Gamalho (2010), construiram um POS tagger para o Português Contemporâneo e o Português Histórico, baseado em uma arquitetura de rede neural recorrente.

3.4 Análise Sintática

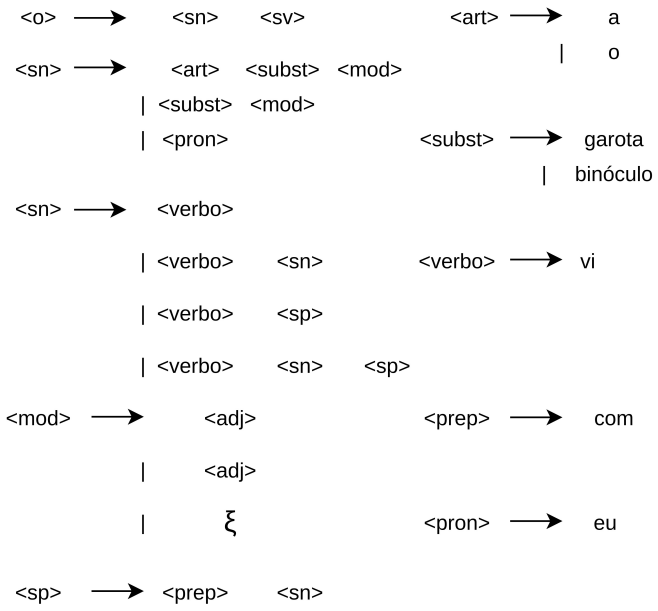
Segundo Barbosa et al. (2017), um pressuposto na maioria dos trabalhos na área de PLN é que a unidade básica de análise de significado é a frase. Uma frase expressa uma proposição, uma ideia ou um pensamento, além de dizer algo sobre o mundo real ou imaginário. As frases não são, no entanto, apenas uma sequência linear de palavras e, por isso, é amplamente reconhecido que para realizar qualquer tarefa com frases, requer uma análise fiel de cada frase, as quais determinam suas estruturas de uma maneira ou de outra. A análise sintática é aquela onde uma sequência de unidades lexicais, tipicamente uma oração, será decomposta para determinar sua descrição estrutural de acordo com uma gramática formal (LJUNGLÖF; WIRÉN, 2010).

Em Junior (2013), é definido que *parse* sintático normalmente não é um fim por si só, entretanto um meio para a seleção de descritores ou extração de significado. Para isso, utilizam-se os resultados das fases anteriores, sejam quaisquer unidades lexicais tokenizadas e classificadas. O resultado da análise sintática é uma hierarquia sintaticamente estruturada preparada para interpretação semântica. Uma das formas de se representar a análise sintática é por meio de árvores sintáticas ou árvore de *parse*. Segundo Wintner (2010), a escolha de uma

árvore é evidente, uma vez que as regras de produção derivam símbolos não terminais até os terminais.

Em Moura (1996), é demonstrada a parte formal da construção de uma gramática, conforme a a Figura 2.

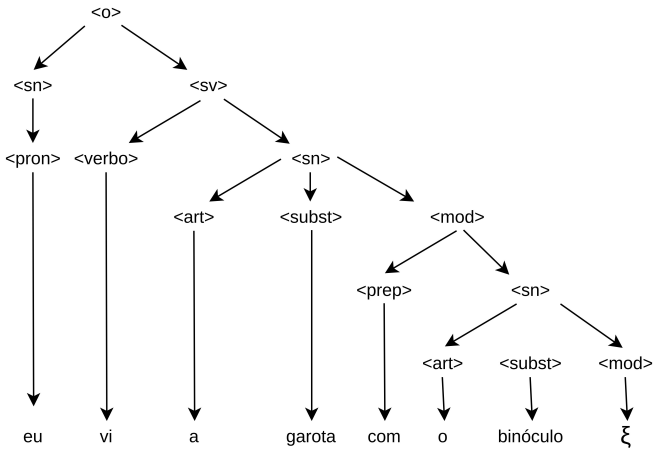
Figura 2 – Utilizando a frase "eu vi a garota com o binóculo" como exemplo.



Fonte: Adaptado de (MOURA, 1996).

A frase "eu vi a garota com o binóculo" é composta por dois lados. Ao lado esquerdo são demonstrados os símbolos pré terminais, que decompõem sempre em uma produção, enquanto o lado direito, corresponde aos termos léxicos. Assim, a árvore sintática resultante é ilustrada na Figura 3.

Figura 3 – Árvore sintática.



Fonte: (MOURA, 1996).

Segundo Barbosa et al. (2017), a árvore sintática é a representação de todas as etapas na derivação da frase a partir do nó raiz. Isso representa que cada nó interno na árvore representa a aplicação de uma regra da gramática. Um projeto desenvolvido por Martins, Hasegawa e Nunes (2012), tem como objetivo um decompositor de sentenças em idioma português. Esse *parser*, denominado Curupira, utiliza um léxico mofossintaticamente anotado para suporte e treinamento da ferramenta. Uma gramática completa, com aproximadamente seiscentas regras é reproduzida para inicializar os algoritmos de *parse* (JUNIOR, 2013).

3.5 Análise Semântica

Segundo Goddard e Schalley (2010), a análise semântica trata do significado da sentença, partindo da organização realizada pela fase anterior, qual seja a análise sintática, um objeto mais estruturado para manipulação e extração é gerado, permitindo a perscrutação do entendimento. Este entendimento pode ser analisado com base nas relações válidas entre as palavras, a partir de seus conceitos (CHAUDIRON, 2007). De acordo com Nirenburg e Raskin (2004), é necessário um formalismo para representação do significado textual e conhecimento estático para processamento semântico, o qual inclui o mapeamento das estruturas de dependências sintáticas e semânticas, tratamento de referências e regras da estruturação textual.

Segundo Barbosa et al. (2017), a análise semântica é bastante utilizada quando se leva em consideração textos longos, aplicações de PLN específicas podem incluir recuperação de informação, extração de informação, sumarização de textos, mineração de dados, tradução para linguagem de máquina e auxiliares de tradução. Conforme Goddard e Schalley (2010), o objetivo final da análise semântica, tanto para pessoas quanto para sistemas de PLN, é entender o enunciado: não apenas ler o que está escrito, mas compreender as circunstâncias, a incorporação de informações providas pelo enunciado e obter uma resposta. A compreensão de um enunciado é um processo complexo que depende do resultado de análises léxicas e sintáticas, assim como contexto e raciocínio comum (BARBOSA et al., 2017).

Na linguística, a análise semântica trata do significado das palavras, expressões fixadas, sentenças completas e enunciados contexualizadas (GODDARD; SCHALLEY, 2010). A línguistica é fundamental para PLN apresentando três características chaves. A primeira são as intuições semânticas do falante nativo no uso das expressões no contexto. A segunda é a importância de relacionar o significado às várias análises superficiais da linguagem, enquanto a terceira é o respeito reconhecido não exclusivamente à linguagem propriamente dita, mas também a todas as suas variações (CRUSE, 2011).

Convencionalmente, a análise semântica é categorizada em semântica léxica e semântica gramatical ou combinatória. Segundo Cruse (2011), na primeira categoria, os substantivos e adjetivos, ou seja, palavras com conteúdo, são mais importantes do que preposições ou artigos. Já a segunda, de acordo com Junior (2013), é focada na compreensão das infinitas

combinações de unidades léxicas em frases que obedeçam as regras gramaticais. Apesar da divisão, atualmente se reconhece que ambas as áreas interagem e se misturam de várias formas (GODDARD; SCHALLEY, 2010).

Segundo Goddard e Schalley (2010), o problema mais destacado da análise semântica é a resolução de ambiguidade. Existem três tipos de ambiguidades, onde, no trabalho de Bird, Klein e Loper (2009), são classificadas como: ambiguidade léxica, ambiguidade de escopo e ambiguidade referencial. As três podem ser descritas resumidamente da seguinte forma: Quando baseada nas múltiplas interpretações de certas palavras de um texto, será ambiguidade léxica, já quando um texto possuir alguns quantificadores, modais ou negativos, podendo aplicar a diferentes áreas do texto, é denominado ambiguidade de escopo. Por fim, quando há possibilidade de um pronome ou qualquer outra unidade referencial não estar evidentemete estabelecida, é denominada ambiguidade referencial.

Concluindo, conforme Aranha (2007), nesta etapa ocorre o encontro de termos ambíguos, de sufixos e afixos, ou seja, questões de significado associados aos morfemas componentes de uma palavra, o sentido real da frase ou palavra, ou seja, na análise semântica concentra-se na tradução da linguagem natural para a linguagem formal, sem ambiguidade, pretendendo depurar e extrair o significado de uma declaração, além de auxiliar na etapa da análise pragmática. Para pesquisas e técnicas de PLN, a análise semântica procura investigar a construção de textos a nível sintático, podendo representá-la por uma rede semântica.

3.6 Análise Pragmática

De acordo com Jurafsky e Martin (2019), a análise pragmática consiste em determinar a relação entre a linguagem e o contexto, onde o contexto inclui perceber como a língua se refere a pessoas e objetos, como o discurso está estruturado e como este é interpretado pelo ouvinte. A análise pragmática interpreta o todo e não somente uma parte, ou seja, verifica se o significado da análise semântica está correto, além de determinar significados que não estejam claros.

Segundo Pinto (2015), esta análise permite resolver algumas situações de ambiguidade sintática, porém, em um contexto em que os participantes não partilham o mesmo contexto, pode ocorrer ambiguidade na comunicação. Um exemplo é a frase "Hoje está calor!", onde é totalmente interpretada se ambos os mediadores conhecerem o conceito de ""estar calor" do emissor. Contextualizando, pode ser que para um russo, onde a temperatura média anual é de $5\,^{\circ}$C, "estar calor" se baseie em $12\,^{\circ}$C a $18\,^{\circ}$C, porém para um brasileiro, onde a temperatura média anual é de $26\,^{\circ}$C a $28\,^{\circ}$C, "estar calor" pode significar estar $32\,^{\circ}$C.

Neste tipo de análise é necessário compreender que a linguagem não consiste em palavras ou frases isoladas, mas em frases relacionadas e agrupadas. Esse grupo de frases é chamado de discurso que, segundo a pragmática, sua interpretação é influenciada pelo contexto

(JURAFSKY; MARTIN, 2019). Conforme Mitkov et al. (2010), o discurso deve ser analisado como um todo, uma vez que a declaração não se concentra exclusivamente em uma oração, mas sim, no conjunto delas.

Em suma, de acordo com Junior (2013), a análise pragmática é a mais imatura em PLN. Sua profundidade e complexidade intimidam pesquisas na área. Percebe-se, porém, que aplicações podem bem utilizar seus resultados, sobretudo aquelas que mais se aproximam do usuário final, como por exemplo, sistemas de pergunta e resposta. Em conformidade com Dale (2010), os resultados alcançados atualmente em nível léxico/morfológico e sintático são mais significativos do que em nível semântico e pragmático.

3.7 Representação de Palavras

Ao trabalhar com idiomas e palavras, é preciso realizar uma transformação nestes dados para um formato que possa ser entendido em conjuntos de algoritmos e processos. Para a criação das representações textuais deste projeto foram utilizadas três abordagens definidas na subseções a seguir.

3.7.1 *Bag-of-Words*

O *Bag-of-Words (BoW)* é um modelo de representação de texto que descreve a ocorrência de palavras em um documento. Este modelo calcula a representação do texto como uma bolsa de palavras, desconsiderando a gramática e até mesmo a ordem das palavras, mas mantendo a multiplicidade, ou seja, a frequência de cada palavra no texto. Assim, o resultado é um vetor n-dimensional do mesmo tamanho do dicionário, que é um conjunto das palavras mais representativas de todo o conjunto.

Segundo Goldberg e Hirst (2017), o BoW é um procedimento de extração de recursos muito comum para frases e documentos, na qual, observa-se o histograma das palavras dentro do texto, ou seja, considerando cada palavra contada como um recurso. Nesta abordagem, o modelo se preocupa apenas com a ocorrência de palavras conhecidas no documento, descartando a ordem e a estrutura conceitual da palavra no texto. A Figura 4 apresenta uma melhor visualização de como funciona este método.

A Figura 4 mostra a conversão do conjunto de textos em uma matriz de distribuição de frequência. Assim, os documentos são analisados e representados, na qual, cada palavra contida no documento recebe uma coluna correspondente e cada valor atribuído é a representação da frequência de vezes em que essa palavra aparece no documento. Segundo Uiterkamp (2019), como cada dimensão do vetor BoW corresponde a uma palavra no vocabulário, sem levar em consideração como essas palavras estão dispostas em relação às outras, neste modelo não é possível usar informações textuais para diferenciar o significado das palavras, assim, a representação BoW funciona como se as palavras fossem independentes umas das outras e não

Figura 4 – Representação da arquitetura BoW.

tivessem relação umas com as outras. Em suma, o modelo BoW é bastante simples e flexível e pode ser usada de várias maneiras para extrair recursos de documentos.

3.7.2 *Term Frequency - Inverse Document Frequency*

Para adentrar nessa técnica, é plausível explicar o problema ocorrido com o *bag-of-words*. O principal problema está relacionado as palavras mais frequentes, pois elas "dominam" o documento e podem não representar tanta informação sobre o contexto, ou seja, o que acontece são as representações de maiores pesos para documentos longos do que para documentos pequenos, assim, se houver um texto de uma página, ele terá uma importância maior do que um texto que possui somente uma linha. Para isso, o modelo TF-IDF (*Term Frequency – Inverse Document Frequency*) busca redimensionar a frequência das palavras por meio de uma constância de aparições em todos os documentos.

Segundo Salton e McGill (1986), TF-IDF é uma medida de atribuição de pesos que favorece termos que ocorrem em poucos documentos de uma coleção. Esta medida é utilizada para avaliar o quão importante é um termo para o documento em que ele ocorre, em relação a todos os documentos da coleção. Pode-se descrever que o termo TF se refere à "frequência do termo", ou seja, quantas vezes um termo aparece em determinado documento, texto, página, etc. Assim, quanto maior for a frequência no documento analisado, maior será a importância do termo. Já o termo IDF significa "frequência inversa dos documentos", ou seja, com que frequência o termo aparece em todos os documentos, textos, páginas, etc.

Conforme Yan et al. (2020), TF-IDF equilibra o peso das palavras que sempre têm uma frequência alta. Ele assume que a importância de uma palavra aumenta proporcionalmente à sua frequência em um documento, mas é compensada por sua frequência em todo o corpus. Em suma, quanto maior a frequência nos documentos, menor será a importância do termo (SALTON; BUCKLEY, 1988).

O cálculo da medida TF-IDF em um documento é a combinação de sua medida local

(TF) e global (IDF). Sua medida global, pode ser observada na Equação 1:

$$idf = log\frac{|D|}{|d \epsilon D : t \epsilon d|}, \tag{1}$$

na qual $|D|$, é o número total de documentos da coleção, e o termo $|d \epsilon D : t \epsilon d|$, representa o número de documentos em que o termo t aparece. Assim, juntando a medida global com a medida local cria-se a Equação 2:

$$tfidf = tf_{x,y} \cdot idf_x, \tag{2}$$

em que $tf_{x,y}$, indica a frequência da palavra (x) no documento atual (y) e, o termo idf_x, corresponde a quão rara é a palavra nos documentos analisados.

3.7.3 *Word Embedding*

Conforme Goldberg (2017), *word embedding* são métodos que fornecem boas representações vetoriais mapeando a semântica e sintática de uma linguagem natural, assim, as palavras de um conjunto de texto são mapeadas para vetores de baixa dimensão reduzindo o custo computacional e possuindo uma melhor precisão em tarefas de processamento de linguagem natural.

O *word embedding* é um tipo de representação de palavras que permite que estas palavras sejam compreendidas por algoritmos de aprendizado de máquina. Essas representações são mapeadas em forma de vetores de alta dimensionalidade, onde cada entrada é uma medida de associação entre a palavra e o contexto onde ela se encontra (GOLDBERG Y.; LEVY, 2014).

Neste contexto, o *word embedding* possui relação matemática entre as palavras, pois pode-se utilizar operações sobre estes vetores, por exemplo, ao possuir uma dimensão de 64 números isso representa que cada palavra analisada possui uma matriz de 64 posições que realiza sua interpretação, assim, pode-se fazer operações nestas matrizes afim de encontrar novos dados (GOLDBERG Y.; LEVY, 2014). Para exemplificar melhor este processo, a Figura 5 mostra operações envolvendo algumas palavras.

O que a Figura 5 remete é, que pode-se, à partir de relações matemáticas, descobrir novos vínculos entre as palavras, formando um contexto de aprendizagem que é muito útil e eficaz para os algoritmos de redes neurais, especialmente. As operações realizadas na Figura 5 mostram um espaço, no qual, a matriz que representa a palavra "rei" está diretamente relacionada a palavra "rainha" assim como, a palavra "homem" está diretamente relacionada a palavra "mulher". Desta forma, pode-se realizar operações nestes pontos, como por exemplo, a matriz "rei" subtraída da matriz "homem" adicionada a matriz "mulher" resulta na matriz "rainha". As mesmas operações realizadas neste contexto, também podem ser manuseadas em tempos verbais.

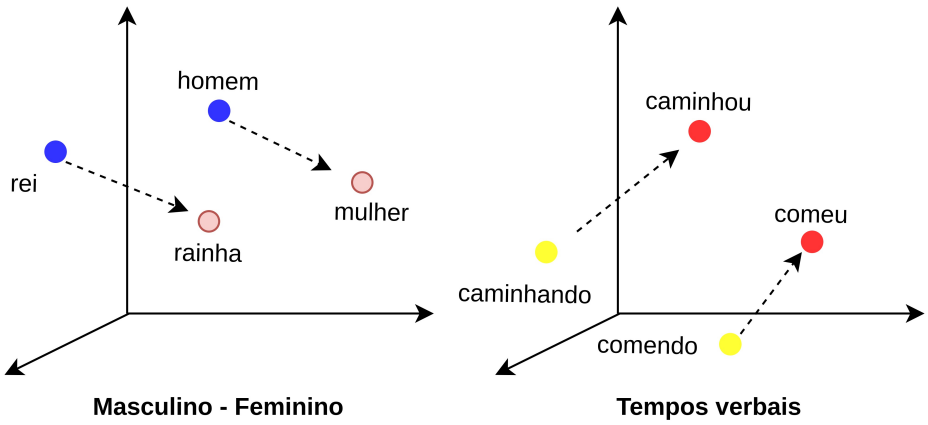


Figura 5 – Representação *Word Embedding*.

3.8 Arquiteturas

São dispostos nesta seção algumas das arquiteturas definidas como estado da arte atualmente, suas definições e conceitos, além de sua importância neste projeto.

3.8.1 Arquitetura *Transformer*

Esta arquitetura foi desenvolvida justamente pelo problema do sobrecarregamento da última camada de entrada de uma rede neural recorrente. Mesmo uma rede neural recorrente clássica que possui um *context vetor* que é um mecanismo de atenção, ela acaba não obtendo um bom desempenho com frases e textos muito grandes, como por exemplo, uma aplicação de tradutor. Assim, os criadores da arquitetura *Transformer* pensaram em adicionar e modificar o comportamento da rede neural recorrente. Para uma melhor exemplificação, a Figura 6 possui uma rede neural recorrente clássica e ao lado a arquitetura *Transformer* (VASWANI et al., 2017).

Pode-se observar que na arquitetura *Transformer* há três vezes o mecanismo de atenção, assim sendo, ela consegue fazer com a que rede neural consiga se adaptar melhor com textos maiores. O foco principal dessa arquitetura é justamente o trabalho com os mecanismos de atenção visualizados de cor laranja na imagem. E assim, como na arquitetura clássica (lado esquerdo da Figura 6), há também na arquitetura (a direita) um *enconder* e um *decoder*. Eles possuem a ideia onde na parte do *encoder* são realizados os codificadores e na parte do *decoder* são realizados os decodificadores, ou seja, ocorrem os *inputs* pela camada do codificador e no final do decodificador possui as respostas por meio de probabilidades que o algoritmo irá indicar quais são as palavras previstas com base na Figura 6 (VASWANI et al., 2017).

A principal diferença entre essas arquiteturas é o fato que na arquitetura clássica ocorre os *inputs* palavra por palavra, entretanto na arquitura *Transformer* ocorrem primeiramente, as entradas de dados, depois existe uma camada *embedding* e, assim, são repassados para a arquitetura as frases inteiras, textos completos, documentos, etc. Ou seja, o processamento

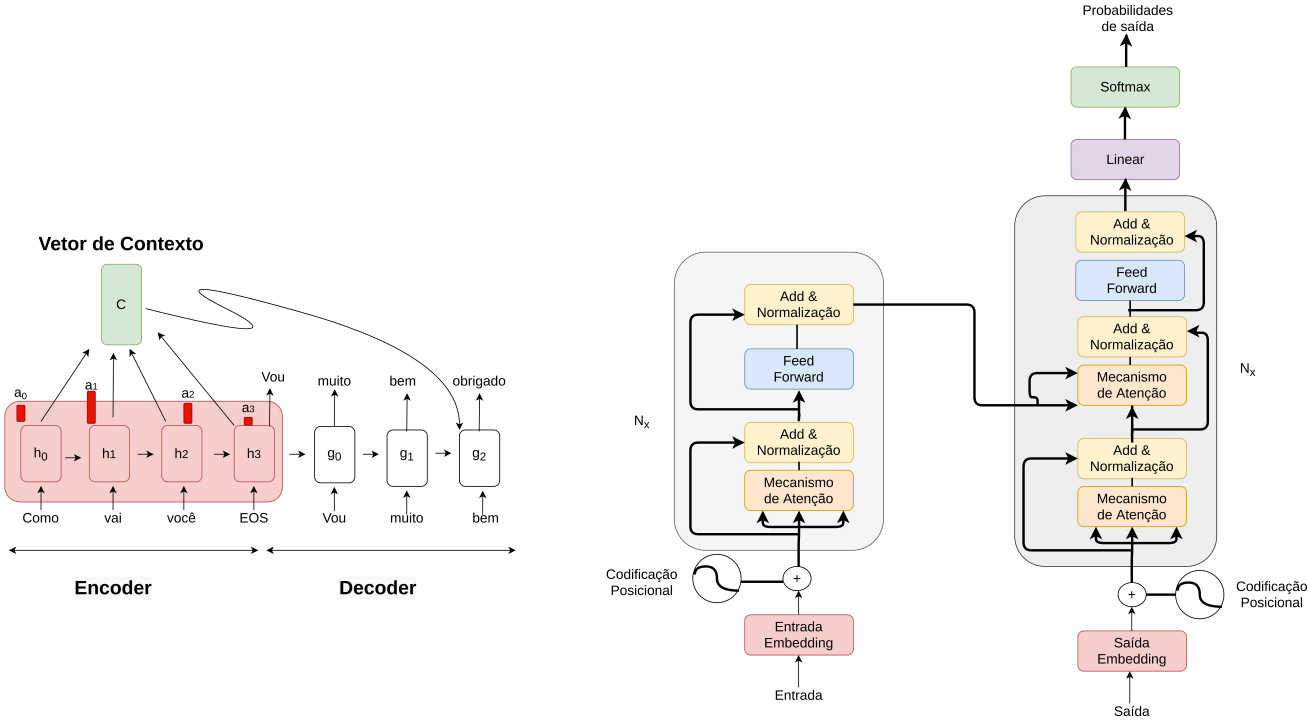


Figura 6 – Arquitetura Clássica e Arquitetura *Transformer*.

não é realizado palavra por palavra, mas, sim, por todo o contexto de uma frase inteira. Para compreender melhor a diferença entre as arquiteturas, pode-se exemplificar com a ação de traduzir uma página escrita em inglês para o espanhol. Na arquitetura clássica irá ocorrer a entrada palavra por palavra do documento, ao contrário do *Transformer* que terá a entrada do tamanho do texto encontrado no documento (VASWANI et al., 2017).

3.8.2 Bidirectional Encoder Representations from Transfomers (BERT)

O *BERT - Bidirectional Encoder Representations from Transfomers*, em português, representação de codificador bidirecional de transformadores. Esta arquitetura foi construída baseando-se na arquitetura *Transformer* por obter um grande desempenho em trabalhos relacionados a área de PLN. Possuindo como base o nome desta arquitetura BERT, pode-se dividi-lá em três partes:

- *Encoder Representations*: Sistema de modelagem de linguagem com pré-treinamento com dados não rotulados, ou seja, o modelo vai retornar uma representação matricial das palavras com base na geração do modelo de linguagem com dados não são rotulados (DEVLIN et al., 2018).
- *from Transformer*: Baseado em uma das arquiteturas mais poderosas de PLN. O Transformer define a arquitetura do BERT (DEVLIN et al., 2018).

- *Bidirectional*: Utiliza o contexto da esquerda e direita para entendimento das palavras, ou seja, ele terá mais informação sobre o contexto, realizando uma análise em cada item do texto com a forma bidirecional (DEVLIN et al., 2018).

Assim como o *word embedding*, o BERT também é uma técnica de representação de texto, sendo uma fusão de vários algoritmos avançados de aprendizado profundo, como o codificador bidirecional *Long Short-Term Memory* (LSTM) e Transformers. O BERT foi desenvolvido por pesquisadores do *Google* em 2018 e provou ser o estado da arte para uma variedade de tarefas de processamento de linguagem natural, como classificação de texto, resumo de texto, geração de texto, etc. Recentemente, o *Google* anunciou que o BERT está sendo usado como parte principal do algoritmo de pesquisa para entender melhor as consultas (DEVLIN et al., 2018).

4 Aprendizado de Máquina

Nesta seção são abordados a definição, tipos de aprendizado, além dos classificadores utilizados nos experimentos desta dissertação.

4.1 Definição

De acordo com Géron (2019), nas últimas décadas, com a crescente complexidade dos problemas a serem tratados computacionalmente e do volume de dados gerados por diferentes setores, tornou-se clara a necessidade de ferramentas computacionais mais sofisticadas, que fossem mais autônomas, reduzindo a necessidade de intervenção humana e dependência de especialistas. Para isso, essas técnicas deveriam ser capazes de criar por si próprias, a partir da experiência passada, uma hipótese ou função, capaz de resolver o problema que se deseja tratar.

Segundo Marsland (2011), aprendizado de máquina é uma área de inteligência artificial que estuda algoritmos que sejam eficientes em aprender e resolver problemas de maneira autônoma. Conforme Mitchell (1997), diz-se que um programa de computador aprende pela experiência E em relação a algum tipo de tarefa T e alguma medida de desempenho P se o seu desempenho em T, conforme medido por P, melhora com a experiência E. Isso quer mostrar, que o processo de pesquisa em aprendizado de máquina consiste em examinar e experimentar as estratégias mais eficazes para a construção de programas que aprendem a partir da experiência, adquirindo conhecimento de forma automática, ou seja, ser capaz de tomar decisões baseado em experiências acumuladas por meio da solução bem sucedida de problemas anteriores (MITCHELL, 1997).

Para Alpaydin (2010), aprendizagem de máquina é o desenvolvimento de programas de computadores capazes de otimizar o seu desempenho utilizando dados de exemplos ou de experiência anterior. Segundo Brink e Richards (2014), AM é uma abordagem guiada a dados para a resolução de problemas, na qual padrões ocultos em um conjunto de dados são identificados e utilizados para ajudar na tomada de decisões. Conforme Géron (2019), aprendizado de máquina é a ciência e a arte da programação de computadores para que eles possam aprender com os dados. Resumidamente, segundo Géron (2019), o aprendizado de máquina é ótimo para:

- Problemas para os quais as soluções existentes exigem muita configuração manual ou longas listas de regras: um algoritmo de AM geralmente simplifica e melhora o código;
- Problemas complexos para os quais não existe uma boa solução quando utilizamos uma

abordagem tradicional: técnicas de AM podem encontrar uma solução;

- Ambientes flutuantes: um sistema de AM pode se adaptar a novos dados;
- Compreensão de problemas complexos e grandes quantidades de dados.

No geral, existem tipos diferentes de sistemas de AM que possuem diversas técnicas e ferramentas para diversos problemas. De acordo com Géron (2019), os sistemas de AM podem ser classificados de acordo com a quantidade e o tipo de supervisão que recebem durante o treinamento. Existem quatro categorias principais de aprendizado: supervisionado, não supervisionado, semi-supervisionado e por reforço.

4.2 Aprendizado Supervisionado

No aprendizado supervisionado, os dados de treinamento que são utilizados servem para fornecer ao algoritmo as soluções desejadas, denominadas rótulos. A classificação é uma tarefa típica do aprendizado de máquina. O filtro de *spam* é um bom exemplo disso: ele é treinado com muitos exemplos de *e-mails* junto às classes (*spam* ou não *spam*) e deve aprender a classificar novos *e-mails* (GÉRON, 2019).

Conforme Maimon e Rokach (2005), métodos supervisionados são métodos que tentam descobrir a relação entre os atributos de entrada e um atributo de destino. A relação descoberta é representada em uma estrutura definida como modelo. Geralmente, modelos descrevem e explicam os fenômenos que estão escondidos no conjunto de dados, e podem ser usados para prever ou classificar o valor da classe final, com base no conhecimento adquirido dos valores dos atributos de entrada.

Segundo Brink e Richards (2014), algoritmos de aprendizado supervisionado são os mais comuns na área de AM, sendo que a maioria dos problemas em que a área de AM pode auxiliar a resolver é supervisionada por natureza. O aprendizado supervisionado possui dois modelos:

- Classificação: onde a predição é realizada para um atributo classificador que assume valores discretos. Por exemplo, os classificadores podem ser usados para classificar os consumidores de hipotecas como bom e ruim (MAIMON; ROKACH, 2005).
- Regressão: como o próprio nome já diz, realiza previsão numérica, onde se utiliza a variável contínua. Por exemplo, um regressor pode prever a demanda de um determinado produto dadas suas características (MAIMON; ROKACH, 2005).

Em suma, a abordagem de aprendizado supervisionado consiste em utilizar uma série de exemplos já rotulados, para induzir um modelo que seja capaz de classificar novas instâncias de forma precisa, com base no aprendizado obtido durante a fase de treinamento.

4.3 Aprendizado Não Supervisionado

No aprendizado não supervisionado, os dados de treinamento não são rotulados. Um exemplo desta aprendizagem é detectar grupos de visitantes semelhantes, dado muitos dados sobre usuários visitantes de um *blog*. (GÉRON, 2019).

Segundo Berry, Mohamed e Yap (2019), o aprendizado de dados não supervisionado envolve o reconhecimento de padrões sem o envolvimento de um atributo de destino, ou seja, todas as variáveis utilizadas na análise são utilizadas como entradas e devido à abordagem, as técnicas são adequadas para técnicas de agrupamento e mineração de associação.

A utilização desse aprendizado é normalmente utilizado quando não se conhece muito a respeito da estrutura de um conjunto de dados. Por exemplo, a utilização do modelo em um conjunto de dados com várias informações sobre plantas, onde será separado esse conjunto em três categorias. Com isso, não é especificado para o modelo o que caracteriza cada categoria, o modelo irá encontrar características semelhantes e diferentes entre todos os dados de maneira que separe todo o conjunto em três grupos.

4.4 Aprendizado Semi-supervisionado

Alguns algoritmos podem lidar com dados de treinamento parcialmente rotulados, ou seja, uma grande quantidade de dados não rotulados e um pouco de dados rotulados. Isso é chamado de aprendizado semi-supervisionado. Alguns serviços de hospedagem de fotos, como o *Google* Fotos, são bons exemplos desta aprendizagem (GÉRON, 2019).

De acordo com Reddy, Viswanath e Reddy (2018), o aprendizado semi-supervisionado é o ramo do aprendizado de máquina preocupado com o uso de dados rotulados e não rotulados para executar determinadas tarefas de aprendizado. Normalmente, algoritmos de aprendizado semi-supervisionado tentam melhorar o desempenho em uma dessas duas tarefas utilizando informações geralmente associadas à outra.

4.5 Aprendizado por Reforço

O aprendizado por reforço é bem diferente dos demais, o sistema de aprendizado, chamado de agente nesse contexto, pode observar o ambiente, selecionar e executar ações e obter recompensas ou penalidades. Ele deve aprender por si só qual é a melhor estratégia, chamada de política, para obter o maior número de recompensas ao longo do tempo. Uma política define qual ação o agente deve escolher quando está em determinada situação. O programa *AlphaGo* da *DeepMind* é um bom exemplo deste tipo de aprendizado (GÉRON, 2019). Neste método, o computador é estimulado a aprender com base em tentativas e erros. O processo é otimizado por meio da prática, ensinando o sistema a priorizar certos hábitos.

Em suma, aprendizagem por reforço é o treinamento de modelos de aprendizado de máquina para tomar uma sequência de decisões. O agente aprende a atingir uma meta em um ambiente incerto e potencialmente complexo, assim, o sistema utiliza métodos de tentativa e erro para encontrar uma solução para o problema. Assim, a inteligência artificial recebe recompensas ou penalidades pelas ações que executa, onde seu objetivo é maximizar a recompensa total.

4.6 Classificadores Padrões

Com o objetivo de classificar automaticamente as notícias, foram explorados algoritmos de classificação baseados em AM de diferentes paradigmas. Entretanto, todos os algoritmos são da categoria de aprendizado supervisionado, pois foram informados quais rótulos devem possuir no final de todo o processo de aprendizagem. No caso, se as notícias são falsas ou verdadeiras.

O resultado final foi gerado com base em seis classificadores: Floresta de Caminhos Ótimos (OPF), Máquinas de Vetores de Suporte (SVM), Perceptron Multicamadas, do inglês *Multilayer Perceptrons* (MLP), *Naive Bayes*, *Random Forest*, Rede Neural Convolucional, do inglês *Convolutional Neural Network* (CNN) e o BERT. Os testes foram realizados com as representações textuais dos seguintes métodos:*Bag-of-Words* (BoW), a incrementação do *Term Frequency – Inverse Document Frequency* (TF-IDF), utilização do FastText, além das representações textuais da arquitetura BERT mencionadas na seção 3. A seguir, é apresentado uma breve introdução sobre os tipos de classificadores utilizados no projeto para um melhor entendimento.

4.6.1 Floresta de Caminhos Ótimos (OPF)

Segundo Falcão, Stolfi e Lotufo (2004), o classificador Floresta de Caminhos Ótimos apresenta uma proposta para classificação supervisionada de padrões utilizando conceitos da teoria dos grafos. O método para classificação supervisionada de padrões baseado em Floresta de Caminhos Ótimos, proposto por (PAPA, 2008), modela o problema de reconhecimento de padrões como sendo um grafo, no qual os nós são as amostras, representadas pelos respectivos vetores de atributos, e os arcos são representados por uma relação de adjacência entre as amostras.

Segundo Papa (2008), as amostras que melhor representam as classes do conjunto de treinamento são denominadas protótipos e, após sua identificação, um processo de competição é iniciado entre elas a fim de oferecer caminhos de custo ótimo para as demais amostras do conjunto de dados e seus respectivos rótulos.

O algoritmo de Floresta de Caminhos Ótimos é o mesmo utilizado na Transformada Imagem Floresta, do inglês *Image Foresting Transform* (IFT) (FALCÃO; STOLFI; LOTUFO,

2004), que tem sido utilizada em diversas aplicações de processamento de imagens. O classificador baseado em OPF estende as aplicações da IFT do domínio da imagem para a classificação de padrões no espaço de características (PAPA, 2008).

Em suma, o classificador OPF baseia-se em:

- Um subconjunto de amostras protótipos e
- Uma função de custo de caminho para o grafo de treinamento a fim de agrupar amostras com características similares.

A ideia é dividir o grafo de treinamento em uma floresta de caminhos de custo mínimo, em que cada árvore tem como raiz um protótipo representando uma das *m* classes do conjunto de treinamento e todas as amostras pertencentes à árvore são classificadas com o mesmo rótulo da raiz. A classificação de uma nova amostra se dá a partir da busca pelo protótipo que lhe oferece o caminho ótimo dentre os caminhos oferecidos por todos os protótipos, isto é, a classificação baseia-se na força de conexidade entre o protótipo e a amostra nova; assim, o rótulo de uma nova amostra a ser classificada passa a ser o mesmo do protótipo mais fortemente conexo a ela.

4.6.2 Máquinas de Vetores de Suporte (SVM)

Uma máquina de vetores suporte é um modelo de aprendizado de máquina capaz de realizar classificações lineares ou não lineares, de regressão e até mesmo detecção de *outliers* (GÉRON, 2019). As SVM são particularmente adequadas para a classificação de conjuntos de dados complexos, porém de pequeno ou médio porte. Seu objetivo principal é criar um modelo que realiza com maior eficiência uma separação de classes no espaço, através de uma linha de separação, chamado de hiperplano (MARUMO, 2018).

Para uma visualização mais detalhada de como funciona este tipo de classificador, a Figura 7, demonstra um problema de classificação linearmente separável pelo hiperplano. No exemplo, as categorias finais são: círculo e quadrado. Assim, com a separação realizada com sucesso, pode-se mapear novos dados nesse espaço. Na Figura 7, caso um novo dado seja mapeado a direita do hiperplano, ele receberá a classe quadrado e, se os novos dados forem mapeados pelo lado esquerdo do hiperplano, esses dados serão da classe círculo.

Conforme mostrado na Figura 7, quanto maior for a distância das classes do hiperplano, melhor será o modelo obtido pois ele consegue generalizar bem os dois tipos de amostras (CAMPBELL, 2006). O objetivo deste classificador é dadas as entradas, tentar acertar em qual tipo de categoria aquelas entradas se encaixam. A criação desse hiperplano é feita através da utilização de vetores de suporte que são responsáveis por selecionar as amostras de cada classe mais próximas de seus respectivos lados (MARUMO, 2018).

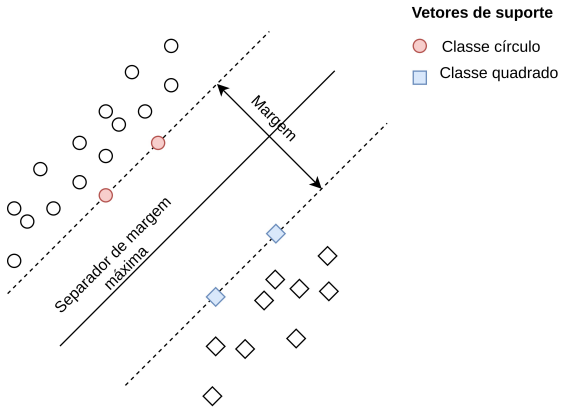


Figura 7 – Classificação de problema linearmente separável.

4.6.3 Naive Bayes

Segundo Pardo e Nunes (2002), *Naive Bayes* (NB) é um classificador probabilístico baseado no Teorema de Bayes, que mostra como determinar a probabilidade de um evento condicional através da probabilidade inversa. Seguindo o paradigma estátistico, o algoritmo faz uso de fórmulas estatísticas e cálculo de probabilidades para realizar a classificação (MITCHELL, 1997).

Conforme Mitchell (1997), o classificador *bayesiano* apresenta uma maneira de calcular a probabilidade da ocorrência de um evento, baseando-se em probabilidades obtidas da análise dos eventos passados. Estas informações são utilizadas para construir a base de conhecimento da aplicação. O modelo possui em seu nome o termo "*Naive*" cuja tradução em português significa simples ou ingênuo. Este nome se deve, pois seus atributos são analisados independentes uns dos outros, o que permite o seu emprego na resolução de diversos problemas (RUSSEL; NORVIG., 2013).

De acordo com Russel e Norvig. (2013), por trabalhar com as informações independentes, o classificador *bayesiano* consegue obter resultados em um menor tempo de execução, além de, ser mais fácil sua implementação, pois não considera o contexto que as informações estão inseridas. Assim, sua precisão é considerada máxima somente quando todas as informações forem independentes de si.

Em suma, para facilitar a computação, este classificador assume que a presença (ou ausência) de um atributo não tem relação alguma com qualquer outro atributo, no qual, seu objetivo é verificar se uma amostra analisada pertence ou não a uma determinada classe. A obtenção desta resposta é encontrada através de uma análise estatística das informações coletadas sobre as instâncias fornecidas (MITCHELL, 1997).

Partindo do teorema de Bayes, o seu funcionamento pode ser analisado através das

Equações 3 e 4:

$$P(h|X) = \frac{P(X|h) \times P(h)}{P(X)}, \tag{3}$$

na qual, os atributos X = $[a_1, a_2, a_3, a_4...a_n]$ são independentes:

$$P(h|X_1...X_n) = \frac{P(X_1...X_n|h) \times P(h)}{P(X_1...X_n)}, \tag{4}$$

em que o objetivo do classificador é encontrar $P(h|X)$, que representa a probabilidade a *posteriori* da classe h em razão do evento X. Para calculá-la, é necessário encontrar a probabilidade condicional $P(X|h)$, que representa a probabilidade de ocorrência do evento X dado a classe h. É preciso encontrar a probabildade a *priori* ($P(h)$) de ocorrência da classe a partir dos dados de treinamento. O termo $P(X)$, representa a probabilidade do evento dentro do conjunto de treinamento. Entretanto, em alguns casos onde são utilizadas mais de uma classe, a $P(X)$ pode ser desconsiderada do cálculo da probabilidade *bayesiana*, pois ela não altera a proporção entre os resultados (MITCHELL, 1997).

Normalmente, deseja-se encontrar a classe mais provável do evento, ou seja, a classe com o máximo valor da *posteriori*. Assim, conforme Pardo e Nunes (2002), para calcular a classe mais provável da nova instância, calcula-se a probabilidade de todas as possíveis classes, com isso, escolhe-se a classe com a maior probabilidade como rótulo da nova instância, onde o algortimo procura a classe que maximize o valor $(argmax)$ do termo $P(h|X)$, observado na Equação 5:

$$argmax P(h|X_1...X_n) = argmax P(X_1...X_n|h) \times P(h). \tag{5}$$

Há também a suposição "ingênua" que o classificador faz, onde é definido que todos os atributos de X da instância que se quer classificar são independentes, resultando no cálculo final do classificador, visto na Equação 6:

$$argmax P(h|X_1...X_n) = argmax \prod_{i=1}^{n} P(X_i|h) \times P(h), \tag{6}$$

na qual o termo $P(X_i|h)$, representa a frequência que o atributo X_i é encontrado na classe h, e n representa o total de atributos do evento. Para exemplificar sua efetivação, Pivetta (2013), sugere a atividade de classificação textual, em que o algoritmo NB realiza classificações de documentos e filtragem de *spam*. Assim, cada *e-mail* analisado é um evento e as palavras contidas no corpo do *e-mail* são os atributos.

Como dito, o NB é um classificador probabilístico baseado na aplicação do Teorema de *Bayes*, entretanto existem vários algoritmos que se beneficiam deste teorema, sendo eles: *Gaussian Naive Bayes (GaussianNB), Complement Naive Bayes Classifier, Bernoulli e Multinomial Models* (ZHANG, 2004).

4.6.4 Random Forest

Segundo Géron (2019), para definir o que é, e como funciona uma floresta aleatória, primeiro deve-se definir o conceito de uma árvore de decisão. Como as máquinas de vetores de suporte (SVM), as árvores de decisão são algoritmos versáteis de AM que podem executar tarefas de classificação, regressão e, até mesmo, tarefas *multioutput*.

Como o próprio nome já diz, esse algoritmo cria um modelo no formato de árvore, assim, esse formato é utilizado para classificar um conjunto de dados no qual é associado uma série de questões, e realizando assoações conforme as respectivas respostas. Cada nó é responsável por uma questão. Elas também são os componentes fundamentais das florestas aleatórias (GÉRON, 2019).

Segundo Marumo (2018), as *Random Forests* funcionam com o treinamento de muitas árvores de decisão em subcojuntos aleatórios das características, e em seguida calculam a média de suas previsões. Nos casos em que a árvore aleatória é usada para classificação, os critérios de partição mais conhecidos são baseados na entropia e no índice Gini. Conforme Rahmati et al. (2022), a entropia é a medida da incerteza associada a uma variável aleatória. A entropia aumenta com o aumento da incerteza ou aleatoriedade e diminui com a diminuição da incerteza ou aleatoriedade, ou seja, a entropia tem o objeitvo de medir a impura dos dados através da verificação da homogeneidade de um conjunto de dados. Sua fórmula pode ser visualizada na Equação 7:

$$H(S) = -\sum_{n=1}^{M} \pi \log_2(\pi), \tag{7}$$

em que S é um subconjunto do conjunto total de dados, o termo π é a probabilidade de um subconjunto de dados pertencer a classe n dentre as possíveis M classes (MARUMO, 2018). Já o índice de Gini é uma outra forma de medir a impureza, a desigualdade da partição de dados ou um conjunto de níveis de treinamento, medindo o grau de heterogeneidade dos dados associados a cada nó (RAHMATI et al., 2022). O índice Gini pode ser observado na Equação 8:

$$GI(S) = 1 - \sum_{n=1}^{M} \pi_i^2, \tag{8}$$

na qual π_i^2, representa a probabilidade de um objeto ser classificado em uma classe particular. Segundo Marumo (2018), quando o índice é igual a zero significa que o subcojunto contém apenas dados de uma única classe, ou seja, um nó puro. No projeto, foi utilizado o critério da entropia para a classificação com as florestas randômicas.

4.6.5 Perceptron Multicamadas

Segundo Haykin e Engel (2007), o princípio básico por trás de uma rede neural é uma coleção de elementos, neurônio artificial ou perceptron interligados, que foram desenvolvidos

pela primeira vez na década de 1950 por Frank Rosenblatt. A rede recebe várias entradas binárias, $[x_1, x_2, ...x_n]$ e produzem uma única saída binária se a soma for maior do que o potencial de ativação (GOYAL; PANDEY; JAIN, 2018).

Uma *Multilayer Perceptrons* (MLP) é constituída por um conjunto de três tipos de camadas: (i) entrada, (ii) saída e (iii) camadas ocultas. A camada de entrada recebe a informação, ou seja, onde os dados iniciais são inseridos. A camada de saída executa a tarefa necessária, como previsão e classificação. Um número arbitrário de camadas ocultas pode ser colocado entre as camadas de entrada e saída (ABIRAMI; CHITRA, 2020). De acordo com Souza (2012), uma das principais características de uma rede MLP é sua capacidade de resolver problemas não lineares, para isso, é necessário que a função de ativação dos neurônios pertencentes às camadas ocultas seja não-linear. Em geral, a função de ativação é *sigmoide*.

Conforme Lima et al. (2016), MLP são sistemas paralelos distribuídos, compostos por camadas de pequenas unidades Perceptrons, cada um contendo funções de ativação específicas e baseiando-se geralmente em uma forma de aprendizado supervisionado, no qual o erro da análise deve ser minimizado constantemente via algoritmos do tipo *Backpropagation*. Uma MLP é uma rede progressiva *(feedforward)*, pois as saídas dos neurônios em qualquer particular camada se conectam unicamente às entradas dos neurônios da camada seguinte, sem a presença de laços de realimentação. Consequentemente, o sinal de entrada se propaga através da rede, camada a camada, em um sentido progressivo (SOUZA, 2012). Em suma, de acordo com Haykin e Engel (2007), a rede MLP têm sido aplicada com sucesso na solução de diversos problemas difíceis, através do treinamento supervisionado com o algoritmo de retropropagação do erro.

4.6.6 Rede Neural Convolucional

As redes neurais convolucionais são bem adaptadas para reconhecimento de imagem e reconhecimento de escrita. Sua estrutura é baseada na amostragem de uma janela ou parte de uma imagem, detectando seus recursos e, em seguida, usando os recursos de várias camadas para construir uma representação, ou seja, esses modelos foram os primeiros modelos de aprendizado profundo.

Apesar da proposta inicial do desenvolvimento desta rede ter sido realizada em 1998 por (LECUN et al., 1998), as CNNs começaram a ser frequentemente utilizadas a partir da última década, visto que uma versão profunda desta foi treinada e bateu 10% com relação as outras técnicas em uma competição de classificar uma imagem de uma mil categorias (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Conforme Hijazi, Kumar e Rowen (2015), o design de uma CNN tem como inspiração o córtex visual humano. O córtex visual contém muitas células responsáveis pela detecção de luz em sub-regiões pequenas e sobrepostas do campo visual, chamadas de campos receptivos.

Essas células atuam como filtros locais sobre o espaço de entrada e quanto mais complexas a célula, maior seu campo receptivo.

Ao contrário das redes neurais simples que têm uma ou mais camadas ocultas, as CNNs têm muitas camadas. Este recurso permite que representem funções e variáveis altamente não lineares de forma compacta (DALTO, 2014). CNNs envolvem muitas conexões, e a arquitetura é tipicamente composta de diferentes tipos de camadas, incluindo convolução, pooling e camadas totalmente conectadas, além de regularização (FERREIRA; GIRALDI, 2017).

4.6.7 BERT

Segundo DEVLIN et al. (2018), o BERT possui duas etapas: pré-treinamento e o ajuste fino. Durante o pré-treinamento, o modelo é treinado em dados não rotulados em diferentes tarefas de pré-treinamento. Para o ajuste fino, o modelo BERT é inicializado primeiro com os parâmetros pré-treinados e todos os parâmetros são ajustados usando dados rotulados das tarefas *downstream* (aprendizado supervisionado que utiliza um modelo ou componentes pré-treinados). Cada tarefa de *downstream* tem modelos separados e ajustados, embora sejam inicializados com os mesmos parâmetros pré-treinados.

De acordo Cordeiro (2019), o BERT, dada uma frase inteira, irá gerar um *word embedding* para cada palavra dentro do contexto da frase, assim, palavras iguais dentro de contextos diferentes terão representações distintas em cada uma das frases. A arquitetura do modelo do BERT é um codificador Transformer bidirecional multicamadas com base na implementação original descrita em (VASWANI et al., 2017), ou seja, a arquitetura do BERT ao treinar e predizer *embeddings*, tanto as palavras anteriores quanto as seguintes são levadas em consideração, além de, utilizar mecanismos de atenção para armazenar cada vez mais informação das palavras na frase.

O modelo-base do BERT contém um codificador com 12 blocos de Transformers, 12 mecanismos de autoatenção e o tamanho oculto de 768. O BERT recebe uma entrada de uma sequência de no máximo 512 *tokens* e produz a representação desta sequência (SUN et al., 2019). Deste modo, a sequência de entrada receberá um ou dois segmentos especiais, em que o primeiro *token* da frase será sempre $[CLS]$ que contém a classificação da frase e ao final de cada sentença, o *token* especial $[SEP]$ é colocado para separar cada frase. Entretanto, quando o BERT é utilizado para tarefas de classificação de textos, apenas o *token* $[CLS]$ é utilizado (SUN et al., 2019).

4.7 Classificadores para Detecção de Anomalia

Para os experimentos de detecção de anomalia envolvendo *fake news*, são utilizados sete classificadores que possuem as seguintes categorias:

1. Modelos probabilísticos:

    a) Detecção de valores discrepantes baseada em distribuição cumulativa empírica (ECOD) (LI et al., 2022): Este algoritmo é inspirado pelo fato de que os valores discrepantes são frequentemente os "eventos" que aparecem nas caudas de uma distribuição. Resumidamente, ECOD primeiro avalia a distribuição subjacente dos dados de entrada de forma não paramétrica, calculando a distribuição cumulativa empírica pela dimensão dos dados usando as distribuições para estimar as probabilidades de cauda por dimensão para cada ponto de dados. Finalmente, calcula uma pontuação atípica para cada ponto de dados agregando as probabilidades de cauda estimadas em todas as dimensões.

2. Modelos baseados em proximidade:

    a) Fator Local Outlier (LOF) (CHENG; ZOU; DONG, 2019): Este classificador é um algoritmo de detecção de discrepância baseado em densidade que encontra discrepâncias calculando o desvio local de um determinado ponto de dados. A determinação do valor discrepante é julgada com base na densidade entre cada ponto de dados e seus pontos vizinhos, portanto, quanto menor a densidade do ponto, maior a probabilidade de ser identificado como o valor atípico.

    b) Fator Outlier Local Baseado em Clustering (CBLOF) (HE; XU; DENG, 2003): É uma medida para identificar o significado físico de um outlier e dá importância ao comportamento dos dados locais.

3. Modelos lineares:

    a) One-class SVM (GÉRON, 2019): O algoritmo busca separar as instâncias no espaço de alta dimensão da origem. Em um espaço de alta dimensão, o algoritmo tenta encontrar um hiperplano que separe as amostras do conjunto de treinamento da origem, isso corresponderá a encontrar uma pequena região que englobe todas as instâncias, portanto, se uma nova instância não caber nessa região, ela é uma anomalia.

    b) One-Class SVM using Stochastic Gradient Descent (SGDOneClassSVM) (PEDREGOSA et al., 2011): O algoritmo implementa uma versão linear online do SVM de classe única usando descida de gradiente estocástica. Combinado com técnicas de aproximação de kernel, este algoritmo pode ser usado para aproximar a solução de uma SVM de uma classe kernelizada, com uma complexidade linear no número de amostras.

4. Conjuntos atípicos:

    a) Isolation Forest (GÉRON, 2019): O algoritmo constrói uma Floresta Aleatória na qual cada Árvore de Decisão cresce aleatoriamente: em cada nó, ele escolhe um

recurso aleatoriamente e, em seguida, escolhe um valor limite aleatório (entre um valor mínimo e um valor máximo) para dividir o conjunto de dados em dois. O conjunto de dados é gradualmente dividido em pedaços até que todas as instâncias acabem isoladas das outras instâncias. Uma anomalia geralmente está longe de outras instâncias e tende a ficar isolada em menos etapas do que as instâncias normais.

5. Modelos baseados em grafo:

   a) Optimum-Path Forest Algorithm: O Optimum-Path Forest para detecção de anomalias, denominado OPF-AD, foi proposto por Passos et al. (PASSOS et al., 2016) e preocupou-se com uma adaptação do OPF não supervisionado para a tarefa. Nesse contexto, o OPF não supervisionado é empregado para agrupar um conjunto de dados de treinamento composto apenas por amostras positivas. Em relação à etapa de teste, cada nova instância é associada ao cluster mais próximo usando uma abordagem baseada em k-NN, conforme realizado pelo OPF não supervisionado padrão. Além disso, a densidade dessa nova amostra é calculada e comparada com um limite, que é pré-definido como um hiperparâmetro. Por fim, a instância é considerada regular se a densidade for superior ao limite. Caso contrário, ele recebe um rótulo de anomalia.

Os códigos de cada classificador podem ser encontrados na biblioteca scikit-learn[3] ou na biblioteca PyOD[4], que é a biblioteca Python mais abrangente e escalável para detectar objetos periféricos em dados multivariados (ZHAO; NASRULLAH; LI, 2019).

Na área de detecção de anomalias, o trabalho de Wang, Bah e Hammad (2019), apresentam uma revisão do progresso dos métodos de detecção de outliers de 2000 a 2019. Enquanto Chalapathy, Menon e Chawla (2018), desenvolvem um modelo de rede neural de uma classe (OC-NN) para detectar anomalias em conjuntos de dados complexos, além de apresentarem uma visão abrangente dos métodos baseados em aprendizado profundo para detecção de anomalias e revisam a adoção de tais métodos em vários domínios e avaliam sua eficácia (CHALAPATHY; CHAWLA, 2019).

Já Mohaghegh e Abdurakhmanov (2021), expõem uma representação em nível de caractere de conjuntos de dados de texto não supervisionados para problemas de detecção de anomalias. Enquanto Pang et al. (2021), revisam 12 perspectivas de modelagem diversas sobre o aproveitamento de técnicas de aprendizado profundo para detecção de anomalias. Kannan et al. (2017), propõem um método de fatoração de matrizes, que é naturalmente capaz de distinguir as anomalias com o uso de aproximações de baixo escalão dos dados subjacentes. Ao mesmo tempo que Ruff et al. (2019), introduzem um novo método de detecção

[3] <https://scikit-learn.org/stable/>
[4] <https://pyod.readthedocs.io/en/latest/>

de anomalias—Context Vector Data Description (CVDD)—que se baseia em modelos de incorporação de palavras para aprender múltiplas representações de sentenças que capturam múltiplos contextos semânticos através do mecanismo de auto-atenção.

No Brasil, (CARVALHO et al., 2018), avaliam como técnicas de aprendizado de máquina podem ser empregadas na tarefa de identificar anomalias no comportamento das abelhas. Já Kintopp (2017), demonstra a aplicação da detecção de anomalias para encontrar anomalias nas informações de gastos divulgadas pelos municípios brasileiros, podendo assim apontar casos suspeitos de improbidade administrativa. No entanto, o uso de técnicas de detecção de anomalias para dados textuais não é comum no contexto brasileiro.

5 Metodologia

Para a realização da coleta dos dados, houve um processo de pesquisa abrangente sobre tudo que envolvia as *fake news* no cenário brasileiro. Como descrito anteriormente, encontrar bases de dados sobre este determinado assunto em português é extremamente díficil, pois grande parte das pesquisas nesta área estão relacionadas ao inglês. A partir de pesquisas bibliográficas, encontrou-se o Fake.Br Corpus que é o *dataset* brasileiro mais utilizado atualmente em pesquisas relacionadas a detecção de notícias falsas. Entretanto, com a grande evolução de algoritmos relacionados a *deep learning* e a grande necessidade de um número maior de amostras, a criação de uma nova base de dados foi discutida e implementada, deste modo, o FakeRecogna (GARCIA; AFONSO; PAPA, 2022), consiste de dados mais atualizados e com mais instâncias que as demais bases encontradas.

5.1 Fake.Br Corpus

O Fake.Br Corpus foi desenvolvido com base em alguns detalhes, são eles: possuem notícias falsas e verdadeiras, para que os métodos preditivos possam encontrar padrões e regularidades, o corpus possui um formato acessível para ferramentas de PLN, possuindo um tamanho homogêneo em relação as notícias para que ocorra uma normalização, possui vários âmbitos de categorias (Política, Esporte, Ciência, etc), as notícias foram coletadas com base em um certo período, pois pode ocorrer em uma mudança no estilo de escrita, informações extras, nome do autor, *link* da notícia, entre outros fatores (MONTEIRO et al., 2018).

A base de dados não possui uma grande quantidade, de acordo com Monteiro et al. (2018), as *fake news* foram coletadas a partir de vários sites, as notícias foram baixadas utilizando um crawler, e passaram por uma análise manual, garantindo que todas contivessem conteúdo falso. As notícias que utilizavam fatos verdadeiros para apoiar conclusões enganosas, não foram incluídas no corpus. A Tabela 1 apresenta os sites utilizados para a coleta das notícias falsas.

Site	**# Notícias**
Diário do Brasil	3.338
A Folha do Brasil	190
The Jornal Brasil	65
Top Five TV	7
Total	3.600

Tabela 1 – Número de notícias falsas coletadas em cada site.

No total, foram coletadas 3.600 notícias falsas e para garantir o balanceamento da base de dados, conforme Monteiro et al. (2018), foram coletadas notícias verdadeiras que fossem

similares às notícias coletadas anteriormente. Assim, foi desenvolvido um *crawler* que possuia a missão de buscar por palavras chaves que correspondiam a similaridade encontrada nas notícias falsas, buscando em portais como: G1[5], Folha de São Paulo[6] e Estadão[7], que são reconhecidos publicamente como veículos de notícias confiáveis. Após um período de coleta, foram utilizados cerca de 40.000 notícias verdadeiras, das quais, foram selecionadas manualmente e, também, com base na métrica da similaridade do cosseno (SALTON; MCGILL, 1986).

A distribuição de temas das notícias podem ser visualizados na Figura 8, ficando da seguinte forma: 58% das notícias coletadas falam sobre Política, $21,4\%$ são sobre TV e Celebridades, $17,7\%$ são sobre Sociedade e eventos do Dia a Dia; $1,5\%$ são sobre Ciência e Tecnologia; $0,7\%$ são sobre Economia; e $0,7\%$ são sobre Religião (MONTEIRO et al., 2018).

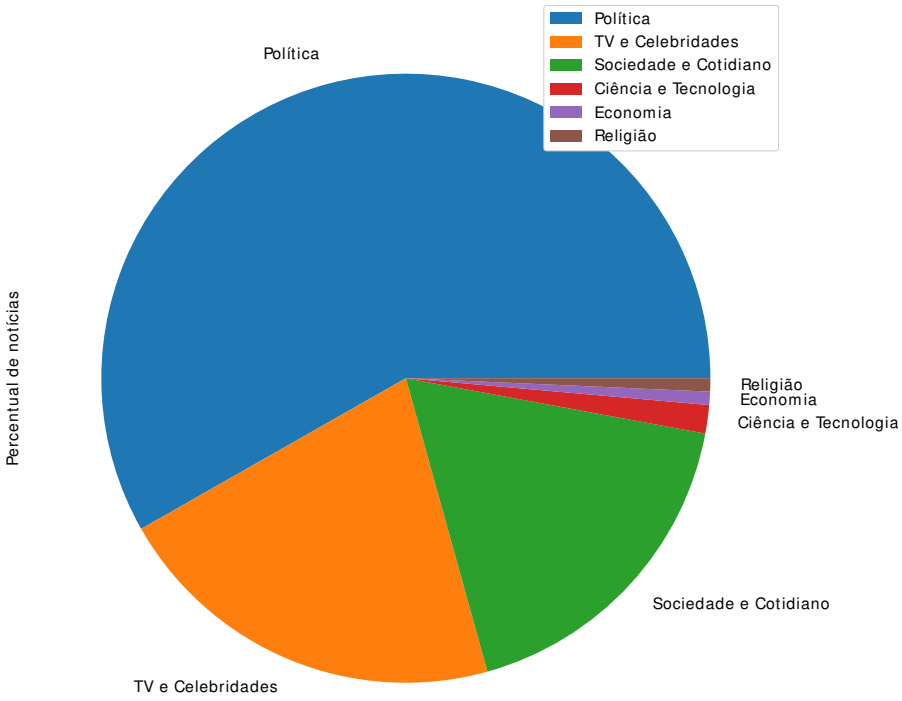


Figura 8 – Distribuição dos temas encontrados na base dados Fake.Br Corpus

Será disposto um exemplo dos textos encontrados na base de dados do Fake.Br Corpus, onde pode-se observar que mesmo a notícia sendo verdadeira não necessariamente ela desmente

[5] https://g1.globo.com/
[6] https://www.folha.uol.com.br/
[7] https://www.estadao.com.br/

a notícia falsa.

- Notícia Real: "*Michel Temer não quer o fim do Carnaval por 20 anos. Notícias falsas misturam proximidade dos festejos, crise econômica e medidas impopulares do governo do peemedebista.*"
- Notícia Falsa: "*Michel Temer propõe fim do carnaval por 20 anos, "PEC dos gastos". Michel Temer afirmou que não deve haver gastos com aparatos supérfluos sem pensar primeiramente na educação do Brasil. A medida pretende cancelar o carnaval de 2018.*"

Pode-se perceber que algumas *fake news* mascaram e sensacionalisam notícias reais, no primeiro caso, a notícia falsa realiza um sensacionalismo e, principalmente, introduz novas palavras e conceitos que denigrem a imagem do político.

5.2 FakeRecogna Corpus

FakeRecogna é um conjunto de dados composto por notícias reais e falsas. A notícia real não está diretamente ligada às notícias falsas e vice-versa, o que pode levar a uma classificação tendenciosa. A coleta de notícias foi realizada por *crawlers* desenvolvidos para minerar páginas de agências de renome e grande importância nacional, os *web crawlers* foram desenvolvidos com base em cada página analisada, onde os as informações são primeiro separadas em categorias e, posteriormente agrupadas por datas.

A pluralidade de notícias em várias páginas e os diferentes estilos de escrita proporcionam ao conjunto de dados uma grande diversidade para análise de processamento de linguagem natural e algoritmos de aprendizado de máquina. A busca considerou uma grande variedade de categorias que incluem Ciências, Entretenimento, Esportes, Política, Saúde e Brasil. A coleção de notícias falsas concentra-se principalmente nas páginas mencionadas pelo Duke Reporters Lab[8], que fornece uma lista de páginas em todo o mundo que verificam a veracidade das notícias. Haviam 160 agências verificadoras de fatos ativas no mundo em 2019. O Brasil tem um ecossistema em crescimento, com atualmente 9 iniciativas, conforme mostrado na Tabela 2.

Devido a restrições de acesso, foram considerados 6 das 9 páginas encontradas no Duke Reporters Lab, durante as buscas foram extraídas um grande número de notícias falsas de cada site, assim, o corpus é formado por diversas páginas e com várias formatações de escrita para não ocorrer nenhum viés de classificação. Ao total, foram extraídos 5.951 amostras, conforme mostrado na Tabela 3.

[8] https://reporterslab.org/fact-checking/

Agência	URL
AFP Checamos	<https://checamos.afp.com/afp-brasil>
Agência Lupa	<https://piaui.folha.uol.com.br/lupa>
Aos Fatos	<https://aosfatos.org>
Boatos.org	<https://boatos.org>
Estadão Verifica	<https://politica.estadao.com.br/blogs/estadao-verifica>
E-farsas	<https://e-farsas.com>
Fato ou Fake ("Fact or Fake")	<https://g1.globo.com/fato-ou-fake>
Projeto Comprova	<https://projetocomprova.com.br>
UOL Confere	<https://noticias.uol.com.br/confere>

Tabela 2 – Agências de checagem de fatos no Brasil.

Agência de Fact-checking	# Notícias
AFP Checamos	509
Agência Lupa	–
Aos Fatos	–
Boatos.org	2,605
Estadão Verifica	–
E-farsas	812
Fato ou Fake ("Fact or Fake")	1,055
Projeto Comprova	388
UOL Confere	582
Total	5.951

Tabela 3 – Número de notícias coletadas de cada agência de checagem de fatos.

Em relação às notícias reais, os crawlers pesquisaram portais como G1[9], UOL[10] e Extra[11], que são publicamente reconhecidos como veículos de notícias confiáveis, além do Ministério da Saúde do Brasil[12] página que resulta em um conjunto de mais de 100.000 amostras. Desse conjunto, foram filtradas amostras de 5.951 para manter o equilíbrio entre as classes. Ao final, a distribuição de temas das notícias podem ser visualizados na Figura 9.

O banco de dados é estruturado por 11.902 linhas com 8 colunas, onde cada linha corresponde a uma notícia real ou falsa. O conjunto de dados é armazenado em um arquivo XLSX de fácil leitura e manipulação. A Tabela 4 descreve os atributos encontrados em cada coluna do corpus.

Para melhor visualização, é disposto um exemplo dos textos encontrados na base de dados do FakeRecogna:

- Notícia Real: "*As escolas da rede estadual de São Paulo, com 3,5 milhões de alunos matriculados, finalizam hoje o ano letivo de 2020. As aulas recomeçam no dia 1º de*

[9] https://g1.globo.com/
[10] https://www.uol.com.br/
[11] https : //extra.globo.com/
[12] https://www.gov.br/saude/pt-br

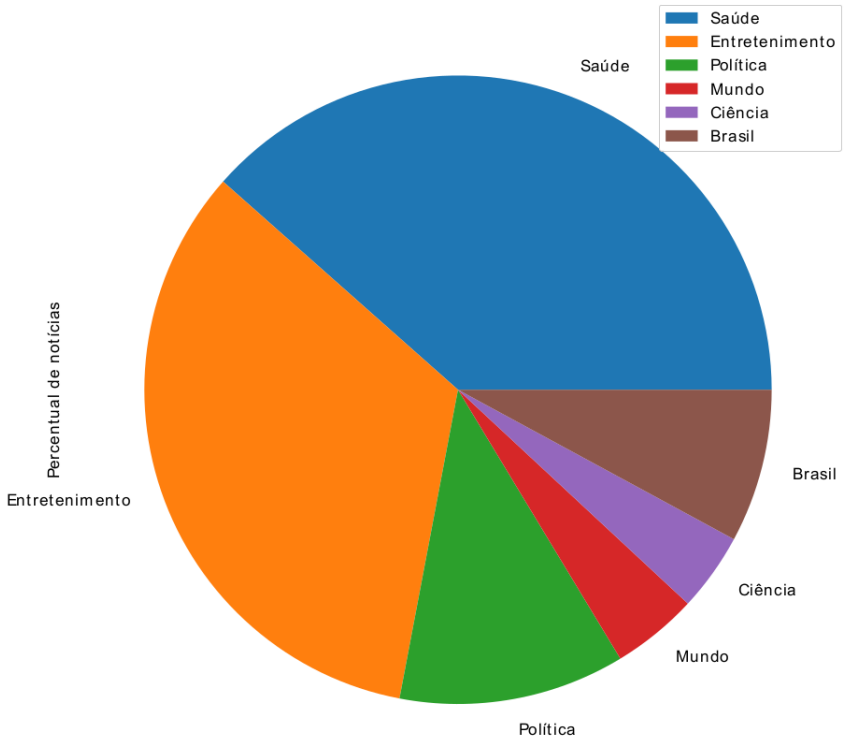


Figura 9 – Distribuição de notícias por categoria.

Coluna	Descrição
Título	Título do artigo
Sub-título (se disponível)	Breve descrição das notícias
Notícia	Informações sobre o artigo
Categoria	Notícias agrupadas de acordo com suas informações
Autor	Autor da publicação
Data	Data de publicação
URL	Endereço do artigo na web
Classe	0 para notícias falsas e 1 para notícias reais

Tabela 4 – Atributos do Dataset.

fevereiro do próximo, e alunos que não tiveram frequência mínima poderão realizar atividades de recuperação em janeiro. Em razão da pandemia do novo coronavírus, os estudantes tiveram de acompanhar as aulas remotamente este ano, por meio do Centro de Mídias SP. De acordo com a Secretaria da Educação, os alunos que entregaram as atividades propostas serão aprovados para o próximo ano letivo, mas terão o aprendizado avaliado ao final de 2021. Os anos letivos de 2020 e 2021 serão considerados como um único ciclo contínuo."

- Notícia Falsa: "*Vizinhos, boa tarde! Tive notícia de que estão aplicando a vacina da Phizer no Hebraica, para pessoas que tenham 50 anos ou mais e com comorbidades que apresentem receita médica de medicamento de uso contínuo. Divulguem, se possível, pq*

se não utilizarem terão que descartar. A informação é que tem 1.800 doses."

5.3 FakeRecogna Anomaly

O desenvolvimento do corpus FakeRecogna Anomaly como um conjunto de dados desbalanceado de notícias reais e falsas foi baseado no FakeRecogna Corpus (GARCIA; AFONSO; PAPA, 2022). A ideia principal do corpus FakeRecogna Anomaly é explorar uma situação da vida real onde o número de notícias reais é maior que o número de notícias falsas.

O corpus FakeRecogna Anomaly possui todos os atributos contidos no corpus FakeRecogna, mas com uma proporção de 1 notícia falsa para 100 notícias reais, assim, nosso corpus possuem 100.000 notícias verdadeiras e 1.000 notícias falsas. A coleta de notícias foi realizada por *crawlers* desenvolvidos para minerar páginas de notícias de agências conhecidas e de grande relevância nacional.

Seguindo o padrão estabelecido no corpus FakeRecogna, as principais agências de notícias utilizadas como base para extração de notícias foram G1[13], UOL[14] e Extra[15], que são reconhecidos publicamente como veículos de notícias confiáveis, além do Ministério da Saúde do Brasil[16]. Para a seleção das *fake news*, foram analisados os principais meios de *fact-checking* já extraídos no desenvolvimento do FakeRecogna e, optamos por extrair aleatoriamente com base na proporção de notícias de cada site. A tabela 5 apresenta as agências de notícias, bem como o número de notícias falsas coletadas de cada fonte.

Agência	**URL**	**# Notícias**
AFP Checamos	<https://checamos.afp.com/afp-brasil>	11
Boatos.org	<https://boatos.org>	504
E-farsas	<https://www.e-farsas.com>	167
Fato ou Fake ("Fact or Fake")	<https://oglobo.globo.com/fato-ou-fake>	110
Projeto Comprova	<https://projetocomprova.com.br>	91
UOL Confere	<https://noticias.uol.com.br/confere>	117
Total		$1,000$

Tabela 5 – As agências de verificação de fatos usadas na FakeRecogna Anomaly.

Diferentemente de FakeRecogna, cada amostra de FakeRecogna Anomaly tem campos de metadados 7 em vez de 8, a coluna "Sub-título" não foi colocada no conjunto de dados porque a grande maioria das notícias reais não possuem este atributo, assim, são descritos os metadados utilizados na Tabela 6.

Visando o problema de detecção de anomalias, o corpus FakeRecona Anomaly será a principal base para os experimentos envolvendo esta área de estudos.

[13] <https://g1.globo.com/>
[14] <https://www.uol.com.br/>
[15] <https://extra.globo.com/>
[16] <https://www.gov.br/saude/pt-br>

Coluna	Descrição
Título	Título do artigo
Notícia	Informações sobre o artigo
Categoria	Notícias agrupadas de acordo com suas informações
Autor	Autor da publicação
Data	Data da publicação
URL	Endereço do artigo na web
Classe	0 para notícias falsas e 1 para notícias reais

Tabela 6 – Metadados usados para descrever cada amostra.

Outra diferença entre os corpora fica em relação a distribuição das categorias, pois para a construção do FakeRecogna Anomaly foram atribuídos as categorias que possuiam mais informações, assim, a Tabela 7 apresenta a distribuição das notícias por categoria e sua respectiva quantidade.

Categoria	# Notícias
Brasil	$2,912$
Entretenimento	$20,081$
Saúde	$37,599$
Política	$33,311$
Ciência	$7,097$
Total	$101,000$

Tabela 7 – Distribuição de notícias por categoria na FakeRecogna Anomaly.

6 Experimentos e Resultados

Os experimentos foram realizados usando o protocolo de validação cruzada de 5 *folds*, na qual, segundo Géron (2019), é um método que divide aleatoriamente o conjunto de treinamento em 5 subconjuntos distintos chamados de partes (*folds*), então, realiza-se o treinamento e avaliação do modelo em 5 vezes, escolhendo uma parte (*fold*) diferente a cada execução para avaliação e as demais partes para treinamento. Assim, ao comparar a classificação de cada instância com a classe real, é possível obter métricas de desempenho, como por exemplo: acurácia, precisão e revocação. Nos experimentos realizados, a acurácia foi selecionada para medir o desempenho dos classificadores.

No trabalho trabalho desenvolvido por Monteiro et al. (2018), os primeiros testes foram realizados durante a construção do corpus, com 674 notícias, contendo 337 notícias falsas e 337 notícias verdadeiras, na qual todos os testes também foram utilizando o método de validação cruzada em 5 *folds*. Ambos datasets passaram por um pré-processamento de truncamento dos textos para evitar um viés na classificação. Para isso, foi analisada cada notícia, onde os maiores textos foram truncados no mesmo número de palavras da menor notícia, desta forma, evitando qualquer tendência de classificação errônea com base no tamanho dos textos.

6.1 Utilização da base Fake.Br Corpus com novas abordagens

Conforme Monteiro et al. (2018), com a base desenvolvida, houve a remoção de palavras com poucas ocorrências, além da exploração de duas outras técnicas de seleção de atributos: baseada em ganho de informação e baseada em Análise de Componentes Principais (do inglês *Principal Component Analysis* - PCA). Deste modo, foram elaborados novos métodos e abordagens para ganhar melhores desempenhos em relação aos trabalhos que envolvem esta base.

A validação cruzada manteve-se com 5 *folds*. Houve também a utilização do *GridSearch* nos classificadores padrões para maximizar o resultado final com base nos melhores parâmetros. Conforme Géron (2019), o *GridSearch* é um teste de hiperparâmetros que tenta encontrar uma ótima combinação de valores, ou seja, essa função é usada para encontrar os hiperparâmetros ideais de um modelo para que o resultado seja o melhor possivel.

Para a representação textual, foram abordados três novos métodos: BoW com TF-IDF, FastText e o BERTimbau. O FastText, é uma técnica baseada em redes neurais para computar representações de texto como o Word2Vec, que são vetores de alta dimensão onde cada entrada é uma medida de associação entre a palavra e o contexto onde ela está localizada (GOLDBERG Y.; LEVY, 2014). Nos testes, foram utilizados o modelo FastText, no qual, estende o modelo

Word2Vec que representa cada palavra como um *n-gram* de caracteres, ajudando assim a capturar o significado de palavras mais curtas e permite embeddings para entender sufixos e prefixos (REDDY, 2019). Os valores de parâmetro para fastText onde: tamanho de incorporação é igual a 200 dimensões, o número máximo de palavras exclusivas usadas é igual a 4.000 e a quantidade máxima de tokens para cada frase é igual a 1.000.

De acordo com Souza, Nogueira e Lotufo (2020), o BERTimbau, é um modelo BERT para Português, no qual, permite o pré-treinamento de representações de palavras bidirecionais profundas - em contraste com os trabalhos anteriores que empregam *Language Models* (LMs) unidirecionais - e tem muitas vantagens sobre modelos recorrentes bem estabelecidos, como LSTM e GRU. Os modelos BERTimbau possuem dois tamanhos: BERT-Base (12 camadas, 768 dimensões ocultas, 12 cabeças de atenção e 110 milhões de parâmetros) e BERT-Large (24 camadas, 1.024 dimensões ocultas, 16 cabeças de atenção e 330 milhões de parâmetros), na qual, foram treinados no *Brazilian Web as Corpus* (BrWaC), que é um grande corpus em português, para $1.000.000$ passos, usando várias técnicas mascaramento de palavras. Neste projeto, foi utilizado o modelo BERT-Base.

Os resultados apresentados com os classificadores: MLP, NB, RF e SVM utilizando apenas o método BoW foram retirados do trabalho de (MONTEIRO et al., 2018). Já os demais experimentos, foram desenvolvidos neste trabalho e podem ser visualizados na Tabela 8 com os melhores resultados de cada método destacados em negrito.

Classificador	Representação Textual	Precisão	Recall	F1-Score	Acurácia
MLP	BoW	-	-	-	$90,00\%$
MLP	BoW + TF-IDF	$92,20\%$	$96,90\%$	$94,50\%$	**94,20%**
NB	BoW	-	-	-	**87,00%**
NB	BoW + TF-IDF	$63,00\%$	**99,00%**	$77,03\%$	$70,60\%$
OPF	BoW	$90,10\%$	$96,00\%$	$92,96\%$	**92,30%**
OPF	BoW + TF-IDF	$85,20\%$	$94,00\%$	$89,38\%$	$87,80\%$
RF	BoW	-	-	-	$77,00\%$
RF	BoW + TF-IDF	**99,10%**	$80,20\%$	$88,70\%$	**89,50%**
SVM	BoW	-	-	-	$89,00\%$
SVM	BoW + TF-IDF	$89,20\%$	$97,20\%$	$93,00\%$	**92,70%**
CNN	FastText	$96,04\%$	$95,97\%$	**96,00%**	**95,72%**
BERT	BERTimbau	$94,70\%$	$94,70\%$	$94,70\%$	**94,55%**

Tabela 8 – Resultados médios de acurácia na base Fake.Br Corpus

No contexto geral, os resultados com o incremento de novas técnicas obtiveram melhores resultados, visto que os resultados obtidos pelo trabalho de Monteiro et al. (2018), utilizaram o método BoW. Deste modo, a aplicação de um experimento com um novo classificador (OPF) utilizando o mesmo método resultou em um grande resultado, obtendo a maior acurácia dentre os demais classificadores. A junção dos métodos *bag-of-words* com TF-IDF retornaram em geral, melhores valores em relação a abordagem utilizada separadamente, possuindo o

classificador MLP, o melhor resultado.

Com a abordagem envolvendo redes neurais e arquiteturas, os resultados obtidos foram com uma acurácia maior do que as demais técnicas, visto que o *word embedding* e o pré-treinamento com o BERTimbau são técnicas mais sofisticada comparada as demais, assim, os resultados alcançados foram satisfatórios e demonstram que essas novas técnicas/arquiteturas dão maiores resultados comparados a abordagens mais antigas.

6.1.1 Comparação de trabalhos relacionados a base Fake.Br Corpus

Após realizada uma vasta pesquisa bibliográfica referente a trabalhos relacionados a detecção de *fake news* na língua portuguesa que envolvem a base de dados *Fake.Br Corpus*, houve um refinamento para escolher aqueles que obtinham maiores detalhes de suas abordagens e conceitos aplicados. Assim, objetivou-se promover e comparar os respectivos resultados encontrados com base nestas diferentes metodologias desenvolvidas pelos autores.

Desta maneira, o trabalho desenvolvido por Okano (2020), em sua primeira abordagem, propõem a utilização de *character n-gram*, que pode ser definido como uma maneira de representar os dados do texto para algoritmos de aprendizagem de máquina. Neste modelo de representação textual, cada documento é representado por uma série de *n-gram*, na qual, um *n-gram* é uma sequência contínua de *n* itens. Também é aplicado um ajuste nos parâmetros do dicionário gerado pelo *char n-gram*, no qual, faz um corte baseado na frequência dos *char 3-gram* utilizando os parâmetros *no below*, que remove os itens do dicionário que apareceram em menos de *n* itens na fase de treinamento e o parâmetro *not above*, que remove os itens do dicionário que aparecem em mais de $n\%$ dos textos da fase de treinamento.

Já sua segunda abordagem de experimentos é baseada na meta-análise escrita por (HAUCH et al., 2015), em que os autores investigaram 79 atributos linguísticos extraídos de 44 estudos acadêmicos. Entretanto, Okano (2020) utilizou 63 dos 79 atributos, pois, alguns destes atributos linguísticos não foram implementados por falta de recursos de PLN na língua Portuguesa. Em seu trabalho mais recente, Okano et al. (2020) utiliza uma camada de atenção, no qual o vetor de sentença pode dar mais distinção para as palavras mais "importantes" para a sentença. O modelo de atenção, é composto por:

- *Gated Recurrent Unit* (GRU): É um mecanismo utilizado em redes neurais recorrentes que consegue processar as informações obtidas levando em conta a sequência das palavras utilizadas.
- *Word Encoder*: Composta pelo que corresponde ao número máximo de palavras por sentença, na qual é transformada em vetor utilizando um algoritmo de aprendizagem não supervisionado para obter representações vetoriais de palavras, denonimado GloVe (HARTMANN et al., 2017).

- *Word Attention*: Calcula pesos para palavras considerando a importância das palavras para a sentença, ou seja, calcula a importância das palavras para aquela determinada frase.
- *Sentence Encoder*: Assim como na camada *Word Encoder* utilizando o GRU bidirecional para processar o vetor de sentenças.
- *Sentence Attention*: Um mecanismo de atenção semelhante a camada *Word Attention*, na qual, é implementada para dar mais importância as sentenças que mais contribuem para a classificação.
- Classificação: Classifica o documento, utilizando o vetor do documento e uma função *SoftMax*, na qual, é uma função usada para forçar a saída de uma rede neural a representar a probabilidade dos dados serem de uma das classes definidas.

Conforme Okano et al. (2020), na camada de *word embedding*, foram utilizados os vetores GloVe treinados por (HARTMANN et al., 2017). Assim, foram utilizados vetores GloVes de diferentes dimensões (100, 300 e 600), entretanto, a dimensão que obteve o melhor resultado foi com o valor 600.

Outro trabalho desenvolvido por Faustini e Covoes (2019), propõe detectar notícias falsas por meio de um modelo de treinamento apenas com amostras falsas no conjunto de dados de treinamento, por meio do *One-Class Classification* (OCC). Assim, com base no projeto elaborado por (FERREIRA; MEDEIROS; FRANCA, 2018), que propõem um algoritmo para reduzir a dimensionalidade denominado *Document-Class Distance* (DCDistance), o projeto visa modificá-lo e transformá-lo de um algoritmo de redução de dimensionalidade em um classificador de uma classe, denominado DCDistanceOCC.

De acordo com o projeto de Santos et al. (2020), é apresentado um estudo sobre o impacto dos recursos de legibilidade na detecção de notícias falsas para o português brasileiro. Baseando-se na ferramenta de extração de recursos denominada Coh-Metrix, utilizada para tirar métricas de coesão e coerência de um texto proposta por (MCNAMARA et al., 2014). Assim, para elaborar sua pesquisa, houve uma exploração dos recursos de legibilidade fornecidos por uma versão brasileira da ferramenta, denominada Coh-Metrix-Port e Coh-Metrix-Dementia (CUNHA, 2015), a utilização do software AIC2 (MAZIEIRO; PARDO; ALUISIO, 2008), além de recursos psicolinguísticos.

Assim, nestes trabalhos difundidos, existem diversas abordagens que promoveram grandes resultados, nos quais, estão dispostos na Tabela 9, com os respectivos resultados finais.

Vale observar que os dados demonstrados na Tabela 9, são experimentos nos quais houveram um pré-processamento dos textos brutos, assim, minimizando qualquer tendência ou viés de classificação. Conforme Okano et al. (2020), ao utilizar o texto bruto seu experimento

Autor	Classificador	Protocolo	Método	Acurácia
Monteiro	SVM	5 *folds*	BOW	$89,00\%$
Okano	SVM	5 *folds*	char n-gram = 3 + Refinamento	$92,32\%$
Okano	SVM	5 *folds*	Características Psicolinguísticas	$75,00\%$
Okano	Hierarchical Attention Network	5 *folds*	GloVe 600	$90,94\%$
Faustini	DCDistanceOCC	10 *folds*	One-Class Classification	$67,00\%$
Santos	SVM	5 *folds*	Coh-Metrix-Port e Coh-Metrix-Dementia.	$93,00\%$
Garcia	CNN	5 *folds*	FastText	$95,72\%$

Tabela 9 – Resultados de experimentos relacionados a base Fake.Br Corpus.

alcançou 96% de acurácia, entretanto o que pode distorcer os resultados, uma vez que as notícias reais possuem um texto maior do que as falsas.

6.2 Experimentos realizados na base FakeRecogna

Para a realização dos experimentos, foram realizados alguns pré-processamentos, divididos em quatro etapas:

1. Truncamento: esta etapa é essencial para evitar qualquer viés de classificação devido à variação significativa de tamanho entre as sentenças, principalmente entre as falsas e reais. Os últimos tendem a ser muito mais longos do que os primeiros.

2. Padronização de termos: retirada de palavras que possam enviesar a notícia, como, "enganoso", "boato", "fake" e, assim por diante. Pontuação, caracteres especiais e URLs também foram removidos e a padronização para letras minúsculas.

3. Lematização / Tokenização: foram realizados dois experimentos utilizando as duas técnicas. A lematização consiste em identificar a lematização de uma palavra, ou seja, a lematização compreende a análise morfológica das palavras resultando em sua forma canônica (HIPPISLEY, 2010). A tokenização, também conhecida como segmentação de palavras, quebra a sequência de caracteres em um texto, encontrando o limite de cada palavra, ou seja, os pontos onde uma palavra termina e outra começa (PALMER, 2010).

4. Remoção de palavras irrelevantes: remoção de palavras consideradas irrelevantes para a compreensão da notícia, como artigos e preposições.

Para a utilização das representações textuais, foram utilizados dois métodos: o BoW e o FastText. Assim, como nos experimentos realizados na Fake.Br Corpus, a Tabela 10 apresenta os resultados médios para cada técnica de representação e classificação de texto, com os melhores resultados em negrito.

Classificadores	Precisão		Recall		F1-Score		Acurácia	
	BoW	FastText	BoW	FastText	BoW	FastText	BoW	FastText
MLP	**0.931**	**0.850**	**0.931**	**0.848**	**0.930**	**0.848**	**93.1%**	**84.8%**
NB	0.896	0.712	0.898	0.680	0.897	0.666	89.7%	68.1%
OPF	0.834	0.784	0.834	0.784	0.834	0.782	83.4%	78.4%
RF	0.924	0.840	0.922	0.840	0.922	0.840	92.3%	84.0%
SVM	0.926	0.832	0.925	0.810	0.926	0.804	92.5%	80.8%
CNN	-	**0.942**	-	**0.942**	-	**0.942**	-	**94.28%**

Tabela 10 – Resultados experimentais sobre FakeRecogna Corpus.

Uma análise mais aprofundada nos mostra que o melhor resultado com o BoW foi alcançado pelo classificador MLP, seguido pelo RandomForest e SVM, com resultados que ultrapassaram mais de 90% de acurácia. Os resultados mostram que mesmo utilizando uma técnica padrão de processamento de linguagem natural, ou seja, BoW, os resultados alcançados foram muito interessantes.

Por outro lado, usando uma arquitetura de pré-processamento de texto mais robusta como FastText e a Rede Neural Convolucional, os melhores resultados superaram 94%. FastText foi considerado por apresentar diferenças em relação a outros métodos de processamento, como por exemplo, o Skip-Gram, pois este método ignora a morfologia da palavra atribuindo um vetor distinto a cada palavra, limitando assim as representações a palavras raras, tornando a incorporação de palavras menos eficiente. Entretanto, o FastText modela a morfologia das palavras, considerando sua subestrutura e modelando seus fragmentos como *n-grams*, permitindo representações de palavras que eventualmente não apareceriam no conjunto de aprendizagem. O modelo FastText também apresenta um desempenho satisfatório em termos de tempo de pré-processamento e treinamento de texto.

6.3 Experimentos realizados na base FakeRecogna Anomaly

Na base FakeRecogna Anomaly a etapa de pré-processamento foi composta pelas etapas executadas nos experimentos realizados pelo FakeRecogna Corpus, sendo seu único diferencial a não utilização do processo de padronização dos termos. Já em relação as representações textuais, foram utilizadas três representações para cada texto e elas não foram combinadas, assim, a ideia principal foi realizar a detecção usando diferentes representações.

Os classificadores são treinados usando 80% das notícias reais do corpus enquanto o conjunto de teste é composto pelos 20% restantes das notícias reais mais todas as notícias falsas, Tabela 11. O desempenho é avaliado por meio de três métricas: (i) f1-score micro, (ii) f1-score macro e (iii) precisão.

Conforme mencionado, cada classificador é avaliado usando três representações de texto como entrada para analisar seu impacto na detecção e os resultados experimentais usando

Divisão	Tipos de notícia	# Amostras
Treino	Real	$80,000$
Teste	Real e Falso	$21,000$

Tabela 11 – Detalhes de cada conjunto experimental.

FakeRecogna Anomaly, a Tabela 12 apresenta os resultados para cada técnica de representação e detecção de anomalias com os melhores resultados em negrito.

Classificador	**Representação Textual**	**F1-score**		**Precisão**	
		micro	**macro**	**real**	**fake**
ECOD	BoW	0.866	0.522	96.0%	8.0%
	TF-IDF	0.869	0.519	96.0%	8.0%
	FastText	0.867	0.519	96.0%	8.0%
LOF	BoW	0.904	**0.703**	99.0%	31.0%
	TF-IDF	**0.928**	0.517	95.0%	9.0%
	FastText	0.866	0.500	95.0%	5.0%
CBLOF	BoW	0.863	0.526	96.0%	9.0%
	TF-IDF	0.902	**0.717**	100.0%	32.0%
	FastText	0.866	0.501	95.0%	5.0%
OneClassSVM	BoW	0.488	0.358	94.0%	4.0%
	TF-IDF	0.519	0.413	100.0%	9.0%
	FastText	0.494	0.354	94.0%	3.0%
SGDOneClassSVM	BoW	0.591	0.457	100.0%	10.0%
	TF-IDF	0.518	0.413	100.0%	9.0%
	FastText	**0.951**	0.491	95.0%	9.0%
IsolationForest	BoW	**0.912**	0.548	96.0%	13.0%
	TF-IDF	0.912	0.515	95.0%	8.0%
	FastText	0.909	**0.522**	95.0%	9.0%
OPF-AD	BoW	0.056	0.054	99.0%	5.0%
	TF-IDF	0.069	0.068	100.0%	5.0%
	FastText	0.049	0.047	97.0%	5.0%

Tabela 12 – Resultados experimentais sobre FakeRecogna Anomaly.

Com relação à representação de texto, o TF-IDF apresentou os melhores resultados ao considerar a precisão. O fato de usarmos a lematização na fase de pré-processamento pode ter prejudicado o FastText, pois ele trabalha com n-gramas, prefixos e sufixos, que são informações que podem ser perdidas quando as palavras são lematizadas. Além da precisão, quando analisamos uma base desbalanceada, uma das principais métricas é o f1-score macro, na qual, o classificador CBLOF com a representação TF-IDF conseguiram a maior média dentre os outros experimentos, seguidos do LOF com a representação BoW e, bem mais abaixo, IsolationForest também com o BoW.

Em relação aos modelos, os métodos de detecção de outliers baseados em proximidade obtiveram os melhores resultados para identificar notícias reais e falsas. Vale destacar ainda dois

outros pontos: (i) os modelos baseados em proximidade para detecção de outliers obtiveram melhores resultados em relação às outras técnicas, e (ii) o classificador ECOD não apresentou nenhuma grande mudança de desempenho para nenhuma das representações textuais.

6.3.1 Experimentos de comparação: Classificador padrão x Detecção de anomalia

Nesta seção será descrita a comparação dos resultados obtidos através de experimentos para averiguar como se comporta a base de dados FakeRecogna Anomaly em classificadores supervisionados e não supervisionados. Todos os testes foram utilizando os classificadores já mencionados anteriormente e sua representação textual foi o BoW. Entretanto, neste novo experimento foi acrescido uma normalização, denominada MinMaxScaler [17], uma técnica de pré-processamento que coloca os dados na mesma escala, ou seja, transforma as features distruibindo os dados em um intervalo, por padrão 0 a 1.

Para a realização desta comparação, criamos novamente folds diferentes dos testes apresentados anteriormente e, assim, reproduzir novamente todos os processos já descritos. Nesta nova etapa de experimentos, visamos as métricas f1-score micro, f1-score macro e precisão. A f1-score micro calcula uma pontuação f1 global contando as somas dos verdadeiros positivos (TP), falsos negativos (FN) e falsos positivos (FP). Já pontuação f1-score macro é calculada usando a média aritmética (também conhecida como média não ponderada) de todas as pontuações f1 por classe, assim, esse método trata todas as classes igualmente, independentemente de seus valores de suporte. A utilização destas métricas deve-se principalmente por conta da base ser desbalanceada, pois o meio de comparação dos resultados envolvendo outra métrica, por exemplo a acurácia, poderia não ser exata ou compreensível pela desproporção dos dados. A Tabela 13, mostra os resultados obtidos com os classificadores citados anteriormente, sendo a tabela dividida primeiro com os algoritmos supervisionados e logo abaixo, os algoritmos não supervisionados.

Os resultados obtidos mostram que mesmo para uma base desbalanceada, classificadores balanceados conseguem um bom desempenho. Entretanto, a utilização de algoritmos não supervisionados possuíram um melhor desempenho analisando as métricas aplicadas, pois, a pontuação f1-score micro dá igual importância a cada observação, ou seja, quando as classes estão desequilibradas, aquelas classes com mais observações terão um impacto maior na pontuação final, resultando em uma pontuação que pode esconder o desempenho das classes minoritárias e amplifica a maioria.

Por outro lado, a pontuação F1-macro dá igual importância a cada classe, portanto, uma classe majoritária contribuirá igualmente para a minoria, permitindo que a pontuação f1-macro ainda retorne resultados objetivos em conjuntos de dados não balanceados. Em suma,

[17] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

Classificador	F1-score		Precisão	
	micro	macro	real	fake
LinearSVC	0.993	0.498	99.0%	0.0%
Naive Bays	0.990	0.499	99.0%	0.0%
SVM	0.990	0.498	99.0%	25.0%
OPF	0.983	0.575	99.0%	15.0%
RandomForest	0.990	0.418	96.0%	0.52%
ECOD [NB]	0.331	0.285	97.0%	5.0%
IsolationForest [RF]	0.931	0.497	95.0%	2.5%
LOF [NB]	0.619	0.413	94.0%	3.0%
OneClassSVM [SVC]	0.931	0.502	100.0%	32.0%
OPF-AD [OPF]	0.070	0.069	100.0%	0.05%
SGDOneClassSVM [SVM]	0.952	0.502	95.0%	0.0%

Tabela 13 – Resultados de aprendizados diferentes no corpus FakeRecogna Anomaly.

se analisarmos os resultados, podemos ver que os algoritmos não supervisionados obtiveram um melhor resultado envolvendo o conjunto de dados FakeRecogna Anomaly do que os algoritmos supervisionados, demonstrando assim que a detecção de fake news pode ser tratada como um problema de detecção de anomalias.

7 Conclusão e Trabalhos Futuros

Uma das preocupações está justamente em coleta de dados, visto que, a base de dados de notícias falsas e verdadeiras na língua portuguesa é precária de amostras, diferentemente de outros idiomas, o inglês por exemplo. Assim, o projeto está em fase de coleta de novos dados para que ocorra uma nova formalização e limpeza, para assim, realizar novos experimentos.

O FakeRecogna é uma proposta de um novo corpus para detecção de notícias falsas em português do Brasil, no qual, o corpus apresenta uma série de vantagens em relação aos demais corpus existentes, que incluem:

- Notícias atualizadas (2019-2021);
- Maior número de categorias;
- Maior número de notícias;
- Ele cobre questões atuais, como a pandemia Covid-19;
- Várias fontes;
- Fontes reconhecidas internacionalmente.

Com a construção dos conjuntos de dados, espera-se fomentar a pesquisa sobre a detecção de notícias falsas em textos em português, uma vez que o baixo número de amostras disponíveis direciona a maioria dos trabalhos para conjuntos de dados baseados em inglês. Além disso, os experimentos obtiveram resultados interessantes considerando a dificuldade do problema. Foram empregadas novas técnicas de representação textual e sete classificadores em ambas base de dados.

Já o corpus FakeRecogna Anomaly possui a proposta de contextualizar com o mundo real, na qual, as notícias falsas são muito menores que as notícias reais. Assim, o corpus apresenta uma vista mais próxima da realidade em relação aos demais corpus existentes, pois em sua grande maioria, corpus que trabalham com a detecção de notícias falsas são balanceados.

Foram realizados experimentos com representações textuais clássicas e diferentes classificadores abrangendo diferentes métodos de trabalho, nos quais conseguimos produzir resultados interessantes considerando a dificuldade do problema. Com a construção do conjunto de dados, esperamos incentivar pesquisas sobre detecção de anomalias envolvendo fake news em português, pois, até onde se sabe, não foram encontradas bases de dados específicas que apresentem a proporção desenvolvida neste trabalho, de acordo com suas características de formação de corpus com 1 notícias falsas para cada 1.000 notícias reais, provenientes de

agências reconhecidas internacionalmente e que têm suas notícias atualizadas em comparação com outras bases de dados.

Também foram executados experimentos utilizando classificadores projetados para o problema de detecção de outliers onde os modelos obtiveram resultados interessantes, além da comparação de desempenho envolvendo algoritmos supervisionados e não supervisionados.

A principal ideia da contrução do FakeRecogna Anomaly é expandir a discussão de detecção de notícias falsas para novas ramos da inteligência artificial, assim, trabalhando com detecção de anomalias e técnicas para desenvolver ainda mais este problema. Possuindo como base o corpus FakeRecogna, o corpus de anomalias pode ser o maior corpus sobre fake news em português atualmente, com isso, o desenvolvimento de novos experimentos envolvendo a base, novas técnicas de representação textual, novos classificadores, novas metodologias, outras bases, junções e impactos de treinamento/teste podem ser abordados, contribuindo para o progresso da comunidade em uma evolução para entender ainda mais este problema atual.

Considerando estes resultados, e analisando-os sob à óptica das perguntas e da hipótese de pesquisa apresentada na introdução, as principais conclusões obtidas neste trabalho foram:

- Desenvolver novas bases de dados com notícias que são vinculadas no dia a dia por agências reconhecidas mundialmente, mantendo sempre as notícias atualizadas e correspondendo com a realidade brasileira é muito importante para o desenvolvimento desta área no Brasil;
- Provar que os corpus desenvolvidos podem trazer muitos benefícios para a comunidade, buscando salientar e instigar novas abordagens e metodologias nesta área de pesquisa;
- Buscar uma nova metodologia, visto que em grande parte, as bases de dados sobre detecção de notícias falsas são balanceadas, assim, propondo uma nova base de dados que se aproximasse mais da realidadea atual;
- Modelar o problema de identificação de fake news como sendo uma tarefa de detecção de anomalias para desenvolver uma nova abordagem no cenário atual.

Os experimentos apresentados nesta dissertação ainda não estão com os melhores resultados possíveis, visto que a principal motivação é a fomentação desta área de pesquisa com novas abordagens na língua portuguesa. Entretanto, considerando os resultados, pode-se concluir que os corpus desenvolvidos podem auxiliar, além de estimular novas pesquisas envolvendo técnicas de processamento de linguagem natural e algoritmos de aprendizado de máquina.

Referências

ABIRAMI, S.; CHITRA, P. Chapter fourteen - energy-efficient edge based real-time healthcare support system. In: RAJ, P.; EVANGELINE, P. (Ed.). *The Digital Twin Paradigm for Smarter Systems and Environments: The Industry Use Cases*. [S.l.]: Elsevier, 2020, (Advances in Computers, 1). p. 339–368.

AHO, A. V. et al. Compilers: Principles, techniques, and tools, second edition. *Addison-Wesley Longman Publishing Company*, Hardcover, 2006. Acesso em: 2 de Agosto de 2020. Disponível em: <https://www.amazon.com/Compilers-Principles-Techniques-Tools-2nd/dp/0321486811>.

ALLCOTT, H.; GENTZKOW, M. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, v. 31, p. 211–236, 2017.

ALPAYDIN, E. Introduction to machine learnin. *MIT Press*, 2010.

ARANHA, C. N. Uma abordagem de pré-processamento automático para mineração de textos em português: Sob o enfoque da inteligência computacional. *Tese (Doutorado em Engenharia Elétrica) Pontifica Universidade Católica do Rio de Janeiro*, 2007.

BARBOSA, J. L. N. et al. Introdução ao processamento de linguagem natural usando python. *Escola Regional de Informática do Piauí*, Livro Anais - Artigos e Minicursos, v. 1, p. 336–360, 2017. Acesso em: 1 de Agosto de 2020. Disponível em: <http://www.facom.ufu.br/~wendelmelo/terceiros/tutorial_nltk.pdf>.

BERRY, M. W.; MOHAMED, A.; YAP, B. W. *Supervised and unsupervised learning for data science*. [S.l.]: Springer, 2019.

BIRD, S.; KLEIN, E.; LOPER, E. *Natural Language Processing with Python*. [S.l.]: O'Reilly Media, 2009.

BOND, C. J.; DEOPAULO, B. M. Individual differences in judging deception: Accuracy and bias. *Psychological Bulletin*, v. 134, p. 477–492, 2008.

BRINK, H.; RICHARDS, J. Real world machine learning. *Manning Publications*, 2014.

BULEGON, H.; MORO, C. M. C. Mineração de texto e o processamento de linguagem natural em sumários de alta hospitalar. *Journal of Health Informatics*, 2010.

BURFOOT, C.; BALDWIN, T. Automatic satire detection: Are you having a laugh? *Association For Computational Linguistics*, ACL-IJCNLP, p. 161–164, 2009.

CAMPBELL, M. W. Support vector machines for speaker and language recognition. *Computer Speech & Language*, v. 20, n. 2-3, p. 210–229, 2006.

CANTRIL, H. The invasion from mars. *NJ: Princeton University Press*, 2005.

CARVALHO, H. V. F. de et al. Detecção de anomalias em comportamento de abelhas utilizando redes neurais recorrentes. In: *Anais do IX Workshop de Computação Aplicada a Gestão do Meio Ambiente e Recursos Naturais*. Porto Alegre, RS, Brasil: SBC, 2018. ISSN 2595-6124.

CARVALHO, R. L. V. R. Notícias falsas ou propaganda?: Uma análise do estado da arte do conceito fake news. *Revista de Epistemologias da Comunicação*, v. 7, 2009.

CHALAPATHY, R.; CHAWLA, S. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

CHALAPATHY, R.; MENON, A. K.; CHAWLA, S. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

CHAUDIRON, S. Technologies linguistiques et modes de représentation de l'information textuelle. Documentaliste - sciences de l'information, v. 44, n. 1, p. 30–39, 2007.

CHENG, Z.; ZOU, C.; DONG, J. Outlier detection using isolation forest and local outlier factor. In: *Proceedings of the conference on research in adaptive and convergent systems*. [S.l.: s.n.], 2019. p. 161–168.

CORDEIRO, B. C. Bert e word2vec: Uma anÁlise inferencial e redes neurais convolucionais. *Universidade Federal do Rio de Janeiro*, 2019. Disponível em: <http://repositorio.poli.ufrj.br/monografias/monopoli10029908.pdf>.

CRUSE, A. Meaning in language: an introduction to semantics and pragmatics. New York: Oxford University Press, 2011.

CUNHA, A. L. V. Coh-metrix-dementia: Análise automática de distúrbios de linguagem nas demncias utilizando processamento de línguas naturais. *Universidade de São Paulo*, 2015.

DALE, R. *Classical approaches to natural language processing*. [S.l.]: Chapman Hall/CRC, 2010.

DALTO, M. Deep neural networks for time series prediction with applications in ultra-short-term wind forecasting. In: . [S.l.: s.n.], 2014.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *Conference on Neural Information Processing Systems, NIPS*, 2018. Acessado em 28 de maio de 2020. Disponível em: <https://arxiv.org/pdf/1810.04805.pdf>.

ERJAVEC, T.; DZEROSKI, S. Machine learning of morphosyntactic structure: lemmatizing unknown slovene words. *Applied artificial intelligence*, v. 18, p. 17–41, 2004.

FALCÃO, A.; STOLFI, J.; LOTUFO, R. The image foresting transform: Theory, algorithms, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, p. 26(1):19–29, 2004.

FAUSTINI, P.; COVOES, T. F. Fake news detection using one-class classification. *Brazilian Conference on Intelligent Systems, BRACIS*, 2019. DOI: 10.1109/bracis.2019.00109.

FERREIRA, A.; GIRALDI, G. Convolutional neural network approaches to granite tiles classification. *Expert Systems with Applications*, Elsevier, v. 84, p. 1–11, 2017.

FERREIRA, C.; MEDEIROS, D. de; FRANCA, F. de. Dcdistance: A supervised text document feature extraction based on class labels. *CoRR*, 2018. Abs/1801.04554.

G1; PORTAL, D. N. É fake que morte de traficante em tiroteio foi noticiada como se fosse por covid-19. 2020. Acesso em: 01 de Outubro de 2020. Disponível em: <https://g1.globo.com/fato-ou-fake/noticia/2020/03/31/e-fake-que-morte-de-traficante-em-tiroteio-foi-noticiada-como-se-fosse-por-covid-1title9>.

GARCIA, G. L.; AFONSO, L.; PAPA, J. P. Fakerecogna: A new brazilian corpus for fake news detection. In: SPRINGER. *International Conference on Computational Processing of the Portuguese Language*. [S.l.], 2022. p. 57–67.

GARCIA, M.; GAMALHO, P. Análise morfossintáctica para português europeu e galego: problemas, soluções e avaliação. *Linguamática*, v. 2, n. 2, p. 59–67, 2010.

GASPERIN, C. V.; LIMA, V. L. S. de. Fundamentos do processamento estatístico da linguagem natural. *Faculdade de Informática – PUCRS*, Technical Report Series, 2016. Acesso em: 1 de Agosto de 2020. Disponível em: <https://www.pucrs.br/facin-prov/wp-content/uploads/sites/19/2016/03/tr021.pdf>.

GÉRON, A. *Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow*. [S.l.]: O'Reilly Media, 2019.

GODDARD, C.; SCHALLEY, A. C. *Syntactic analysis*. 2. ed. [S.l.]: Handbook of natural language processing, 2010.

GOLDBERG, Y. Neural network methods for natural language processing. *Synthesis lectures on human language technologies*, Morgan & Claypool Publishers, v. 10, n. 1, p. 1–309, 2017.

GOLDBERG, Y.; HIRST, G. *Neural Network Methods in Natural Language Processing*. [S.l.]: Morgan amp; Claypool Publishers, 2017. ISBN 1627052984.

GOLDBERG Y.; LEVY, O. Word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *Conference on Neural Information Processing Systems, NIPS*, 2014. Acessado em 28 de maio de 2020. Disponível em: <https://arxiv.org/pdf/1402.3722.pdf>.

GOMES, C. A. Os 7 tipos de fake news sobre a covid-19. 2020. Acesso em: 01 de Outubro de 2020. Disponível em: <https://www.blogs.unicamp.br/covid-19/os-7-tipos-de-fake-news-sobre-a-covid-19/>.

GOYAL, P.; PANDEY, S.; JAIN, K. *Deep Learning for Natural Language Processing: Creating Neural Networks with Python*. Apress, 2018. ISBN 9781484236857. Disponível em: <https://books.google.com.br/books?id=Jy9iDwAAQBAJ>.

GUIMARÃES, L. M. S.; MEIRELES, M. R. G.; ALMEIDA, P. E. M. d. Avaliação das etapas de pré-processamento e de treinamento em algoritmos de classificação de textos no contexto da recuperação da informação. *Perspectivas em Ciência da Informação*, SciELO Brasil, v. 24, p. 169–190, 2019.

GÜNGÖR, T. *Part-of-speech tagging*. 2. ed. [S.l.]: Handbook of natural language processing, 2010.

HARTMANN, N. S. et al. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In: *Anais do XI Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*. Porto Alegre, RS, Brasil: SBC, 2017. p. 122–131. Disponível em: <https://sol.sbc.org.br/index.php/stil/article/view/4008>.

HAUCH, V. et al. Are computers effective lie detectors? a meta-analysis of linguistic cues to deception. *Personality and Social Psychology Review*, v. 19, p. 307–342, 2015. Acessado em 24 de novembro de 2020. Disponível em: <http://journals.sagepub.com/doi/10.1177/1088868314556539>.

HAYKIN, S.; ENGEL, P. Redes neurais: princípios e prática. In: . Artmed, 2007. Disponível em: <https://books.google.com.br/books?id=vV7uzgEACAAJ>.

HE, Z.; XU, X.; DENG, S. Discovering cluster-based local outliers. *Pattern recognition letters*, Elsevier, v. 24, n. 9-10, p. 1641–1650, 2003.

HIJAZI, S.; KUMAR, R.; ROWEN, C. Using convolutional neural networks for image recognition by. In: . [S.l.: s.n.], 2015.

HIPPISLEY, A. Lexical analysis. In: INDURKHYA, N.; DAMERAU, F. J. (Ed.). *Handbook of Natural Language Processing, Second Edition*. [S.l.]: Chapman and Hall/CRC, 2010. p. 31–58.

INDURKHYA; DAMERAU. *Handbook of Natural Language Processing*. [S.l.]: Chapman Hall/CRC, 2010.

JUNIOR, A. T. da C. Processamento de linguagem natural para indexação automática semântico-ontológica. *Universidade de Brasília*, 2013.

JURAFSKY, D.; MARTIN, J. H. Speech and language processing. *Prentice Hall*, 2019. Acesso em: 7 de Agosto de 2020. Disponível em: <https://web.stanford.edu/~jurafsky/slp3/>.

KANNAN, R. et al. Outlier detection for text data. In: SIAM. *Proceedings of the 2017 siam international conference on data mining*. [S.l.], 2017. p. 489–497.

KINTOPP, P. M. *Aplicação de técnicas de aprendizado de máquina em dados públicos para detecção de anomalias*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2017.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. v. 25. Disponível em: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998.

LI, Z. et al. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, 2022.

LIMA, M. et al. Uso de redes neurais artificiais (rna) do tipo multilayer perceptrons (mlp) modificado com processamento estatístico em paralelo para estudo do problema de classificação da origem de vinho tinto. *Revista Brasileira de Agropecuária Sustentável*, v. 6, 06 2016.

LIU, B.; ZHANG, L. A survey of opinion mining and sentiment analysis. *Mining Text Data*, Springer US, p. 415–463, 2012. Acesso em: 2 de Agosto de 2020. Disponível em: <https://doi.org/10.1007/978-1-4614-3223-4_13>.

LJUNGLÖF, P.; WIRÉN, M. *Syntactic parsing*. 2. ed. [S.l.]: Handbook of natural language processing, 2010.

LOPES, G. Bolsonaro decretou estado de emergência antes do carnaval, mas os governadores ignoraram. 2020. Acesso em: 01 de Outubro de 2020. Disponível em: <https://www.e-farsas.com/bolsonaro-decretou-estado-de-emergencia-antes-do-carnaval-mas-os-governadores-ignoraram>.

MACHADO, A. P. et al. Mineração de texto em redes sociais virtuais aplicada à educação a distância. *Revista Digital da CVA - Ricesu*, v. 6, 2010.

MAIMON, O.; ROKACH, L. Data mining and knowledge discovery handbook. *Springer*, v. 2, 2005.

MARSLAND, S. *Machine learning: an algorithmic perspective*. [S.l.]: Chapman and Hall/CRC, 2011.

MARTINS, R. T.; HASEGAWA, R.; NUNES, M. G. V. Curupira: um parser funcional para a língua portuguesa. 2012. Acesso em: 8 de Agosto de 2020. Disponível em: <http://www.nilc.icmc.usp.br/nilc/download/nilc-tr-02-26.zip>.

MARUMO, S. F. Deep learning para classificação de fake news por sumarização de texto. *Universidade Estadual de Londrina*, 2018.

MATSUKI, E. Água tônica cura o coronavírus porque tem quinino da cloroquina boato. 2020. Acesso em: 01 de Outubro de 2020. Disponível em: <https://www.boatos.org/saude/agua-tonica-cura-o-coronavirus-porque-tem-quinino-da-cloroquina-boato.html>.

MAZIEIRO, E. G.; PARDO, T. A.; ALUISIO, S. Ferramenta de análise automática de inteligibilidade córpus (aic). *Universidade de São Paulo*, 2008. Acessado em 25 de novembro de 2020. Disponível em: <http://www.nilc.icmc.usp.br/nilc/download/NILCTR0808-MazieroPardo.pdf>.

MCNAMARA, D. S. et al. Automated evaluation of text and dis-course with coh-metrix. *Cambridge University Press*, 2014.

MIHALCEA, S.; STRAPPARAVA, C. The lie detector: Explorations in the automatic recognition of deceptive language. ACL-IJCNLP, 2009.

MITCHELL, T. *Machine Learning*. [S.l.]: McGraw-Hill, 1997.

MITKOV, R. et al. The handbook of computational linguistics and natural language processing. Wiley-Blackwell, 2010.

MOHAGHEGH, M.; ABDURAKHMANOV, A. Anomaly detection in text data sets using character-level representation. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2021. v. 1880, p. 012028.

MONTEIRO, R. et al. Contributions to the study of fake news in portuguese: New corpus and automatic detection results. *Computational Processing of the Portuguese Language*, Springer International, p. 324–334, 2018.

MOURA, R. S. Sos: Um sistema orto-sintático para a língua portuguesa. *Master's thesis, Universidade Federal de Pernambuco*, 1996.

MUIGAI, J. W. W. Understanding fake news. *International Journal of Scientific and Research Publications*, v. 9, p. 29–35, 2019.

NIRENBURG, S.; RASKIN, V. Ontological semantics. *Cambridge: MIT Press*, 2004.

OKANO, E. Y. Análise e caracterização de textos intencionalmente enganosos escritos em português usando métodos de processamento de textos. *Universidade de São Paulo*, 2020. Acessado em 24 de novembro de 2020. Disponível em: <https://www.teses.usp.br/teses/disponiveis/59/59143/tde-29062020-171001/publico/DissertacaoEmersonCorrigida.pdf>.

OKANO, E. Y. et al. Fake news detection on fake.br using hierarchical attention networks. *Computational Processing of the Portuguese Language. Lecture Notes in Computer Science*, 2020. DOI: 10.1007/978-3-030-41505-1.

PALMER, D. D. Text preprocessing. In: INDURKHYA, N.; DAMERAU, F. J. (Ed.). *Handbook of Natural Language Processing, Second Edition*. [S.l.]: Chapman and Hall/CRC, 2010. p. 9–30.

PANG, G. et al. Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 54, n. 2, p. 1–38, 2021.

PAPA, J. P. *Classificação supervisionada de padrões utilizando floresta de caminhos ótimos*. Tese (Doutorado) — UNICAMP, 2008.

PARDO, T. A. S.; NUNES, M. das G. V. Aprendizado bayesiano aplicado ao processamento de línguas naturais. 2002. Acesso em: 30 de Outubro de 2020. Disponível em: <https://sites.icmc.usp.br/taspardo/NILCTR0225-PardoNunes.pdf>.

PASSOS, L. A. et al. Unsupervised non-technical losses identification through optimum-path forest. *Electric Power Systems Research*, v. 140, p. 413–423, 2016.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

PINTO, S. C. S. Processamento de linguagem natural e extração de conhecimento. *Departamento de Engenharia Informática Faculdade de Ciências e Tecnologias*, 2015. Acesso em: 2 de Agosto de 2020. Disponível em: <http://hdl.handle.net/10316/35676>.

PIVETTA, S. P. Classificação de documentos do exército brasileiro utilizando o classificador naive bayes e técnicas de seleção de sentenças. 2013. Acesso em: 30 de Outubro de 2020. Disponível em: <http://dspace.unipampa.edu.br:8080/jspui/handle/riu/1569>.

RAHMATI, O. et al. Assessment of gini-, entropy-and ratio-based classification trees for groundwater potential modelling and prediction. *Geocarto International*, Taylor & Francis, v. 37, n. 12, p. 3397–3415, 2022.

REDDY, S. Glove and fasttext — two popular word vector models in nlp. 2019.

REDDY, Y.; VISWANATH, P.; REDDY, B. E. Semi-supervised learning: A brief review. *Int. J. Eng. Technol*, v. 7, n. 1.8, p. 81, 2018.

RIBEIRO, B. *Dona de casa foi linchada no Guarujá após oferecer fruta a criança*. 2014. Acesso em: 20 de Julho de 2020. Disponível em: <https://sao-paulo.estadao.com.br/noticias/geral,dona-de-casa-foi-linchada-no-guaruja-apos-oferecer-fruta-a-crianca,1163438>.

RIBEIRO, R.; OLIVEIRA, L.; TRANCOSO, I. Using morphossyntactic information in tts systems comparing strategies for european portuguese. In: *Proceedings of the 6th international workshop on computational processing of the portuguese language*. [S.l.: s.n.], 2003. p. 143–150.

RUFF, L. et al. Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019. p. 4061–4071.

RUSSEL, S.; NORVIG., P. Inteligência artificial. Elsevier Editora Ltda, p. 648–652, 2013.

SALTON, G.; BUCKLEY, C. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, v. 24, p. 513–523, 1988.

SALTON, G.; MCGILL, M. J. *Introduction to Modern Information Retrieval*. [S.l.]: McGrawHill, 1986.

SANTOS, W. P. L. S. et al. Measuring the impact of readability features in fake news detection. *Conference on Language Resources and Evaluation, LREC*, p. 1404–1413, 2020. Acessado em 25 de novembro de 2020. Disponível em: <https://www.aclweb.org/anthology/2020.lrec-1.176.pdf#cite.gomes3A19>.

SHAO, C. et al. The spread of fake news by social bots. *CoRR*, abs/1707.07592, 2017. Acesso em: 22 de Maio de 2020. Disponível em: <http://arxiv.org/abs/1707.07592>.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Bertimbau: Pretrained bert models for brazilian portuguese. In: CERRI, R.; PRATI, R. C. (Ed.). *Intelligent Systems*. [S.l.]: Springer International Publishing, 2020. p. 403–417.

SOUZA, F. A. A. d. *Análise de desempenho da rede neural artificial do tipo multilayer perceptron na era multicore*. Universidade Federal do Rio Grande do Norte, 2012. Disponível em: <https://repositorio.ufrn.br/handle/123456789/15447>.

SUN, C. et al. How to fine-tune bert for text classification? *ArXiv*, abs/1905.05583, 2019.

TANDOC, J. E.; LIM, Z. W.; LING, R. Defining "fake news": A typology of scholarly definitions. *Digital Journalism*, p. 1–17, 2017.

THU, P. P.; NEW, N. Implementation of emotional features on satire detection. *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, IEEE/ACIS, p. 149–154, 2017.

TRAYLOR, T. et al. Classifying fake news articles using natural language processing to identify in-article attribution as a supervised learning estimator. *13th International Conference on Semantic Computing, ICSC*, 2019. DOI = 10.1109/ICOSC.2019.8665593.

UITERKAMP, L. S. *Improving text representations for NLP from bags to strings of words*. Dissertação (Mestrado) — University of Twente, 2019.

VASWANI, A. et al. Attention is all you need. *Conference on Neural Information Processing Systems, NIPS*, 2017. Acessado em 5 de maio de 2020. Disponível em: <https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>.

WANG, H.; BAH, M. J.; HAMMAD, M. Progress in outlier detection techniques: A survey. *Ieee Access*, IEEE, v. 7, p. 107964–108000, 2019.

WANG, W. Y. *"Liar, Liar Pants on Fire":A New Benchmark Dataset for Fake News Detection*. Conference on Neural Information Processing Systems, NIPS, 2017. Acesso em: 18 de Maio de 2020. Disponível em: <https://arxiv.org/pdf/1705.00648.pdf>.

WARDLE, C. *Fake news. It's complicated*. 2017. Acesso em: 10 de Junho de 2020. Disponível em: <https://firstdraftnews.org/latest/fake-news-complicated/>.

WINTNER, S. *Syntactic parsing*. 2. ed. [S.l.]: Handbook of natural language processing, 2010.

YAN, D. et al. Network-based bag-of-words model for text classification. *IEEE Access*, IEEE, v. 8, p. 82641–82652, 2020.

YANG, Y. et al. *Ti-cnn: Convolutional neural networks for fake news detection.* 2018. Acesso em: 20 de Maio de 2020. Disponível em: <https://arxiv.org/pdf/1806.00749.pdf>.

ZHANG, H. The optimality of naive bayes. 2004. Acesso em: 10 de Agosto de 2020. Disponível em: <http://www.cs.unb.ca/~hzhang/publications/FLAIRS04ZhangH.pdf>.

ZHAO, Y.; NASRULLAH, Z.; LI, Z. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, v. 20, n. 96, p. 1–7, 2019. Disponível em: <http://jmlr.org/papers/v20/19-011.html>.

Apêndices

APÊNDICE A – Trabalhos publicados

Este apêndice lista outros trabalhos publicados ou em revisão durante o programa de mestrado em ordem cronológica do mais recente para o mais antigo.

A.1 Artigos para conferência

- *FakeRecogna Anomaly: Fake News Detection in a New Brazilian Corpus*: Este trabalho foi aceito e apresentado em 2023 na *International Conference on Computer Vision Theory and Applications* - VISAPP (Qualis-CC A3).
- *FakeRecogna: a new Brazilian corpus for fake news detection:* Este trabalho foi aceito e apresentado em 2022 na *International Conference on Computational Processing of the Portuguese Language* - PROPOR (Qualis-CC B1).