Engenharia de Software Inteligente para a Predição de Notícias Falsas: Representações de Codificador Bidirecional de Transformadores

Lara S. Moreira, Matheus de O. Ribeiro, Vitor X. Siqueira, Magino A. Correa, Cassio C. Junior, Eduardo A. Oliveira, João P. Merlugo, Fábio P. Basso, Williamson Silva, Gabriel M. Lunardi

[1]Universidade Federal do Pampa (UNIPAMPA)
Av. Tiaraju, 810 – 97546-550 – Alegrete – RS – Brasil

{laramoreira.aluno,matheusdor.aluno}@unipampa.edu.br

{vitorsiqueira.aluno,maginoleguissamo.aluno}@unipampa.edu.br

{cassioceolin.aluno,eduardojesus.aluno,joaomerlugo.aluno}@unipampa.edu.br

{fabiobasso,williamsonsilva,gabriellunardi}@unipampa.edu.br

Abstract. *In the digital and increasingly data-driven world, fake news spreads quickly, causing harmful effects. In this context, Intelligent Software Engineering emerges as a means to construct data-oriented intelligent applications. However, there are few efforts in the Portuguese language, especially those related to the experimentation of recent strategies such as the Representation of a Bidirectional Transformer Encoder (BERT). Therefore, we evaluated BERT's ability to detect fake textual news compared to traditional algorithms in seeing fake news based on text classification. The results show BERT superiority over other algorithms, showing a statistically significant difference in all cases.*

Resumo. *No mundo digital e cada vez mais orientado a dados, notícias falsas espalham-se rapidamente causando efeitos danosos. Nesse contexto, a Engenharia de Software Inteligente surge como meio para a construção de aplicações inteligentes orientadas a dados. Todavia, são escassos os esforços no idioma Português, especialmente os relacionados à experimentação de estratégias recentes, como a Representação de Codificador Bidirecional de Transformadores (BERT). Portanto, este trabalho avalia a capacidade do BERT, quando comparado com algoritmos tradicionais na detecção de notícias falsas baseada em classificação de textos.*

1. Introdução

De acordo com a Gartner [2021], uma das principais empresas mundiais especializadas em pesquisas e consultorias na área de Tecnologia da Informação, a receita mundial envolvendo software inteligente totalizará US$62,5 bilhões até o fim de 2022, um aumento de 21,3% em relação a 2021. Corroborando a isso, a Escola de Dagstuhl, um dos maiores encontros científicos anuais no qual são discutidos vários aspectos da Computação, teve como foco em 2020 a importância da Engenharia de Sistemas de Software baseados em Inteligência Artificial / Aprendizado de Máquina, também conhecida como SE4ML (*Software Engineering for Machine Learning)* [Kersting et al., 2020].

A SE4ML ou, ainda, Engenharia de Software Inteligente, tem por objetivo entender e adaptar práticas da Engenharia de Software para construção de sistemas inteligentes [Xie, 2018]. Sistemas dessa natureza foram popularizados por empresas como: Google (quando autocompleta ou advinha um termo de busca, recomenda uma notícia ou um vídeo no YouTube); Netflix (quando recomenda um filme ou série de acordo com as preferências do usuário); Amazon (quando sugere itens para compra com base no comportamento de navegação do usuário ou quando a assistente pessoal Alexa interage com as pessoas dentro de casa) [Reis et al., 2018]. Soluções de software como as mencionadas são desenvolvidas utilizando um ferramental orientado a dados que é pouco frequente na Engenharia de Software tradicional [Serban et al., 2020]. Alguns exemplos incluem: limpeza e análise de dados, estatística, processamento de linguagem natural, seleção de características, aprendizado de máquina (supervisionado, não-supervisionado, semi-supervisionado), avaliação de modelos, ajuste de parâmetros, recuperação de informações, dentre vários outros [Faceli, 2018, Lunardi et al., 2018].

O Processamento de Linguagem Natural (PLN) é uma área que apoia a construção de aplicações baseadas em dados textuais, sejam eles escritos ou falados [Manning and Schutze, 1999]. Uma das tarefas dentro dessa área é a classificação / predição textual que consiste em alocar um trecho de texto a uma determinada categoria de forma automática, baseada em algoritmos de aprendizado de máquina [Faceli, 2018]. Uma das aplicações emergentes da tarefa de classificação de textos é a detecção de notícias falsas cuja ideia é classificar o texto de uma notícia como "falso" ou "verdadeiro".

A disseminação de notícias falsas atingiu níveis alarmantes nas eleições presidenciais americanas de 2016 [Moraes, 2017]. Em 2018, o mesmo efeito aconteceu no Brasil nas eleições para presidente e, mais recentemente, em escala global, envolvendo o vírus SARS-CoV-2, causador da COVID-19, e as vacinas contra ele [Falcão et al., 2021]. Cenários como esses motivaram intensamente a concepção de soluções de PLN, em diferentes idiomas, para minimizar a desinformação causada pelo compartilhamento de notícias falsas. No Português, entretanto, existem poucos esforços relacionados à detecção de notícias falsas de forma automática. Isso ocorre porque existem poucos *datasets* com notícias previamente rotuladas, essencial para a detecção de notícias falsas baseada em classificação de textos [Monteiro et al., 2018]. Um esforço considerado referência é o *dataset* Fake.Br, fruto de trabalhos desenvolvidos no Núcleo Interinstitucional de Linguística Computacional (NILC-USP) utilizando abordagens tradicionais de PLN para a detecção de notícias falsas.

Outra lacuna que norteia este trabalho é o BERT (*Bidirectional Transformers for Language Understanding*), um algoritmo de aprendizado profundo, lançado pelo Google, que tem revolucionado a área de PLN [Devlin et al., 2018]. Entretanto, pouco esforço tem sido dedicado na análise de sua performance no idioma Português, lançando mão da sua variante BERTimbau [Souza et al., 2020]. Portanto, este artigo visa avaliar a capacidade de detecção de notícias falsas do algoritmo BERTimbau frente a algoritmos considerados tradicionais na classificação de textos. Para tanto, tem-se a seguinte hipótese a ser testada: o BERTimbau apresenta maior acurácia em relação aos algoritmos tradicionais.

2. Trabalhos Relacionados

A seguir serão apresentados alguns dos principais trabalhos relacionados a esta pesquisa.

Barbosa et al. [2022] conduziram um estudo comparativo em que compararam algoritmos de *machine learning* para detecção de *Fake News* na internet. Para isso, utilizaram os *datasets* Fake.BR e Sirene News, que junto totalizam 11.942 notícias. Os *datasets* foram analisados por quatro algoritmos de aprendizado de máquina: *Logistic Regression* (LR), *Stochastic Gradient Descent* (SGD), *Support Vector Machine* (SVM) e *Multilayer Perceptron* (MLP). Os quatro algoritmos foram submetidos a técnica de validação cruzada com 10 iterações, ou seja, a base de dados foi dividida em 10 grupos para obter-se um panorama da acurácia de cada um dos algoritmos, dividindo o *dataset* em partes para treinamento e testes. Os resultados obtidos no estudo mostraram que os modelos gerados pelos quatro algoritmos obtiveram uma precisão superior a 90%. O algoritmo *SVM* obteve o melhor desempenho, seguido do MLP, LR e SGD, com 96,39%, 95,14%, 94,30% e 92,90% de acurácia, respectivamente.

Villela et al. [2022] apresentaram uma análise da acurácia de trinta e oito algoritmos que se apresentaram capazes de identificar notícias falsas. Além dos resultados apresentados, os pesquisadores identificaram dezesseis *datasets* utilizados para a aplicação das técnicas de identificação de *fake news*. Dos trinta e oito algoritmos analisados, os três que apresentaram a maior acurácia foram: *Stacking Method*, *Bidirectional Recurrent Neural Network* (BiRNN) e o *Convolutional Neural Network* (CNN), com 99.94%, 99.82% e 99.80% de acurácia, respectivamente. Quanto aos demais algoritmos, todos tiveram uma acurácia superior a 90%. Os *datasets* mais utilizados pelos autores foram: O Kaggle (uma plataforma da Google, utilizada por cientistas de dados que contempla diversos *datasets* em sua plataforma para estudos de IA) foi o maior fornecedor de *datasets*; o Weibo (um microblog chinês, similar ao Twitter) foi o segundo com 3 ocorrências; Fake News Challenges (FNC) apresentou 2 ocorrências; e os demais com apenas uma ocorrência.

Almeida et al. [2021] realizaram uma comparação entre os algoritmos SVM e *Naive Bayes*. Os autores definiram métricas para identificar qual dos dois possui os melhores resultados quanto a classificação de notícias de política, buscando identificar se uma determinada notícia é falsa (ou não). As técnicas utilizadas para o processamento da linguagem natural foram: BOW *(Bag of words)* e TF-IDF *(Term frequency–inverse document frequency)*. Os resultados demonstraram que o algoritmo SVM revelou-se mais preciso, visto que, o algoritmo *Naive Bayes* apresentou dificuldades na identificação das notícias. Desta forma, o SVM fazendo uso da técnica BOW de processamento obteve a maior precisão(80,4%). Almeida et al. [2021] atribuíram a limitação do *Naive Bayes* a um problema de *underfitting*, apresentando uma hipótese muito simples para a resolução do problema de classificação de notícia falsa, resultando em valores ruins nas métricas, tanto no treino quanto no teste.

Nota-se que na literatura foram realizados diversos experimentos comparando algoritmos de *machine learning* capazes de identificar noticias falsas. No entanto, até onde se investigou, notou-se que poucos trabalhos avaliam o eficácia do algoritmo BERTimbau no idioma Português.

3. Metodologia

Neste seção, será apresentada a metologia utilizada para execução deste trabalho. Foram utilizados algoritmos de Aprendizado de Máquina para as classificações de notícias. A Figura 1 ilustra as etapas definidas para a condução deste trabalho, sendo: escolha de uma

base de dados, transformação dos dados, treinamento e comparação dos modelos.

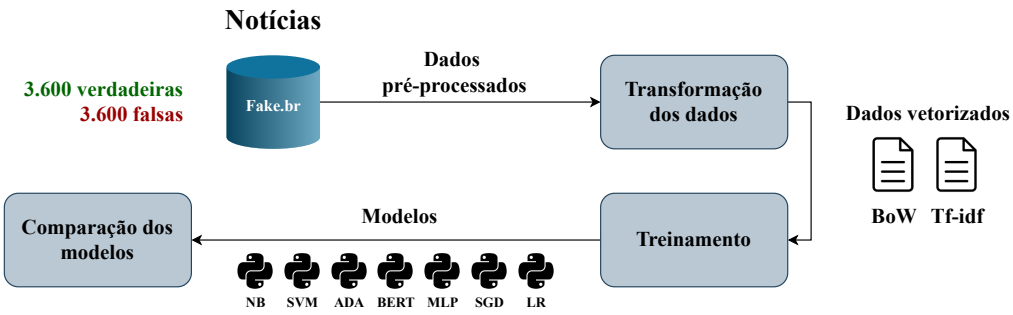

Figura 1. Metodologia adotada nesta pesquisa.

3.1. Conjunto de dados

Para que os testes e avaliações dos algoritmos pudessem ser realizados, foi-se necessário uma base de dados que serviria como entrada para tais algoritmos. Neste trabalho, a base de dados utilizada foi a Fake.br, uma das poucas em português e que possui anotação dos dados, no caso, notícias, em verdadeiras e falsas. Nela, existem 3.600 notícias verdadeiras e 3.600 notícias falsas, totalizando 7.200 notícias. As notícias estão divididas em seis categorias, sendo [Santos et al., 2018]: 4.180 sobre "política", 1.544 sobre "TV & celebridades", 1.276 sobre "sociedade & notícias diárias", 112 sobre "ciência & tecnologia", 44 sobre "economia" e 44 notícias sobre "religião". Todas elas datam entre janeiro de 2016 e janeiro de 2018, com texto livre de formatação.

O fato da base de dados Fake.br já possuir uma grande quantidade de notícias em português, e apresentar os dados já anotados, foi um dos fatores que influenciou a não criação de uma base de dados própria. Além disso, a produção de uma nova base de dados demandaria, além da coleta de um elevado número de notícias, a anotação manual de cada notícia como verdadeira ou falsa. Essa tarefa, portanto, necessitaria de uma avaliação humana para assegurar sua validação e utilização no treinamento dos algoritmos. Logo, demandaria muito tempo para o projeto, desenvolvimento e avaliação desta nova base.

A fim de sumarizar, de forma ilustrativa, o conteúdo da base de dados, utilizou-se a técnica de visualização de dados baseada em nuvem de palavras. A Figura 2b é uma nuvem das entidades nomeadas mais frequentes nas notícias verdadeiras, e na Figura 2a com as entidades nomeadas mais frequentes em notícias falsas. Entidades nomeadas são classificações de alguns elementos do texto, podendo ser pessoas, lugares, datas entre outras. Esse tipo de estratégia é útil para extrair informações relevantes de dados textuais [Sekine and Ranchhod, 2009].

3.2. Pré-processamento e Transformação dos Dados

A Fake.br fornece um arquivo CSV com o texto pré-processado, incluindo a remoção de *stopwords*, acentos e caracteres especiais que não agregam semântica aos modelos de aprendizado de máquina. Em seguida, as notícias foram submetidas à uma etapa de transformação dos dados, um processo no qual os textos são transformados em um vetor de valores numéricos que pode ser interpretado pelo classificador. A biblioteca *Scikit-learn* fornece um conjunto de ferramentas que permite realizar essa vetorização.

(a) ***Fake News*** **(b)** ***True News***

Figura 2. Nuvens de entidades nomeadas

Duas formas de vetorização foram utilizadas. A primeira, por meio da classe *CountVectorizer* que aplica o método *Bag of Words*, no qual é realizada a contagem de cada palavra. Outra forma de vetorização utilizada foi por meio da classe *TfidfVectorizer*, que por sua vez aplica a Frequência do Termo - Frequência Inversa do Documento (TF-IDF), na qual é calculado um valor para cada palavra que leva em consideração a aparição nos documentos, de forma que limite a relevância de termos que não possuem tanta importância para os algoritmos de aprendizado. Para a utilização no algoritmo BERT, os dados são tratados pela arquitetura *Transformers* que possui os métodos para instanciar o modelo e transformar os textos em *tokens*.

3.3. Treinamento e Avaliação dos Modelos

Existem inúmeros algoritmos de aprendizado de máquina para modelagem preditiva. Dentre eles, foram escolhidos seis algoritmos tradicionais e o BERT, que tem como principal inovação técnica a aplicação do treinamento bidirecional do *Transformer*, um modelo de atenção popular, à modelagem de linguagem.

O BERT consiste em um modelo que estabelece uma estrutura de representação de palavras que pode ser pré-treinado para utilização em uma ampla série de tarefas [Devlin et al., 2018]. Vários pré-modelos baseados em BERT foram treinados para um idioma em especifico, como o francês [Martin et al., 2020] e espanhol [Canete et al., 2020], para tentar superar os resultados do BERT, que é um modelo multilíngue com suporte a 104 idiomas. O BERTimbau consiste em um modelo pré-treinado para o idioma Português e que utiliza dados do brWaC [Souza et al., 2020]. Ainda, são disponibilizadas duas versões, sendo elas a *Base* e a *Large*.

Os algoritmos tradicionais utilizados foram: o *SGDClassifier* (SGD) que implementa o Gradiente Descendente Estocástico; o *LogisticRegression* (LR) que implementa a Regressão Logística; o MLPClassifier (MLP) que implementa Perceptron Multicamadas; o *Support Vector Machine* (SVM) que implementa o a Maquina de Vetores de Suporte; o *AdaBoostClassifier* (ADA) que implementa o AdaBoost-SAMME; e por fim, o *MultinomialNB* (NB) que implementa o algoritmo Naive-Bayes.

Para cada algoritmo de aprendizado de máquina, foram criados modelos usando várias combinações de hiper parâmetros ajustáveis que foram empregados para controlar o processo de aprendizado de algoritmos. Os hiper parâmetros foram ajustados de acordo com as configurações dos algoritmos nos trabalhos relacionados e também com base na documentação do *Scikit-learn*, descritos na Tabela 1. Depois de desenvolver esses modelos para cada combinação de hiper parâmetros, foram testados o desempenho de cada

Algoritmo	Configuração
SGD	loss='hinge',penalty='l2',alpha=0.0001,random_state=0, max_iter=5,tol=None
LR	random_state=0, solver='saga', multi_class= 'ovr'
MLP	solver='adam' ,alpha=1e-4,hidden_layer_size=(100),random_state=None
SVM	kernal='linear', probability=True,random_state=0
ADA	base_estimator=None, n_estimators=50, learning_rate=1,algorithm=SAMME.R, random_state=None
NB	alpha=1, fit_prior=True, class_prior=None
BERT	'bert', 'neuralmind/bert-base-portuguese-cased',args=model_args, use_cuda=device

Tabela 1. Configurações Algoritmos

uma dessas combinações usando uma técnica de validação cruzada com 10 dobras. A validação cruzada é um método amplamente empregado na predição de dados para avaliar a capacidade de generalização de modelos preditivos [Berrar, 2019].

As métricas utilizadas para comparação dos algoritmos foram acurácia, precisão, *recall* e medida-F. Foi gerado um relatório de classificação com as métricas usadas para medir a qualidade das previsões de todos os algoritmos de classificação. A acurácia representa a porcentagem de previsões corresponde às respostas reais esperadas; a precisão pode ser vista como a medida de exatidão do modelo; o *recall* é definido como a medida de sua completude; por fim, a medida-F é usada para combinar precisão e *recall* e pode fornecer um desempenho geral na previsão das notícias [Faceli, 2018].

4. Resultados e Discussão

Nesta seção serão apresentados os resultados da comparação de cada um dos algoritmos, considerando os dois modelos de representação vetorial.

Como mencionado anteriormente, o desempenho dos algoritmos foi testado utilizando a técnica de validação cruzada. Essa técnica consiste, no contexto deste trabalho, dividir o conjunto de dados em 10 partes iguais e alternar essas subdivisões entre os conjuntos de treino e teste, de modo com que todas passem por ambos conjuntos. Ao observar a Tabela 2, que mostra os resultados dos algoritmos por cada dobra, pode-se verificar que todos os algoritmos, exceto o algoritmo *Naive Bayes*, obtiveram uma acurácia média superior a 94%. Vale ressaltar que aqui, a vetorização utilizada foi o método *Bag of Words*. O algoritmo que obteve o melhor desempenho, com média de 96,62% de acurácia foi o LR, que calcula a soma dos recursos de entrada e calcula a logística do resultado. O algoritmos que obteve o pior desempenho foi *Naive Bayes*, tendo como média 83,21% de acurácia.

Já na Tabela 3, que mostra os resultados obtidos ao utilizar o método de vetorização Tf-idf, o algoritmo com melhor desempenho foi o SVM que classifica classes distintas. A partir desses dados, o algoritmo agrupa em classes que possuem mesma similaridade. Novamente observa-se que o algoritmo NB obteve o pior desempenho, sendo este com uma acurácia ainda inferior a apresentada anteriormente com o *Bag of Words*. Parte disso vem do fato que o NB não é um algoritmo que se comporta bem com dados contínuos. Analisando a Tabela 2 e a Tabela 3, também percebe-se que o BERT, mesmo estabelecendo uma estrutura de representação dos dados diferente, tem melhor desempenho que todos os algoritmos tradicionais, independente da forma de representação de textos escolhida.

	BoW						
Algoritmos	**SGD**	**LR**	**MLP**	**SVM**	**ADA**	**NB**	**BERTimbau**
Iteração 1	96,14%	96,67%	94,44%	96,39%	94,58%	82,08%	94,72%
Iteração 2	96,39%	96,25%	94,17%	96,39%	94,86%	81,94%	98,47%
Iteração 3	94,44%	95,83%	92,64%	95,14%	93,33%	82,22%	97,64%
Iteração 4	96,53%	97,64%	94,86%	96,25%	94,86%	85,56%	98,75%
Iteração 5	95,69%	96,53%	93,75%	95,83%	95,83%	85,69%	99,86%
Iteração 6	95,56%	97,08%	94,44%	96,81%	95,28%	84,31%	99,86%
Iteração 7	97,64%	96,25%	94,03%	97,64%	96,25%	82,08%	99,72%
Iteração 8	96,11%	96,39%	94,58%	96,67%	92,36%	78,61%	99,72%
Iteração 9	93,47%	96,94%	93,89%	96,11%	92,78%	84,58%	100,00%
Iteração 10	95,56%	96,67%	94,31%	96,53%	93,89%	85,00%	99,86%
Média	**95,65**%	**96,62**%	**94,11**%	**96,38**%	**94,40**%	**83,21**%	**98,86**%

Tabela 2. Resultados de acurácia da Validação Cruzada BoW

	Tf-idf						
Algoritmos	**SGD**	**LR**	**MLP**	**SVM**	**ADA**	**NB**	**BERTimbau**
Iteração 1	97,50%	95,28%	95,56%	97,36%	95,14%	58,61%	94,72%
Iteração 2	96,39%	96,11%	96,39%	96,94%	95,69%	58,75%	98,47%
Iteração 3	95,00%	93,33%	95,83%	95,28%	92,64%	61,11%	97,64%
Iteração 4	97,50%	95,97%	97,36%	97,64%	94,44%	64,72%	98,75%
Iteração 5	96,67%	95,42%	96,53%	96,81%	96,25%	61,11%	99,86%
Iteração 6	96,25%	95,42%	96,81%	96,39%	95,28%	63,06%	99,86%
Iteração 7	96,39%	95,56%	96,25%	96,53%	95,69%	60,14%	99,72%
Iteração 8	95,28%	94,17%	93,47%	95,56%	94,72%	57,92%	99,72%
Iteração 9	95,42%	93,75%	95,83%	95,69%	92,92%	61,67%	100,00%
Iteração 10	96,53%	96,39%	95,83%	96,81%	93,47%	60,42%	99,86%
Média	**96,29**%	**95,14**%	**95,99**%	**96,50**%	**94,63**%	**60,75**%	**98,86**%

Tabela 3. Resultados de acurácia da Validação Cruzada Tf-idf

Analisando o algoritmo SVM juntamente com o BOW, pode-se perceber que ele obteve uma média de 96% de acurácia, ficando empatado com ADA e atrás do BERTimbau e LR em acurácia. Por outro lado, na sua implementação utilizando TF-IDF acabou atingindo uma média de acurácia de mesma similaridade dos algoritmos ADA, SGD e MLP, estando somente atrás do BERTimbau. O fato dele obter este resultado se deve por seu funcionamento no qual é traçado um hiperplano no conjunto de dados para classificar suas classes distintas. A partir desses dados, ele agrupa em classes que possuem mesma similaridade.

Após a execução da validação cruzada, foi calculada a média dos resultados das dobras para criação de um relatório de classificação com todas as métricas para comparação dos algoritmos, como mostra a Tabela 4 para os algoritmos com a vetorização *Bag of Words* e a Tabela 5 para a vetorização Tf-idf.

Com base nos resultados, o BERT apresentou média de acurácia de **98,86%** e precisão, recall e medida-F de **99%** para notícias verdadeiras e falsas, o que demonstra superioridade em todos as métricas avaliadas em comparação com os algoritmos de aprendizado de máquina tradicionais.

		Precisão		*Recall*		Medida-f	
Algoritmos	**Acurácia média**	**True**	**Fake**	**True**	**Fake**	**True**	**Fake**
SGD	95%	96%	94%	94%	96%	95%	95%
LR	97%	97%	96%	96%	97%	97%	97%
MLP	94%	87%	91%	91%	97%	93%	94%
SVM	96%	97%	95%	94%	97%	96%	96%
ADA	94%	95%	94%	94%	95%	94%	94%
NB	83%	77%	94%	96%	71%	85%	81%

Tabela 4. Tabela de resultados BoW

		Precisão		*Recall*		Medida-f	
Algoritmos	**Acurácia média**	**True**	**Fake**	**True**	**Fake**	**True**	**Fake**
SGD	96%	96%	95%	95%	96%	96%	96%
LR	95%	97%	94%	94%	97%	94%	95%
MLP	96%	95%	96%	97%	94%	96%	95%
SVM	96%	97%	95%	95%	97%	96%	96%
ADA	95%	95%	94%	94%	95%	95%	95%
NB	61%	56%	99%	100%	22%	72%	35%

Tabela 5. Tabela de resultados Tf-idf

Após a comparação dos algoritmos em termo de acurácia, *recall* e medida-F, foram realizados testes estatísticos comparando a acurácia de todos algoritmos tradicionais com o BERT. Este tipo de teste foi realizado visando avaliar a hipótese apresentada na introdução deste artigo: o BERTimbau apresenta maior acurácia em relação aos algoritmos tradicionais. Para a avaliação do fator acurácia de cada algoritmo tradicional comparado com o BERT, foi utilizado o teste não-paramétrico de *Mann-Whitney U*, considerando que os dados de validação cruzada da acurácia do BERT não apresentaram distribuição normal [Wohlin et al., 2012].

A Tabela 6 apresenta o resultado do teste de *Mann-Whitney U*, visando avaliar estatisticamente a superioridade do BERT com relação a algoritmos de classificação tradicionais. De acordo com o resultado da Tabela 6, percebe-se que as diferenças entre as medianas do BERT e dos demais algoritmos possuem diferenças estatisticamente significativas (valor-p $<$ 0,05). Com isso, pode-se afirmar que a hipótese foi aceita, porém esse resultado deve ser considerado com acuidade, pois é sensível ao conjunto de dados utilizado para treinar e testar os algoritmos.

		BoW	Tf-idf
Algoritmo		**Valor p**	
BERT	SGD	0.0019	0.0028
BERT	LR	0.0032	0.0013
BERT	MLP	0.0002	0.0022
BERT	SVM	0.0032	0.0032
BERT	ADA	0.0008	0.0009
BERT	NB	0.0002	0.0002

Tabela 6. Teste de Mann-Whitney U Comparação Acurácia

5. Conclusão

Neste estudo, foi realizada a avaliação da capacidade de vários algoritmos de aprendizado de máquina. Esta avaliação foi conduzida com o objetivo de prever a classificação de notícias, como verdadeiras e falsas, utilizando dados disponíveis no *dataset* Fake.br.

Com base em nossa análise de sete algoritmos, conseguimos aceitar estatisticamente a hipótese levantada e descobrir que o algoritmo BERTimbau têm acurácia maior do que outros algoritmos de aprendizado de máquina testados.

6. Agradecimentos

Os autores agradecem a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código Financeiro 001 e a Universidade Federal do Pampa (UNIPAMPA - Alegrete) pelo apoio. Williamson Silva agradece pelo apoio financeiro da FAPERGS (Projeto ARD/ARC – processo 22/2551-0000606-0).

Referências

Laura D. De Almeida, Victor Fuzaro, Falmer V. Nieto, and André Luiz Maciel Santana. Identificação de "fake news" no contexto político brasileiro: uma abordagem computacional. *Anais do II Workshop sobre as Implicações da Computação na Sociedade (WICS 2021)*, 2021.

Vinícius Nunes Barbosa, Francisco Mendes Mendes Neto, Sebastiao Alves Filho, and Lenardo Silva. A comparative study of machine learning algorithms for the detection of fake news on the internet. In *XVIII Brazilian Symposium on Information Systems*, SBSI, page 8, 2022. ISBN 9781450396981.

Daniel Berrar. Cross-validation. In *Encyclopedia of Bioinformatics and Computational Biology - Volume 1*, pages 542–545. 2019. doi: 10.1016/b978-0-12-809633-8.20349-x.

José Canete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. Spanish pre-trained bert model and evaluation data. *Pml4dc at iclr*, 2020:1–10, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL `https://arxiv.org/abs/1810.04805`.

K. Faceli. *Inteligência artificial: uma abordagem de aprendizado de máquina*. Grupo Gen - LTC, 2018. ISBN 9788521618805.

Paula Falcão, Aline Batista de Souza, et al. Pandemia de desinformação: as fake news no contexto da covid-19 no brasil. 2021.

Gartner. Gartner forecasts worldwide artificial intelligence software market to reach $62 billion in 2022. https://www.gartner.com/en/newsroom/press-releases/2021-11-22-gartner-forecasts-worldwide-artificial-intelligence-software-market-to-reach-62-billion-in-2022, 2021. (Accessed on 08/19/2022).

Kristian Kersting, Miryung Kim, Guy Van den Broeck, and Thomas Zimmermann. Se4ml-software engineering for ai-ml-based systems (dagstuhl seminar 20091). In *Dagstuhl Reports*, volume 10. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

Gabriel Machado Lunardi, Guilherme Medeiros Machado, Fadi Al Machot, Vinícius Maran, Alencar Machado, Heinrich C Mayr, Vladimir A Shekhovtsov, and José Palazzo M de Oliveira. Probabilistic ontology reasoning in ambient assistance: predicting human

actions. In *32nd International Conference on Advanced Information Networking and Applications (AINA)*, pages 593–600. IEEE, 2018.

C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999. ISBN 9780262303798. URL https://books.google.com.br/books?id=3qnuDwAAQBAJ.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. CamemBERT: a tasty French language model. In *58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, 2020.

Rafael A Monteiro, Roney LS Santos, Thiago AS Pardo, Tiago A de Almeida, Evandro ES Ruiz, and Oto A Vale. Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In *International Conference on Computational Processing of the Portuguese Language*, pages 324–334. Springer, 2018.

Isabela Moraes. Notícias falsas e pós-verdade: o mundo das fake news e da (des)informação — politize! https://www.politize.com.br/noticias-falsas-pos-verdade/, 2017. (Accessed on 08/28/2022).

Arsénio Reis, Dennis Paulino, Hugo Paredes, Isabel Barroso, Maria João Monteiro, Vitor Rodrigues, and João Barroso. Using intelligent personal assistants to assist the elderlies an evaluation of amazon alexa, google assistant, microsoft cortana, and apple siri. In *2018 2nd International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW)*, pages 1–5. IEEE, 2018.

Roney L. S. Santos, Rafael A. Monteiro, and Thiago Alexandre Salgueiro Pardo. The fake . br corpus-a corpus of fake news for brazilian portuguese. 2018.

Satoshi Sekine and Elisabete Ranchhod. *Named entities: recognition, classification and use*, volume 19. John Benjamins Publishing, 2009.

Alex Serban, Koen van der Blom, Holger Hoos, and Joost Visser. Adoption and effects of software engineering best practices in machine learning. In *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, ESEM '20, 2020. ISBN 9781450375801.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. Bertimbau: Pretrained bert models for brazilian portuguese. In Ricardo Cerri and Ronaldo C. Prati, editors, *Intelligent Systems*, pages 403–417, Cham, 2020. Springer International Publishing. ISBN 978-3-030-61377-8.

Humberto Fernandes Villela, Fábio Corrêa, Jurema Suely de Araújo Nery Ribeiro, Air Rabelo, and Emerson Eustáquio Costa. Uma analise da acuracia obtida e datasets utilizados em algoritmos de identificação de fake news. In *ISLA 2022 Proceedings*. Lacais, 2022.

Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

Tao Xie. Intelligent software engineering: Synergy between ai and software engineering. In *International symposium on dependable software engineering: Theories, tools, and applications*, pages 3–7. Springer, 2018.