Detecção de Fake News com Processamento de Linguagem Natural

João Pedro da Silva Zampoli
Ciência e Tecnologia
UNIFESP
São José dos Campos, Brasil
joao.zampoli@unifesp.br

Luiza de Souza Ferreira
Ciência e Tecnologia
UNIFESP
São José dos Campos, Brasil
souza.luiza@unifesp.br

***Abstract*—Atualmente, a divulgação de informações na internet é dificilmente diferenciada entre verdadeira e falsa, resultando nas chamadas *Fake News*, sendo assim, um sistema que faça a diferenciação dessas informações seria benéfico em detrimento ao combate à desinformação. Este artigo descreve o treinamento de modelos de Inteligência Artificial para detecção de notícias falsas usando técnicas de classificação. O objetivo é treinar um modelo de Processamento de Linguagem Natural capaz de distinguir notícias falsas de notícias reais.**

I. Introdução

Em um mundo cada vez mais influenciado pela tecnologia, a diferenciação entre o que é real e o que é falso ganha um novo nível de dificuldade [1]. A rápida comunicação entre pessoas, que têm sido facilitada por meio das redes sociais provocou um novo fenômeno de impulsionamento e propagação de notícias falsas, também popularmente conhecidas como *Fake News*, podem prejudicar negativamente a estabilidade social, processos democráticos e credibilidade da imprensa.

Dessa forma, este trabalho busca diferenciar notícias falsas e verdadeiras por meio de técnicas de Inteligência Artificial, o que está diretamente relacionado aos ODS 16 (Paz, Justiça e Instituições eficazes) e ODS 4 (Educação de Qualidade) por propor uma abordagem de detecção que promova o pensamento crítico e o combate à desinformação.

A detecção de *Fake News* já foi explorada por autores anteriormente. Neste trabalho, 5 estudos encontrados foram fundamentais para entender os conceitos e análises necessárias para atingir o objetivo de detecção, sendo esses estudos de [2], [3], [4], [5] e [6].

No estudo deste artigo, após a seleção do *dataset*, os textos foram representados como vetores pelas métodos *Bag-of-words* e *TF-IDF*. Após essas vetorizações, 3 modelos distintos foram executados, sendo estes: *KNN*, *Naive Bayes* e *SVM*. Além disso, também foi executado um modelo adicional *BERT* para fins comparativos.

II. Trabalhos Relacionados

Em [2], os autores apresentaram técnicas de aprendizado de máquina para detectar notícias falsas por meio de regressão logística, árvore de decisão, floresta aleatória e algoritmos passivo-agressivos.

Em [3], o estudo foi focado na classificação de *Fake News* em redes sociais por meio de textos. Neste estudo, foram usados 4 métodos para extrair características dos textos: *TF-IDF*, *count vector*, *character level vector* e *N-gram*. Além destes diferentes métodos para extração dos dados, foram usados 10 classificadores diferentes de *Machine Learning* e *Deep Learning* para categorizar as *Fake News*.

Em [4], os autores apresentaram uma implementação para a detecção de *Fake News* usando 13 diferentes métodos de representação de textos, sendo categorizados como: métodos baseados em contagem, métodos independentes de contexto e métodos dependentes de contexto. Além de diferentes métodos, foram utilizados 17 diferentes modelos para a classificação, sendo categorizados como: modelos clássicos de *Machine Learning*, modelos de aprendizado por agrupamento e modelos *Deep Learning*.

Em [5], os autores propuseram um modelo de detecção de *Fake News* em redes sociais. Neste estudo, os textos no *dataset* foram representados como vetores por meio dos métodos *TF-IDF* e *Document-Term Matrix*. Após esta representação, 23 modelos supervisionados de inteligência artificial foram aplicados nos dados e comparados por meio de 4 métricas de avaliação.

Em [6], os autores apresentaram um novo *dataset* público com notícias reais e falsas em português, e promoveram uma análise de detecção de *fake news* por meio de métodos de *Machine Learning*. A análise foi feita com diferentes conjuntos de características e diferentes métodos de classificação.

III. Metodologia

Este trabalho propõe a utilização de técnicas de Processamento de Linguagem Natural com o objetivo de diferenciar as notícias verdadeiras das chamadas *Fake News*.

Os dados utilizados foram obtidos por meio do repositório do *Github* do estudo [6] com *stop-words* e acentos removidos. Esses dados, sendo 7200 notícias, foram separados em 70% para treino e 30% para teste, de forma aleatória, para serem usados no estudo.

Para a representação dos textos foram utilizadas duas formas de vetorização, sendo estas: *Bag-of-words* e *TF-IDF*.

Bag-of-words é uma técnica de vetorização, onde o texto é transformado em um conjunto não-ordenado de palavras, sendo representado como uma matriz. Esta técnica é simples, porém eficiente para modelos básicos.

TF-IDF é uma técnica de vetorização, onde cada palavra é transformada de acordo com sua frequência em um documento e sua frequência inversa em todos os documentos, conforme as equações 1.

$$\begin{aligned} \text{TF}(t,d) &= \frac{\text{número de vezes que } t \text{ aparece em } d}{\text{número total de termos em } d} \\ \text{IDF}(t) &= \log\left(\frac{N}{1+df}\right) \\ \text{TF-IDF}(t,d) &= \text{TF}(t,d) \times \text{IDF}(t) \end{aligned} \tag{1}$$

Para a classificação das notícias, foram treinados 4 modelos diferentes, sendo estes: *K-nearest neighbors* (KNN), *Naive Bayes*, *Support Vector Machine* (SVM) e *Bidirectional Encoder Representations from Transformers* (BERT).

KNN é uma técnica de classificação que consiste em três etapas. Dado um novo elemento, o algoritmo calcula a medida de similaridade com os demais elementos, ranqueia os k elementos mais similares e classifica o novo elemento de acordo com a classe majoritária dos k elementos mais próximos. Para este estudo, a medida de similaridade escolhida foi a Distância Euclidiana evidenciada na equação 2, e o valor de k escolhido foi 3.

$$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2} \tag{2}$$

Naive Bayes é uma técnica de classificação que, dado um novo elemento, calcula a probabilidade de cada classe para esse elemento com o Teorema de Bayes evidenciado na equação 3, resultando na classe com maior probabilidade sendo escolhida para esse novo elemento.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \tag{3}$$

SVM é uma técnica de classificação não paramétrica que utiliza as posições das classes no espaço para classificar os elementos, utilizando uma fronteira de decisão para separar as classes. Para esse estudo, a fronteira de decisão utilizada é linear.

BERT é um modelo de classificação de *Deep Learning* que utiliza *Encoder*, ou seja, cada palavra é transformada em um vetor, que é refinado por meio de várias camadas.

IV. Análise Experimental

A. Conjunto de dados

Na tabela I, encontra-se a disposição do *dataset* utilizado com 7200 notícias.

Foi analisado o número de elementos em cada classe para averiguar o balanceamento de dados, sendo 3600 notícias verdadeiras e 3600 notícias falsas, conforme mostrado na Figura 1.

TABLE I
Disposição da base de dados

index	label	preprocessed_news
0	fake	katia abreu diz vai colocar expulsao moldura n...
1	fake	ray peita bolsonaro conservador fake entrevist...
2	fake	reinaldo azevedo desmascarado policia federal ...
3	fake	relatorio assustador bndes mostra dinheiro pub...
4	fake	radialista americano fala sobre pt vendem ilus...
...	...	...
7195	true	jornal britanico acao contra lula lava jato se...
7196	true	temer diz acionou pf cade investigar aumentos ...
7197	true	obstaculos politicos temer especialistas ouvid...
7198	true	setembro boa noite aqui estao principais notic...
7199	true	envolvo politica diz brasileiro preso venezuel...

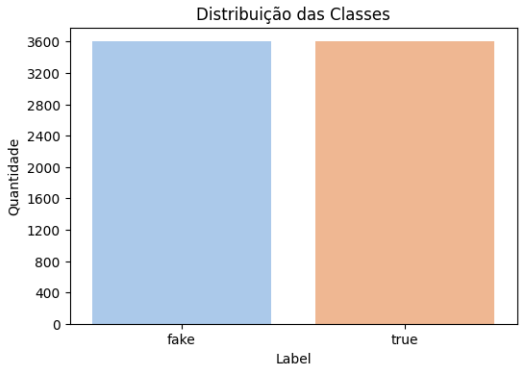

Fig. 1. Distribuição das classes no *dataset*

B. Configuração do algoritmo e do ambiente computacional

A linguagem de programação *Python* foi escolhida por possuir um grande número de bibliotecas com funções úteis para o estudo proposto.

As bibliotecas utilizadas foram:

- *Pandas*: Para manipulação dos dados.
- *Numpy*: Realização de operações matemáticas de forma otimizada.
- *Matplotlib*: Plotagem de gráficos e matrizes de correlação.
- *Seaborn*: De forma a auxiliar nas plotagens do *MatPlotlib*.
- *Tensorflow*: Para a utilização dos algoritmos de *Machine Learning*.
- *SciKit Learn*: Também para a utilização de algoritmos de Inteligência Artificial.
- *Transformers*: Para a utilização de modelos transformers (no caso deste relatório, o *BERT*).

A ferramenta utilizada para edição e execução do código foi o *Google Colab* por possuir uma interface de fácil organização e com a possibilidade para colaboração entre membros. Além disso, foi utilizada a aceleração de GPU baseada na placa *Nvidia Tesla T4*, de forma a otimizar e diminuir o tempo necessário para a geração do modelo *BERT*.

Por conveniência, foi adotada uma estrutura de *Jupyter Notebook*, de forma a facilitar a separação de códigos por

função, comentários e execuções de áreas separadas do programa.

C. Critérios de análises

Cada modelo de classificação foi avaliado por meio de sua acurácia e matriz de confusão, sendo os resultados comparados entre os modelos. Além disso, os tempos de execução dos conjuntos de treinamentos e testes dos modelos *KNN*, *Naive Bayes* e *SVM* foram comparados entre si juntamente com suas técnicas de vetorização, sendo *Bag-of-words* e *TF-IDF*.

D. Resultados e discussão

1) KNN: Com *Bag-of-words*, o método de classificação *KNN* obteve uma acurácia de 70% conforme mostrado na figura 2. Essa acurácia pode também ser observada por meio da matriz de confusão na figura 3.

```
Acurácia do KNN usando bag-of-words: 0.7046296296296296
```

Fig. 2. *Print* da acurácia do *KNN* usando *Bag-of-words*

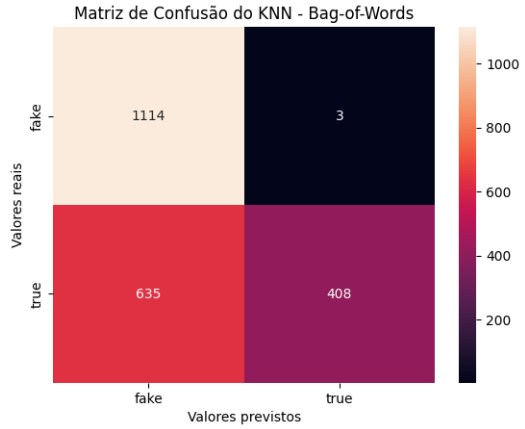

Fig. 3. Matriz de confusão do *KNN* usando *Bag-of-words*

Já com *TF-IDF*, o *KNN* obteve uma acurácia de 69% conforme mostrado na figura 4. Essa acurácia também pode ser observada por meio da matriz de confusão na figura 5.

```
Acurácia do KNN usando TF-IDF: 0.6893518518518519
```

Fig. 4. *Print* da acurácia do *KNN* usando *TF-IDF*

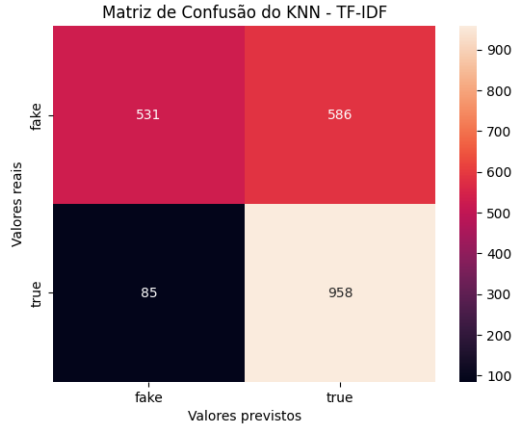

Fig. 5. Matriz de confusão do *KNN* usando *TF-IDF*

2) Naive Bayes: Com *Bag-of-words*, o método de classificação *Naive Bayes* atingiu uma acurácia de 82% conforme mostrado na figura 6. Essa acurácia também pode ser observada por meio da matriz de confusão na figura 7.

```
Acurácia do Naive Bayes usando bag-of-words: 0.8199074074074074
```

Fig. 6. *Print* da acurácia do *Naive Bayes* usando *Bag-of-words*

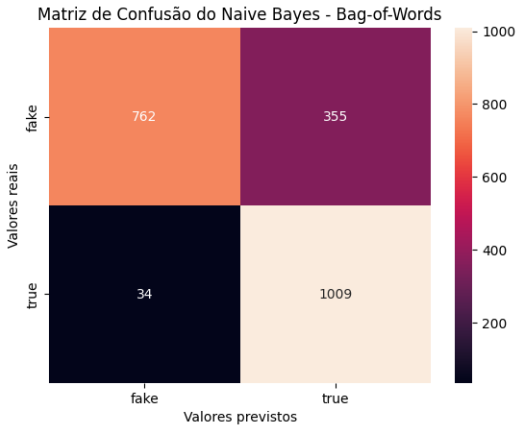

Fig. 7. Matriz de confusão do *Naive Bayes* usando *Bag-of-words*

Já com *TF-IDF*, o *Naive Bayes* obteve uma acurácia de 59% conforme mostrado na figura 8. Essa acurácia também pode ser observada por meio da matriz de confusão na figura 9.

```
Acurácia do Naive Bayes usando TF-IDF: 0.5925925925925926
```

Fig. 8. *Print* da acurácia do *Naive Bayes* usando *TF-IDF*

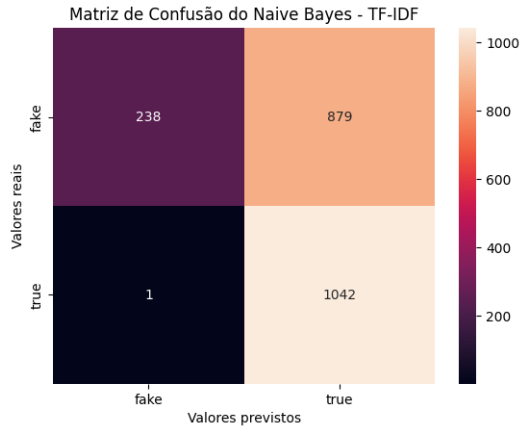

Fig. 9. Matriz de confusão do *Naive Bayes* usando *TF-IDF*

3) SVM: Com *Bag-of-words*, o método de classificação *SVM* obteve uma acurácia de 0.96% conforme mostrado na figura 10. Essa acurácia pode também ser observada pela matriz de confusão na figura 11.

```
Acurácia do SVM usando bag-of-words: 0.9574074074074074
```

Fig. 10. *Print* da acurácia do *SVM* usando *Bag-of-words*

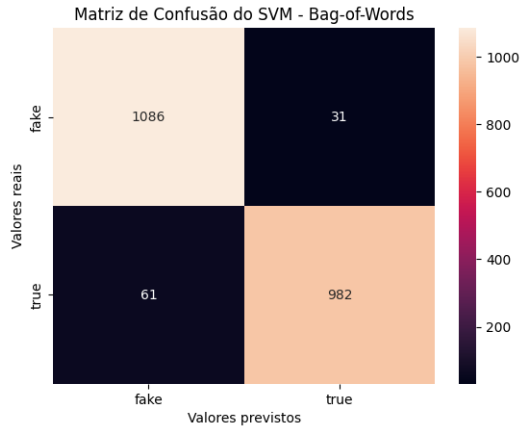

Fig. 11. Matriz de confusão do *SVM* usando *Bag-of-words*

Já com *TF-IDF*, o *SVM* atingiu uma acurácia de 96% conforme mostrado na figura 12. Essa acurácia pode também ser observada pela matriz de confusão na figura 13.

```
Acurácia do SVM usando TF-IDF: 0.9574074074074074
```

Fig. 12. *Print* da acurácia do *SVM* usando *TF-IDF*

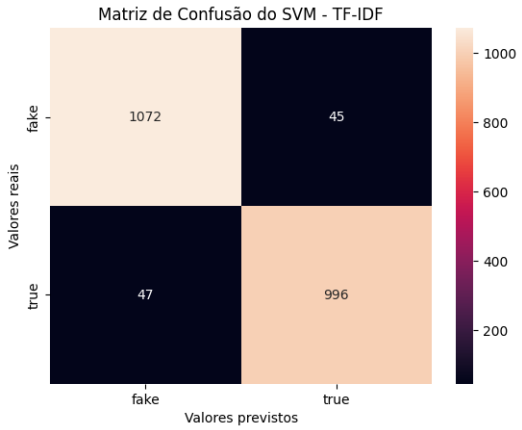

Fig. 13. Matriz de confusão do *SVM* usando *TF-IDF*

4) BERT: O método de classificação *BERT* obteve uma acurácia de 94% como mostrado na figura 14. Essa acurácia pode também ser observada pela sua matriz de confusão na figura 15.

```
Acurácia do BERT no conjunto de teste: 0.9407
```

Fig. 14. *Print* da acurácia do *BERT*

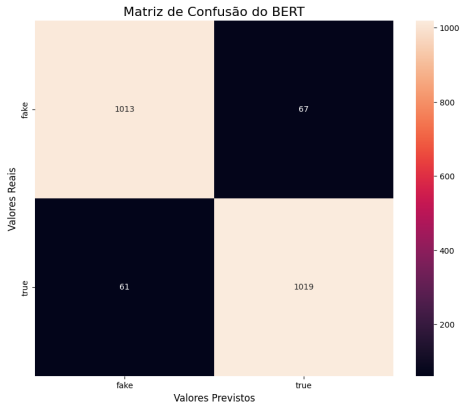

Fig. 15. Matriz de confusão do *BERT*

Para visualização das diferentes acurácias obtida entre os modelos de classificação, foi feito um gráfico de acordo com cada modelo e vetorização conforme mostrado na figura 16.

Além da diferença das acurácias, é importante observar o tempo de execução de cada algoritmo, um gráfico foi feito com esse objetivo, onde os tempos de treinamento e teste são observados separadamente, assim como a vetorização. Esse gráfico compara os tempos de execução apenas do *KNN*, *Naive Bayes* e *SVM*, e é mostrado na figura 17.

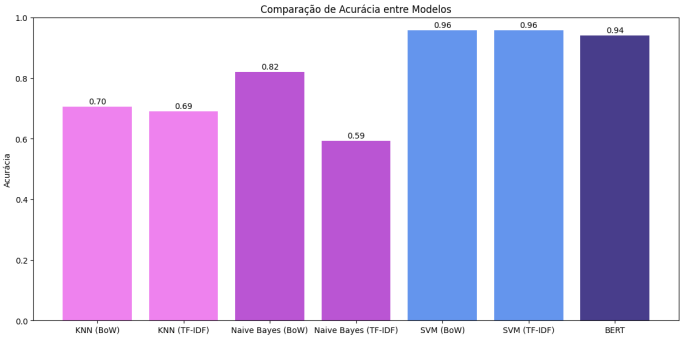

Fig. 16. Gráfico de comparação de acurácias entre modelos

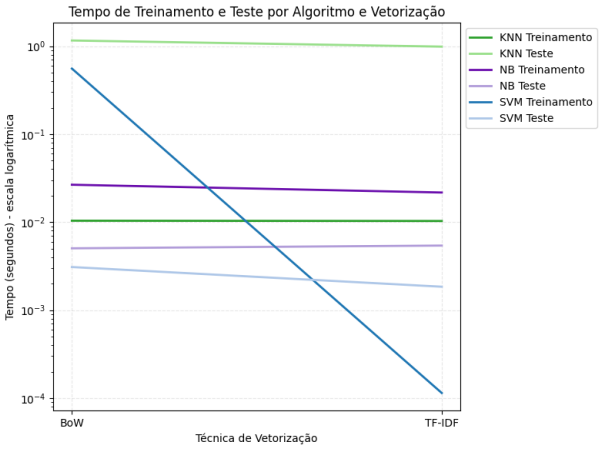

Fig. 17. Gráfico de tempos de treinamento e teste por modelo e vetorização

V. Conclusão

A partir dos resultados obtidos na figura 16, pode-se observar que os modelos *SVM* e *BERT* atingiram a melhor acurácia.

Além disso, o modelo *SVM* com a vetorização *TF-IDF* apresentou uma redução significativa no tempo de execução do treinamento em relação ao *SVM* com a vetorização *Bag-of-words*, conforme visto na figura 17.

Porém, observando a matriz de confusão do *SVM* usando *Bag-of-words* na figura 11, o número de notícias falsas que foram classificadas como verdadeiras é menor comparado com o *SVM* usando *TF-IDF* na figura 13 e o *BERT* na figura 15.

Além dessas observações, o modelo que teve o menor número de notícias falsas classificadas como verdadeiras foi o *KNN* com *Bag-of-words*, como mostrado na figura 3, sendo esse número igual à apenas 3 notícias.

Portanto, a escolha do modelo de classificação e sua vetorização depende do objetivo requerido, tendo que diferentes modelos foram melhores classificados em diferentes categorias.

References

[1] L. R. Serrano, "Imprensa e mídias sociais: o desafio de separar o joio do trigo," *Jornal da USP*, 2023.

[2] M. A. Shaik, M. Y. Sree, S. S. Vyshnavi, T. Ganesh, D. Sushmitha, and N. Shreya, "Fake news detection using nlp," in *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, pp. 399–405, 2023.

[3] A. Abdulrahman and M. Baykara, "Fake news detection using machine learning and deep learning algorithms," in *2020 International Conference on Advanced Science and Engineering (ICOASE)*, pp. 18–23, 2020.

[4] F. Farhangian, R. M. Cruz, and G. D. Cavalcanti, "Fake news detection: Taxonomy and comparative study," *Information Fusion*, vol. 103, p. 102140, 2024.

[5] F. A. Ozbay and B. Alatas, "Fake news detection within online social media using supervised artificial intelligence algorithms," *Physica A: Statistical Mechanics and its Applications*, vol. 540, p. 123174, 2020.

[6] R. M. Silva, R. L. Santos, T. A. Almeida, and T. A. Pardo, "Towards automatically filtering fake news in portuguese," *Expert Systems with Applications*, vol. 146, p. 113199, 2020.