

Identificação de notícias falsas em aplicativos de mensagens

Marcos Medeiros da Silva Filho

Orientador: Dr. Luciano A. Digiampietri

Relatório Final - Iniciação Científica Voluntária de um semestre

Período: 23/09/24 a 31/03/25

Resumo: Atualmente, a velocidade e intensidade da propagação de notícias falsas (fake news) atingiu um patamar inédito. Essa propagação pode causar diversos prejuízos para a sociedade como a deslegitimação de pesquisas realizadas em universidades públicas. Os principais mecanismos utilizados para a divulgação destas notícias são os aplicativos de mensagens e as redes sociais. O objetivo do presente projeto é desenvolver e testar uma ferramenta capaz de acompanhar um ou mais grupos de um aplicativo de trocas de mensagens e a cada mensagem publicada avalie a chance de se tratar de uma notícia falsa. Assim, foram desenvolvidos dois modelos baseados no BERTimbau, os quais foram testados no Bot construído para a plataforma Telegram.

Palavras-chave: Fake News; Notícias Falsas; BERTimbau; Aplicativos de Trocas de Mensagem

Abstract: Currently, the speed and intensity of the spread of fake news has reached an unprecedented level. This spread can cause several losses to society, such as the delegitimization of research carried out in public universities. The main mechanisms used to disseminate this news are messaging applications and social networks. The objective of this project is to develop and test a tool capable of monitoring one or more groups on a messaging application and assessing the chance of each message being fake news for each message published. Thus, two models based on BERTimbau were developed, which were tested on the Bot built for the Telegram platform.

Keywords: Fake News; BERTimbau; Messaging Applications.

1. Introdução e Justificativa

No contexto moderno, a propagação de notícias falsas (fake news) tem se apresentado como um entrave para correta transmissão e compartilhamento de informação entre os indivíduos. Esse fenômeno pode acarretar diversos efeitos nocivos à sociedade (SHARMA et al., 2019), como a divulgação de tratamentos de saúde sem nenhuma comprovação científica e eventualmente com efeitos colaterais; o incentivo à não vacinação, entre outros. Ademais, é notório o uso recorrente de fake news no campo político: ou supervalorizando de forma mentirosa as ações de um político ou partido político ou de maneira oposta, desvalorizando ou difamando um partido ou um político.

Sob essa análise, já é possível observar as consequências negativas do uso de notícias falsas em acontecimentos recentes. Por exemplo, durante a pandemia do coronavírus, onde a divulgação de notícias falsas dificultou o enfrentamento da doença e levou parte da população a não se vacinar. No Brasil, informações enganosas relacionadas à pandemia circularam principalmente nas plataformas WhatsApp e Facebook, sendo 20,1% sobre política, 19,5% sobre epidemiologia e estatística e 16,1% sobre prevenção. Um exemplo de notícia sobre epidemiologia e estatística ocorrido na pandemia é "Hospital das Clínicas de São Paulo vazio em meio a pandemia". Vale ressaltar que tais notícias utilizaram palavras que maximizam a propagação nas redes sociais, as quais aumentaram 34,3% as buscas que utilizam os termos presentes nas notícias falsas. Assim, é inegável que as notícias falsas são a principal problemática do contexto atual (BARCELOS et al., 2021).

O termo "fake news" está relacionado a criação e distribuição intencional de boatos ou conteúdo sabidamente falso, eventualmente combinado com algum fato, via televisão, rádio, aplicativos de troca de mensagens ou redes sociais tipicamente com o intuito de maximizar a reverberação da informação para algum tipo de ganho (financeiro ou político, por exemplo) (HUNT, 2016; SCHLESINGER, 2017).

Diferentes serviços foram desenvolvidos nos últimos anos para a identificação de notícias falsas e a divulgação de quais, entre as notícias mais compartilhadas em redes sociais ou aplicativos, são verdadeiras ou falsas. Apenas para ilustrar, podemos citar o projeto Eleições Sem Fake¹, projeto desenvolvido por pesquisadores da UFMG; o site e-farsas² e a seção fato-ou-fake do portal G1³.

De um modo geral, os serviços de identificação de notícias falsas são baseados na ação de pessoas (por exemplo, jornalistas) que verificam “manualmente” a veracidade das principais notícias que estão circulando nas mídias sociais e então rotulam essas notícias como falsas ou não (RESENDE et al, 2018). A identificação automática de notícias falsas é uma atividade interdisciplinar (ZHOU e ZAFARANI, 2019), que envolve o processamento de língua natural (para a identificação do assunto tratado, polarização, etc.) (VICARIO et al., 2019), ciência da informação combinada com análise de redes sociais (para a análise da propagação da informação) e mineração de dados (para a predição de comportamento ou classificação das notícias) (ZHOU et al., 2019).

Entretanto, detectar notícias falsas manualmente é uma tarefa árdua, visto que a taxa de crescimento de notícias falsas é atípica e o processo de verificação manual é trabalhoso. Nesse sentido, é necessário o desenvolvimento de mecanismos automáticos de detecção de fake news para amenizar o problema em pauta, tendo em vista a limitação da detecção manual e as consequências negativas as quais as notícias falsas propagam na sociedade (FAGUNDES, 2024).

¹ <https://dcc.ufmg.br/category/eleicoes-sem-fake/> (acessado em 04/02/2025)

² <https://www.e-farsas.com/> (acessado em 04/02/2025)

³ <https://g1.globo.com/fato-ou-fake/> (acessado em 04/02/2025)

2. Objetivo

O objetivo geral deste projeto é desenvolver, testar e disponibilizar uma ferramenta capaz de acompanhar um ou mais grupos em um aplicativo de mensagens e avaliar as postagens como provável notícia falsa ou não.

Para atingir o objetivo geral, define-se os seguintes objetivos específicos:

- Construir modelos preditivos baseado no BERTimbau;
- Disponibilizar os modelos através de uma biblioteca em Python;
- Desenvolver um Bot em Python que utilize os modelos;
- Aplicar o Bot desenvolvido para a plataforma Telegram.

3. Conceitos fundamentais

Esta seção resume os principais conceitos e tecnologias utilizados ao longo do presente trabalho, incluindo a descrição de ferramentas, algoritmos ou bibliotecas de funções.

3.1 *Bidirectional Encoder Representations from Transformers (BERT)*

O BERT é um modelo de aprendizado de máquina baseado na arquitetura *Transformers*. A principal ideia é capturar o significado de cada token com base no contexto direito e esquerdo conjuntamente (AKABANE e BARBON, 2022). Assim, o BERT resolve o problema da captura do contexto, o qual a maioria dos modelos anteriores não conseguia.

Ademais, o BERT é pré-treinado em um grande conjunto de dados não rotulados, com a finalidade do modelo aprender sobre os aspectos da linguagem. Durante o pré-treinamento, são usadas duas tarefas não supervisionadas: *Masked LM* e Previsão da próxima sentença. Em *Masked LM*, são mascarados tokens aleatórios na entrada e o modelo deverá prever quais tokens estão por trás das máscaras. Enquanto na previsão de próxima sentença, o objetivo é verificar, dadas as entradas A e B, se B é uma frase posterior a A (FAGUNDES, 2024).

Dessa forma, o modelo consegue, até certo ponto, compreender a relação de contexto do idioma através de tais atividades.

Posteriormente, com o modelo pré-treinado, realiza-se um ajuste fino para atender tarefas específicas. O ajuste fino consiste em ajustar o BERT pré-treinado para uma tarefa específica, usando dados rotulados (AKABANE e BARBON, 2022). Por exemplo, para classificar discurso de ódio usando um conjunto de textos retirados de alguma rede social.

Assim, com a disponibilização do modelo BERT pré-treinado, é possível que várias pessoas consigam resolver problemas específicos de processamento de linguagem natural sem grandes investimentos, pois o ajuste fino não exige poder computacional exorbitante, diferente do pré-treinamento.

3.2 Webhook

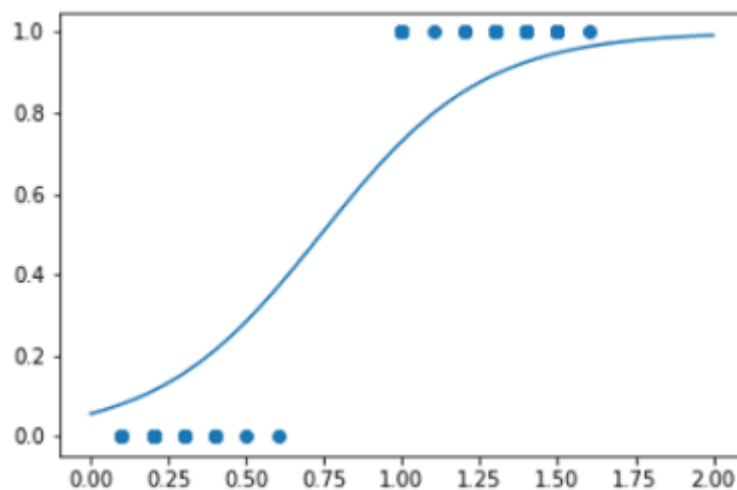
Webhook é um mecanismo de comunicação entre servidores baseado em eventos. Nesse contexto, o servidor dispara uma requisição HTTP caso ocorra um determinado evento, onde o destinatário trata a requisição. A principal diferença entre *Application Programming Interface* (API) e webhook está no modo de emissão, onde a API define um conjunto de serviços e os provê caso seja requisitado explicitamente e o webhook envia a notificação automaticamente em detrimento da ocorrência de eventos (NUGRAHA e SEBASTIAN, 2021).

3.3 Regressão Logística

A regressão logística é um método de aprendizado de máquina que permite relacionar variáveis independentes categóricas ou numéricas com uma variável dependente binária. Nesse viés, o modelo classifica as instâncias em 0 ou 1, retornando também a probabilidade associada. Tal método é possível por conta da transformação logit, a qual iguala o logaritmo

natural da razão de chances com uma regressão linear múltipla. Vale salientar que tal modelo pode ser estendido para versões não dicotômicas, porém o tradicional trata apenas para eventos binários (BITTENCOURT, 2003). A Figura 1 apresenta a transformação citada e a curva característica do modelo.

Figura 1 - Curva Característica da Regressão Logística



Fonte: HashTag Treinamentos

<https://www.hashtagtreinamentos.com/wp-content/uploads/2022/12/REGRESSAO-LOGISTICA-3.png>

A Figura 2 apresenta as equações envolvidas na transformação Logit.

Figura 2 - Equações da Transformação Logit

$$\text{logit}(Y) = \ln\left(\frac{Y}{1-Y}\right) = b_0 + b_1X_1 + b_2X_2 + \dots$$

$$Y = \frac{e^{(b_0+b_1X_1+b_2X_2+\dots)}}{1+e^{(b_0+b_1X_1+b_2X_2+\dots)}}$$

Fonte: Material da disciplina ACH2036 - Métodos Quantitativos para Análise Multivariada - [MOA-RegressaoLogistica-4.pdf](#)

É possível, por exemplo, analisar quais são os fatores que influenciam o ingresso no ensino superior através de uma regressão logística. Nesse exemplo, pode-se usar várias variáveis independentes como renda familiar e a presença de parentes graduados para prever a probabilidade de ingresso de um estudante no ensino superior. Ademais, como tal método

exige poucas suposições estatísticas, é fácil aplicar esse aprendizado de máquina em diversos contextos.

3.4 API Bot Telegram

A plataforma Telegram oferece diversas APIs para a comunidade de desenvolvedores. Nesse sentido, a API Bot é responsável por fornecer diretrizes para a construção de bots dentro da rede social. Para registrar um bot, deve-se interagir com Botfather dentro da plataforma, onde se informa o nome da ferramenta e obtém o token de acesso. Assim, constrói-se um servidor em alguma linguagem de programação como Java ou Python e estabelece a comunicação da aplicação com o Telegram através de requisições HTTP para a API Bot⁴.

3.5 BERTimbau

Sabe-se que o BERT representa um grande avanço no campo do processamento de linguagem natural. Entretanto, o modelo citado foi pré-treinado em língua inglesa, podendo tal fato afetar o desempenho em outros idiomas. Dessa forma, o BERTimbau foi criado como uma versão do modelo BERT pré-treinada para o português brasileiro (AKABANE e BARBON, 2022) .

3.6 Word Embeddings

Word Embeddings são representações vetoriais utilizadas para processar o significado e a complexidade de palavras da língua natural. Dessa forma, por exemplo, palavras com significado semelhantes estão próximas em um espaço euclidiano. Há vários modelos e

⁴ Telegram APIs: <https://core.telegram.org/#api-methods> (acessado em 01/02/2025)

algoritmos para extrair os word embeddings, onde cada estratégia possui suas forças e deficiências (ALMEIDA E XEXÉO, 2023).

3.7 BERT Word Embeddings

Em oposição a outras metodologias, o BERT gera *word embeddings* contextuais, onde uma mesma palavra pode possuir diferentes representações dependendo do contexto. Assim, pode-se usar a saída do BERT pré-treinado como entrada em outros modelos, por exemplo, em uma regressão logística. Tal estratégia evita a necessidade de realizar o ajuste fino e exige menor poder computacional (FAGUNDES, 2024).

4. Metodologia

Com base na dissertação de Matheus Fagundes (FAGUNDES, 2024), foram comparados os desempenhos dos modelos apresentados em sua dissertação para a identificação de notícias falsas, permitindo a escolha do modelo mais adequado para o presente estudo. Conforme observado na dissertação, modelos que integram informações léxicas e sintáticas não apresentaram um ganho significativo de desempenho em comparação àqueles baseados exclusivamente em informações léxicas, como o BERTimbau com ajuste fino (FAGUNDES, 2024). Dessa forma, optou-se pelo uso do BERTimbau com ajuste fino e do BERTimbau combinado à regressão logística.

Foram desenvolvidos dois modelos para classificar as notícias como falsas ou verdadeiras. O primeiro, BERTimbau com ajuste fino, consiste em treinar o modelo pré-treinado em um conjunto de notícias, ajustando seus parâmetros para prever a veracidade das informações. Já o segundo, BERTimbau combinado à regressão logística, utiliza os embeddings gerados pelo BERTimbau pré-treinado como entrada para um modelo de

regressão logística, eliminando a necessidade do ajuste fino e reduzindo o custo computacional.

Para a geração dos modelos preditivos, o treinamento foi realizado com o corpus FAKE.BR, enquanto o teste final utilizou o corpus FACTCK.BR. O treinamento dos modelos usou a técnica de validação cruzada com 5 folds, com a finalidade de levantar métricas com maior confiabilidade e generalizar os modelos. O teste final foi realizado para evidenciar a necessidade de treinamentos periódicos, uma vez que o desempenho do modelo tende a diminuir quando aplicado a dados diferentes do conjunto original.

O corpus FAKE.BR possui 7.200 notícias, das quais 3.600 são verdadeiras e 3.600 são falsas, representando um conjunto de dados balanceados. Nesse sentido, as notícias do corpus são exclusivamente falsas ou verdadeiras e são categorizadas em seis eixos temáticos: política, TV e celebridades, sociedade, ciência e tecnologia, economia e religião (FAGUNDES, 2024). Por ser um corpus balanceado, usou-se o FAKE.BR para treinamento dos modelos.

O corpus FACTCK.BR contém 1.309 notícias, sendo 943 falsas, 246 parcialmente falsas e 120 verdadeiras (FAGUNDES, 2024). Como o objetivo do estudo é realizar classificações de notícias de forma binária, foram removidas as notícias parcialmente falsas. Diferentemente do FAKE.BR, o FACTCK.BR apresenta um desbalanceamento entre classes, o que exige uma interpretação cuidadosa das métricas de desempenho.

Ademais, foi utilizada a plataforma Telegram para aplicar a detecção de notícias falsas no presente estudo. Tal escolha foi tomada pelo fato da API Bot Telegram ser gratuita e bem documentada, além da grande relevância que a plataforma possui. Vale ressaltar que os modelos desenvolvidos podem ser aplicados a qualquer mídia social, bastando apenas realizar algumas adaptações para cada API específica.

Os modelos foram desenvolvidos em Python versão 3.12.4, usando parâmetros padrão das implementações. O ajuste fino foi realizado em GPU utilizando a plataforma Google Colab, enquanto o restante foi processado em CPU. A biblioteca *transformers* facilitou o uso do modelo BERTimbau e foi usada a biblioteca sklearn para instanciar a regressão logística e avaliar o desempenho dos modelos. O código fonte, disponibilizado em um repositório Github⁵, permite a reprodução dos resultados.

5. Resultados e Discussões

Foram produzidos dois modelos, treinados e inicialmente avaliados usando validação cruzada com 5 folds sobre o FAKE.BR. Disponibilizou-se os modelos através de classes da linguagem Python, onde qualquer indivíduo consegue reutilizar os preditores. Vale ressaltar que a classe positiva são as notícias verdadeiras. A seguir, será apresentado o desempenho de cada modelo construído.

5.1 BERTimbau com ajuste fino

Foi realizado um ajuste fino do modelo pré-treinado BERTimbau, com o objetivo de prever a veracidade das notícias. Nesse contexto, realizou-se uma validação cruzada com 5 folds sobre o corpus FAKE.BR. Seguindo a orientação do autor do corpus, usou-se os textos truncados, visto que há diferenças de tamanho entre notícias verdadeiras e falsas, o que pode causar viés. Para avaliação, a acurácia é confiável por conta da natureza balanceada do FAKE.BR.

A Tabela 1 apresenta os resultados de acurácia, revocação, precisão F1 e macro-F1 obtidos pelo modelo para cada um dos *folds*.

⁵ GitHub do projeto: <https://github.com/Marcos-Med/IC> (acessado em 04/02/2025)

Tabela 1 - Resultado do desempenho da classificação usando BERTimbau com ajuste fino

Fold	Acurácia	Revocação	Precisão	F1	Macro-F1
0	0,9923	0,9875	0,9972	0,9923	0,9923
1	0,9993	0,9985	1,0000	0,9992	0,9993
2	1,0000	1,0000	1,0000	1,0000	1,0000
3	0,9993	1,0000	0,9986	0,9993	0,9993
4	0,9993	0,9986	1,0000	0,9993	0,9993
Média	0,9980	0,9969	0,9991	0,9980	0,9980

O BERTimbau com ajuste fino apresentou um resultado surpreendentemente alto, demonstrando alta capacidade de aprendizado. Percebe-se que não ocorre diferença entre as métricas F1 e Macro-F1, indicando boa capacidade em prever ambas as classes de notícias.

5.2 BERTimbau combinado à regressão logística

Os embeddings do BERTimbau pré-treinado foram utilizados como entrada para uma regressão logística, responsável pela classificação das notícias. Nesse sentido, treinou-se a regressão logística com os embeddings, utilizando a técnica de validação cruzada com 5 folds sobre o FAKE.BR. Como já salientado anteriormente, usou-se os textos truncados, com o objetivo de evitar criar viés. Pela natureza do corpus, a acurácia é métrica robusta para avaliação.

A Tabela 2 apresenta os resultados de acurácia, revocação, precisão F1 e macro-F1 obtidos pelo modelo.

Tabela 2 - Resultado do desempenho da classificação usando regressão logística e embedding do BERTimbau

	Acurácia	Precisão	Revocação	F1	Macro-F1
Média	0,9683	0,9672	0,9695	0,9683	0,9683

Conforme os resultados, o modelo com regressão logística apresentou um desempenho semelhante ao preditor anterior, mas levemente inferior. Visto que foi obtido um alto desempenho com menos recursos computacionais, esta abordagem pode ser considerada vantajosa em alguns contextos. Tal modelo também não favoreceu nenhuma classe específica, pois o F1 e o Macro-F1 são parecidos.

5.3 Teste final

Para verificar o comportamento dos modelos em dados diferentes do conjunto original, aplicou-se um teste sobre o corpus FACTCK.BR. Como tal conjunto possui desbalanceamento entre as classes, logo a acurácia não é uma métrica confiável. Pela natureza citada, é notória a importância do uso das métricas de F1 e Macro-F1. Vale ressaltar que a classe positiva são as notícias verdadeiras.

A Tabela 3 apresenta os resultados de acurácia, revocação, precisão F1 e macro-F1 obtidos pelo modelo dos dois modelos treinados utilizando o conjunto FACTCK.BR.

Tabela 3 - Resultado do desempenho da classificação para o conjunto FACTCK.BR

Modelo	Acurácia	Precisão	Revocação	F1	Macro-F1
BERTimbau com ajuste fino	0,4714	0,1264	0,6218	0,2102	0,4065
BERTimbau com regressão logística	0,6730	0,1451	0,3865	0,2110	0,5023

Analisando os resultados, ambos os modelos apresentaram desempenho inferior em relação aos primeiros testes, indicando que os modelos possuem dificuldades em dados

distintos do corpus de treinamento. Esta queda de desempenho já havia sido relatada na literatura. A métrica Macro-F1 foi superior ao F1 para ambos modelos devido à natureza do FACTCK.BR, o qual possui uma grande quantidade de notícias falsas. Dessa forma, com tal desbalanceamento, os modelos favorecem a classe predominante, que é a classe negativa.

Tal cenário negativo é esperado, pois os estilos de notícias são diversos e variam de acordo com o tempo e o assunto. Para que o modelo tivesse um bom desempenho em qualquer notícia, exigiria que o preditor possuísse um grande repertório. Para contornar a problemática vista, é necessário realizar treinamentos periódicos, com a finalidade de aumentar a capacidade do modelo.

5.4 Bot Telegram

Foi desenvolvido um Bot na rede social Telegram, o qual prevê a veracidade da notícia quando solicitado. Nesse sentido, o Bot foi construído utilizando a linguagem Python, integrando o Bot e a plataforma por meio de um webhook. O Bot pode ser utilizado tanto em grupos quanto de forma independente.

Para adicionar o Bot a um grupo no Telegram, é necessário apenas acessar a seção “Adicionar membros” e pesquisar pelo nome do Bot, neste caso, “FakeBot”. Após a adição, qualquer membro do grupo poderá utilizar suas funcionalidades. Para interações individuais, basta pesquisar pelo nome do Bot na plataforma e iniciar uma conversa.

Foram desenvolvidos três comandos: “/start”, “/help” e “/prever [notícia]”. O primeiro comando solicita uma breve apresentação do Bot, descrevendo seu papel no combate à desinformação. O segundo lista todos os comandos disponíveis com suas respectivas finalidades para os usuários. Por fim, o terceiro comando verifica a veracidade do conteúdo, informando a classificação em conjunto com a probabilidade.

As Figuras 3, 4 e 5 apresentam cópias de tela que apresentam as funcionalidades do Bot. Desde sua apresentação (Figura 3), o resultado da execução do comando “/help” (Figura 4) e a apresentação do veredicto em relação a uma notícia (Figura 5).

Figura 3 - O uso do comando “/start”

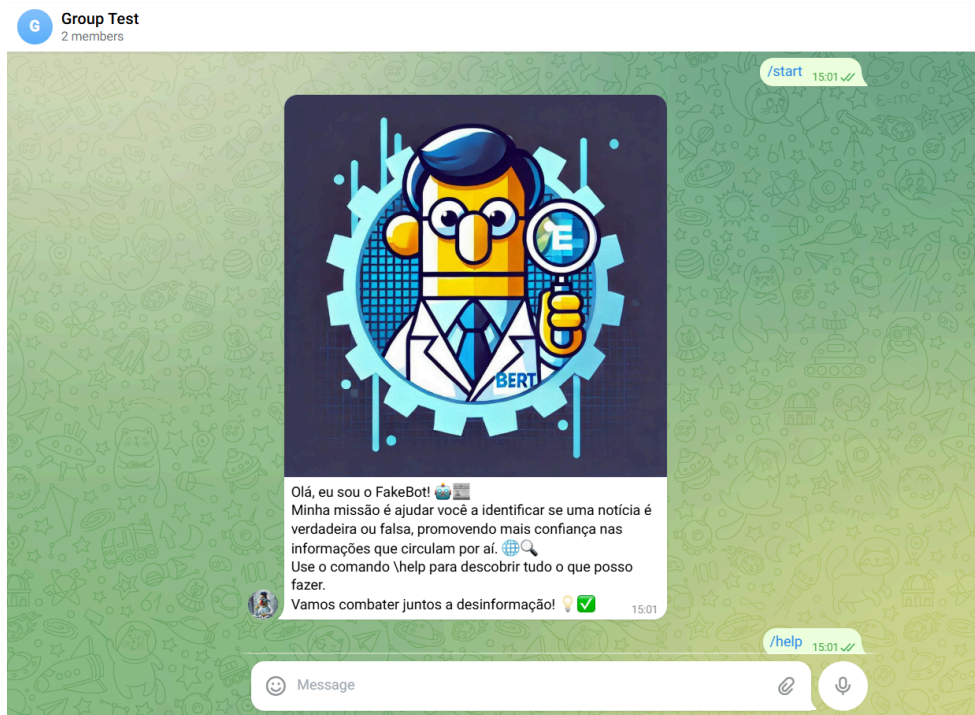


Figura 4 - O uso do comando “/help”

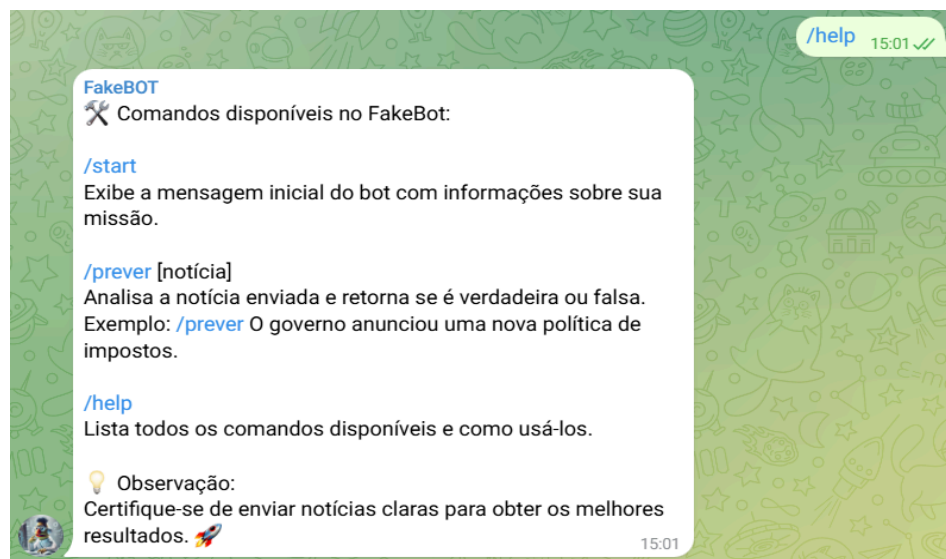
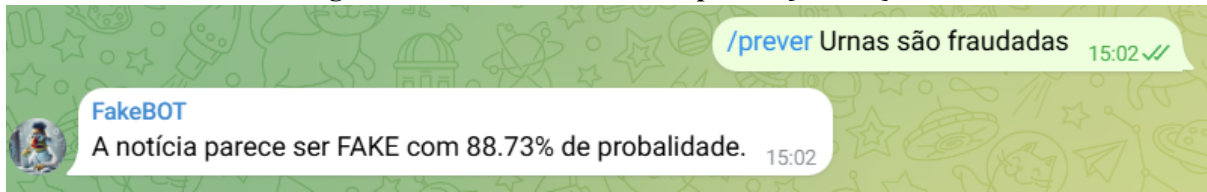


Figura 5 - O uso do comando “/prever [notícia]”



Consideramos que o Bot obteve um bom funcionamento e conseguiu atender vários usuários e grupos simultaneamente. Foram testados ambos modelos dentro do Bot, mas o desempenho de predição varia de acordo com o modelo e de características da notícia testada. Vale ressaltar que os modelos construídos podem ser reaproveitados a outras plataformas de mensagens, bastando somente realizar adaptações do Bot de acordo com a API específica.

6. Considerações finais

Conforme discutido, a propagação de notícias falsas é nociva para vários campos da sociedade, como democracia e saúde. Para mitigar a problemática, a criação de mecanismos automáticos de detecção de notícias falsas é válida, pois a verificação manual de conteúdo é árdua e impraticável. Com isso, o presente trabalho teve como objetivo contribuir para o desenvolvimento de modelos capazes de identificar *fake news* em plataformas de mensagens.

Como resultado, foram obtidos modelos de alto desempenho no corpus FAKE.BR, com resultados similares entre si. Entretanto, tais modelos tendem a enfrentar dificuldades em dados diferentes do corpus original, por conta da variedade de estilos e gêneros presentes nas notícias. Para contornar tal entrave, deve-se realizar treinamentos periódicos, com o objetivo de aumentar o repertório dos modelos.

Foram testados ambos os modelos dentro da plataforma Telegram, utilizando a linguagem Python. Observou-se um bom desempenho do Bot construído, onde a ferramenta conseguiu atender múltiplos usuários e grupos simultaneamente.

Para pesquisas futuras, é indicado testar outros tipos de informação como semânticas, para verificar o impacto sobre a generalização dos modelos para novos dados. Além disso, é interessante adicionar funcionalidades à ferramenta, como a capacidade de indicar sites confiáveis que estão comentando sobre a notícia solicitada. Com tais pesquisas, espera-se que haja uma diminuição dos impactos negativos causados pela disseminação de desinformação.

7. Referências

Almeida, F., & Xexéo, G. (2019). Word embeddings: A survey. *arXiv preprint <https://arxiv.org/abs/1901.09069>*.

Barcelos TN, Muniz LN, Dantas DM, Cotrim Junior DF, Cavalcante JR, Faerstein E. (2021) Análise de fake news veiculadas durante a pandemia de COVID-19 no Brasil. *Rev Panam Salud Publica*; 45:e65. <https://doi.org/10.26633/RPSP.2021.65>

Bittencourt, H. R. (2003). Regressão logística politômica: revisão teórica e aplicações. *Acta Scientiae*, 5(1), 77-86. <http://www.periodicos.ulbra.br/index.php/acta/article/view/146>

Fagundes, Matheus José Garcia (2024). Combinando características léxicas e morfossintáticas para a detecção de notícias falsas em português. Dissertação de Mestrado. Programa de Sistemas de Informação, Universidade de São Paulo.

Hunt, Elle (2016). What is fake news? How to stop it and what you can do to stop it. *The Guardian*. Disponível em: <https://www.usnews.com/opinion/thomas-jefferson-street/articles/2017-04-14/what-is-fake-news-maybe-not-what-you-think> acessado em 20/01/2025

Nugraha, K. A., & Sebastian, D. (2021). Designing consultation chatbot using telegram api and webhook-based nodejs applications. In *7th International Conference on Education and Technology (ICET 2021)* (pp. 119-122). Atlantis Press. <https://doi.org/10.2991/assehr.k.211126.047>

Resende, Gustavo; Messias, Johnnatan; Silva, Márcio; Almeida, Jussara; Vasconcelos, Marisa; Benevenuto, Fabrício (2018). A System for Monitoring Public Political Groups in WhatsApp. In *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web (WebMedia '18)*. ACM, New York, NY, USA, 387-390. DOI: <https://doi.org/10.1145/3243082.3264662>

Schlesinger, Robert (2017). Fake News in Reality. *U.S. News & World Report*. Disponível em: <https://www.usnews.com/opinion/thomas-jefferson-street/articles/2017-04-14/what-is-fake-news-maybe-not-what-you-think> acessado em 20/05/2019

Sharma, Karishma; Qian, Feng; Jiang, He; Ruchansky, Natali; Zhang, Ming; Liu, Yan (2019). Combating Fake News: A Survey on Identification and Mitigation Techniques. *ACM Trans. Intell. Syst. Technol.* 10, 3, Article 21 (April 2019), 42 pages. DOI: <https://doi.org/10.1145/3305260>

Silva Barbon, R.; Akabane, A. T. (2022). Towards transfer learning techniques—bert, distilbert, bertimbau, and distilbertimbau for automatic text classification from different languages: A case study. *Sensors*, 22(21), 8184. <https://doi.org/10.3390/s22218184>

Vicario, Michela Del; Quattrociocchi, Walter; Antonio Scala; Zollo, Fabiana (2019). Polarization and Fake News: Early Warning of Potential Misinformation Targets. *ACM Trans. Web* 13, 2, Article 10, 22 páginas. DOI: <https://doi.org/10.1145/3316809>

Zhou, Xinyi; Zafarani, Reza (2019). Fake News Detection: An Interdisciplinary Research. In *Companion Proceedings of The 2019 World Wide Web Conference (WWW '19)*, Ling Liu and Ryen White (Eds.). ACM, New York, NY, USA, 1292-1292. DOI: <https://doi.org/10.1145/3308560.3316476>

Zhou, Xinyi; Zafarani, Reza; Shu, Kai; Liu, Huan (2019). Fake News: Fundamental Theories, Detection Strategies and Challenges. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19)*. ACM, New York, NY, USA, 836-837. DOI: <https://doi.org/10.1145/3289600.3291382>