Detecção de Tweets Tóxicos em pt-BR

Eduardo Barros Innarelli
170161
e170161@dac.unicamp.br

João Pedro C. Martins
176117
j176117@dac.unicamp.br

Pedro Alan T. Ramos
185531
p185531@dac.unicamp.br

***Resumo*—Nos últimos anos, a quantidade de comentários tóxicos em redes sociais vem crescendo firmemente. Além de causar injúrias, esses comentários são capazes até mesmo de movimentar o cenário político global. Mais especificamente, os internautas brasileiros estão entre os mais ativos nas redes sociais, gerando grande parte dos comentários nessas plataformas. Dado a inviabilidade de se combater esse tipo de conteúdo de forma manual, a identificação de comentários tóxicos utilizando métodos de *machine learning* é um passo natural para a solução desse problema. Apesar do tema ser relativamente bem explorado em línguas como a inglesa, que dispõe de uma grande quantidade de dados para treinar modelos para identificar comentários tóxicos, a língua portuguesa é uma exceção. Até 2019, a maior base de dados disponível nesse tema para a língua portuguesa continha pouco mais de 5 mil exemplos. Foi apenas em 2020 que surgiu uma base de dados aproximadamente quatro vezes maior, o ToLD-Br, com 21 mil exemplos. Neste trabalho, utilizamos um modelo linear de classificação (NBSVM) como baseline para a tarefa de classificação da toxicidade de comentários em português do Brasil e comparamos com os resultados obtidos após realizar um *fine-tuning* em um modelo BERT pré-treinado para pt-BR, o BERTimbau, para a mesma tarefa. Em ambos experimentos, utilizamos o ToLD-Br para o treinamento. Também comparamos a performance do nosso modelo monolíngue com outros trabalhos que utilizaram o BERT multilíngue. Obtivemos resultados muito superiores ao baseline e ao BERT monolíngue, utilizado para a mesma tarefa em outros trabalhos. Em específico, o valor de *recall* para comentários tóxicos, o que seria importante para uma aplicação de filtragem, foi superior em relação ao BERT multilíngue, com um valor de 83%, em comparação com os 82% encontrados em outros trabalhos.**

***Palavras-chave*—** *NLP, Sentiment Analysis, Deep Learning, NBSVM, BERT, BERTimbau*

I. Introdução

As redes sociais oferecem um ambiente propício para o cultivo de comunidades tóxicas que, muitas vezes no anonimato, se sentem à vontade para atacar indivíduos ou grupos de indivíduos (em especial minorias) sem serem penalizadas. Além de isso prejudicar diretamente as vítimas dos ataques, a disseminação e naturalização do discurso de ódio em redes sociais é capaz de impactar inclusive o cenário político global, vide a ascensão na última década de grupos de extrema-direita respaldados por essas comunidades.

Como afirma Luiz Trindade [9], "discursos depreciativos exploram e amplificam diferenças percebidas entre grupos raciais a fim de ressaltar atributos de cunho negativo dos 'outros' (p. ex, perpetrador, sem escolarização, não atraente, etc.) e, em contraste, reafirmar a condição normativa e privilegiada do grupo hegemônico". O autor se refere a difusão de discursos racistas nas redes sociais, mas é possível estender seu argumento para discursos que refletem as demais estruturas de opressão. Trindade também contrapõe a crença de que os ambientes virtual e *off-line* são dissociados um do outro, mito que contribui para que as pessoas descarreguem livremente tais discursos em ambiente virtual e evitem manifestá-los em público. Sendo diretamente responsável pelo desenvolvimento desses meios de comunicação, é também dever do cientista da computação estar atento a essas questões, visando uma sociedade mais justa e igualitária.

Monitorar comentários tóxicos não é uma questão simples, tanto pelo grande volume e variedade de injúrias, quanto pela subjetividade envolvida em determinar se uma postagem é, de fato, tóxica. Técnicas de processamento de linguagem natural (PLN) podem ajudar a automatizar esse monitoramento, não a toa existindo competições e vários conjuntos de dados voltados ao tema. Por conta da relevância do problema e da disponibilidade de datasets recentes, achamos pertinente abordá-lo neste projeto. Em específico, focamos no reconhecimento de comentários danosos em português do Brasil, com o intuito de aproximar o estudo ainda mais da nossa realidade.

Este trabalho encontra-se organizado da seguinte forma: na Seção II, apresentamos alguns trabalhos anteriores que abordam o problema de classificação de comentários ofensivos na língua portuguesa, culminando na pesquisa de Leite et al. [5], que desenvolveu a base de dados utilizada neste trabalho, e no modelo BERT disponibilizado por Souza et al. [8], que é o modelo principal utilizado neste trabalho; na Seção III, apresentamos as características principais da base de dados utilizada; na Seção IV, explicamos de um ponto de vista mais teórico o NBSVM (*Naive Bayes - Support Vector Machine*) e o BERT, as duas técnicas escolhidas para resolver o problema; na Seção V, detalhamos como esses modelos foram implementados e os resultados obtidos com os mesmos; finalmente, na Seção VI, relatamos nossas conclusões e ideias para trabalhos futuros.

II. Trabalhos Relacionados

Embora vários pesquisadores tenham abordado o tópico de discurso de ódio, focamos a revisão da literatura em trabalhos anteriores relacionados à detecção de comentários tóxicos, o tópico de nosso trabalho, com atenção especial a trabalhos que tratam de bases de dados na língua portuguesa.

O problema de identificação de comentários tóxicos utilizando métodos de processamento de linguagem natural é pouco explorado na língua portuguesa. Até 2019, a maior base de dados disponível para o tema em português foi a base criada

por Fortuna et al. [3], com 5668 exemplos. O trabalho que apresenta essa base de dados estabelece que técnicas baseadas em *machine learning* requerem bases de dados anotadas suficientemente grandes e que a língua portuguesa não tem estado em foco no assunto. De fato, algumas línguas, como a inglesa, têm à sua disposição base de dados grandes desde 2017. Um dos exemplos seria a base de dados com mais de 100 mil exemplos disponibilizada por Wulczyn et al. [10]. Apesar de ser possível transferir os estados internos aprendidos de um modelo treinado em uma língua e obter bons resultados em tarefas como *named entity recognition (NER)* e *question answering* em uma outra língua (Pires et al. [7]), resultados obtidos por Leite et al. [5] mostram que a utilização de dados monolíngues em larga escala é essencial para obtenção de resultados superiores.

Em Outubro de 2020 foi criada a *Toxic Language Dataset For Brazilian Portuguese (ToLD-Br)*, uma base de dados anotada para identificação de comentários tóxicos. Essa base contém 21 mil tweets em português do Brasil, cerca de quatro vezes mais dados do que a base de dados criada por Fortuna et al. [3]. O ToLD-Br foi introduzido com o paper por Leite et al. [5], que explora o problema de identificação de comentários tóxicos na base de dados estudando o impacto nos resultados devido à utilização de dados monolíngues em larga escala. Para isso é feita uma comparação de desempenho de modelos treinados apenas com dados em outras línguas e modelos que receberam *fine-tuning* utilizando a ToLD-Br. Essa comparação é feita resolvendo o problema de classificação binária de comentários em tóxicos ou não-tóxicos, utilizando modelos *BERT*. Com o modelo que não teve acesso a nenhum dado em português do Brasil, utilizando *zero-shot learning*, conseguiram uma *precision* de 0.61, um *recall* de 0.60 e um *F1-score* de 0.57 (valores da *weighted average*). No entanto, com o modelo multilíngue que recebeu *fine-tuning* com a ToLD-Br, conseguiram uma *precision* de 0.76, um *recall* de 0.75 e um *F1-score* de 0.75 (valores da *weighted average*). Com esses resultados, o paper demonstra a importância da utilização de uma base de dados monolíngue em larga escala para o treinamento de modelos com o intuito de, por exemplo, fazer a filtragem automática de comentários tóxicos em uma plataforma de rede social, uma vez que os resultados obtidos sem o uso de tal base não seriam satisfatórios.

O BERT multilíngue (Devlin et al. [2]) é um modelo considerado *state-of-the-art* para resolver várias tarefas em processamento de linguagem natural. Porém, em 2020, Souza et al. [8] disponibilizaram um BERT pré-treinado com a Wikipédia brasileira e com o BrWaC (Brazilian Web as Corpus), um grande corpus em português do Brasil obtido por *web crawling*. Os autores nomearam o modelo de BERTimbau, que foi avaliado em três tarefas *downstream* de PLN: *sentence textual similarity*, *recognizing textual entailment* e *named entity recognition*. Segundo os autores, o modelo BERTimbau supera o BERT multilíngue *state-of-the-art* em todas essas tarefas, confirmando a efetividade de grandes modelos pré-treinados para português.

O intuito do nosso trabalho é se apoiar nessas duas novidades promissoras que cercam o tema a ser explorado para extrair resultados novos. Esperamos que, com o *fine-tuning* do modelo BERTimbau utilizando a base de dados ToLD-Br, possamos atingir resultados ainda não vistos para o problema de classificação binária de comentários tóxicos em português do Brasil.

III. Base de Dados

Utilizamos para o projeto a *Toxic Language Dataset for Brazilian Portuguese (ToLD-Br)*, a maior base de dados disponível para análise de comentários tóxicos em redes sociais para a língua portuguesa. No restante dessa seção faremos uma síntese das informações fornecidas no trabalho de Leite et al. [5].

O ToLD-Br consiste de 21 mil tweets (comentários da plataforma Twitter) anotados manualmente em sete categorias: não-tóxico, LGBTQfobia, obsceno, insulto, racismo, misoginia, e xenofobia. Os 21 mil tweets foram coletados de Julho à Agosto de 2019 utilizando o GATE Cloud's Twitter Collector [4]. Desses tweets, 12.600 (60%) foram obtidos por meio de buscas por palavras-chave específicas (e.g. gay, mulherzinha, nordestino) que levariam à tweets com alta chance de conter conteúdo tóxico. Os demais tweets foram obtidos por meio de buscas por menções à usuários influenciadores (e.g. Jair Bolsonaro, Neymar Jr.).

Para a anotação dos tweets, foram selecionados 42 candidatos dentre 129 alunos da Univerdade Federal de São Carlos (UFSCar), baseado na informação demográfica dos candidatos e com o intuito de balancear o bias nas anotações. A classificação demográfica foi realizada pelos próprios candidatos levando em conta sexo, orientação sexual e etnia, com as opções sendo retiradas do Instituto Brasileiro de Geografia e Estatística (IBGE). Também foi disponibilizado aos candidatos a opção de não declarar algum desses aspectos. Embora os autores do paper tenham tentado manter os aspectos demográficos dos anotadores balanceados, a maioria dos voluntários se identificavam como *brancos* e *heterossexuais*. Sexo foi uma característica que se mostrou balanceada. A idade dos anotadores variam entre 18 e 37 anos, com a maioria deles em um intervalo de 19 e 23 anos. A tabela I apresenta os dados demográficos e a figura 1 apresenta a distribuição de idade de todos os anotadores.

Cada candidato anotou 1.500 tweets em uma ou mais das seguintes categorias: LGBTQ+fobia, obscenidade, insulto, racismo, misoginia e xenofobia. Caso nenhuma categoria se aplicasse, o anotador as deixava em branco. Além disso, cada tweet foi anotado por 3 diferentes candidatos. Foi utilizada o α *de Krippendorf* [1] para comparar os diferentes níveis de *agreement* entre os anotadores, ou seja, quais categorias tiveram mais concordância para a anotação dos tweets e quais tiveram mais divergência de anotação. A tabela II mostra estes valores.

A categoria de *LGBTQ+fobia* obteve o maior nível de concordância, talvez indicando que tweets nessa classe utilizem um vocabulário mais específico. No entanto, as classes de

Tabela I
INFORMAÇÕES DEMOGRÁFICAS DOS ANOTADORES.

	Categorias	# anotadores
Sexo	Masculino	18
	Feminino	24
Orientação sexual	Heterossexual	22
	Bissexual	12
	Homossexual	5
	Pansexual	3
Cor ou raça	Branco	25
	Pardo	9
	Preto	5
	Amarelo	2
	Não-declarado	1

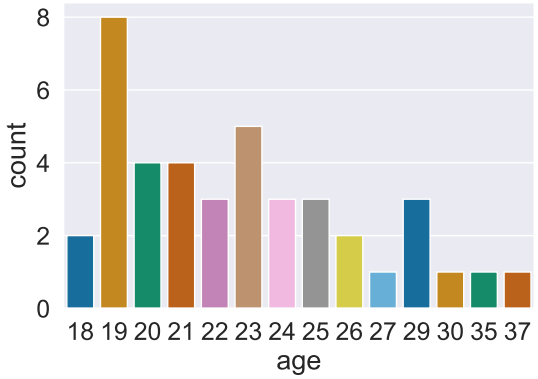

Figura 1. Distribuição de idade dos anotadores.

Tabela II
α DE KRIPPENDORFF PARA CADA CATEGORIA.

	α
LGBTQ+fobia	0.68
Insulto	0.56
Xenofobia	0.57
Misoginia	0.52
Obsceno	0.49
Racismo	0.48
Média	0.55

racismo e *obsceno* obtiveram os menores índices de concordância.

A base de dados em si está disponível em duas diferentes variações. Na primeira, cada tweet possui os dados de quantas vezes ele foi anotado para cada categoria, um valor que vai de 0 a 3. Na segunda, as anotações não estão agregadas para cada categoria, então os valores são de 0 a 1 para cada categorização que cada anotador deu para o tweet.

IV. METODOLOGIA

Neste trabalho, propomos resolver o problema de classificação binária de tweets como **tóxicos** ou **não-tóxicos**, utilizando a mesma metodologia de Leite et al. [5], porém, utilizando o BERTimbau [8] como modelo do tipo BERT [2], pré-treinado para português do Brasil.

É importante ressaltar que, assim como realizado no trabalho utilizado como base [5], um tweet foi agrupado como tóxico no nosso modelo binário se ao menos um anotador o tiver anotado em uma das categorias tóxicas.

A. NBSVM

A técnica que utilizamos como baseline para o problema de classificação de tweets do dataset ToLD-Br é chamada NBSVM (Naive Bayes - Support Vector Machine). O ponto principal da técnica é o Naive Bayes, um algoritmo de aprendizado supervisionado probabilístico usado para classificação.

Nesse algoritmo, a classificação é baseada no teorema de Bayes, com a suposição de que as features, que no caso de classificação de textos são palavras ou *n-grams*, são independentes. Isto é, a probabilidade da presença de uma palavra não afeta a probabilidade da presença de outra. Isso geralmente não é verdade, daí o nome *Naive* do algoritmo. Geralmente, as features de um problema de NLP são palavras. No Naive Bayes, para uma dada palavra f, é utilizada uma razão r definida por:

$$r = log \frac{\text{proporção da feature f em documentos positivos}}{\text{proporção da feature f em documentos negativos}}$$

onde documentos são textos ou, no caso do nosso problema, tweets. Essa razão r é chamada de *log-count* ratio e é ela que o algoritmo usa para para fazer a classificação de um tweet em tóxico ou não-tóxico, já que ela é relacionada com o quão frequentemente cada palavra aparece em tweets tóxicos, ou seja, qual o peso de cada palavra na toxicidade de um tweet.

Quanto mais palavras com alto *log-count ratio* um tweet conter, mais provável é que esse tweet seja tóxico. Porém, como mencionado anteriormente, o Naive Bayes é *naive* em assumir que as palavras nos textos são independentes, o que significa que os *log-count ratios* calculados pelo algoritmo são apenas estimativas nesse sentido.

O **NBSVM** funciona unindo as forças do Naive Bayes, que é um *generative classifier*, e de um modelo linear como o SVM com kernel linear, que é um *discriminative classifier*, criando um modelo híbrido que é um ótimo baseline para classificação de texto. Na prática, o SVM recebe como features os *log-count ratios* do Naive Bayes, para que possa aprendê-los ao invés de utilizar uma estimativa desses valores, que é o que o algoritmo original faz.

B. BERT

Para obter um modelo que proporcione um recall aceitável para uma aplicação real de filtragem de tweets tóxicos, fomos atrás do que é considerado hoje um modelo *state-of-the-art* em processamento de linguagem natural: o BERT (Bidirectional Encoder Representations from Transformers) [2].

A inovação técnica do BERT está no treinamento bidirecional utilizando o Transformer, um modelo popular de *attention* que aprende relações contextuais entre palavras em um texto. Em sua forma original, o Transformer utiliza duas ferramentas. A primeira é um *encoder* que lê as entradas de texto e a segunda é um decoder que produz uma predição que depende

da tarefa sendo realizada. Como o BERT tem como objetivo gerar um modelo de linguagem, somente o mecanismo de *encoder* é necessário.

Enquanto modelos anteriores conseguiam manter o contexto do texto em apenas uma direção, o BERT, devido à aplicação de treinamento bidirecional do Transformer, é capaz de manter um contexto bidirecional do texto que processa. Isso se deve ao fato de que, no BERT, a sequência de palavras do texto é lida de uma vez, ao invés de palavra por palavra.

Como o BERT tem como objetivo gerar um modelo geral de linguagem, o modo como esse modelo será utilizado é chamado de *fine-tuning*. Esse *fine-tuning* pode ser tanto a adição de layers na rede, como a alteração de hiper-parâmetros específicos, detalhados pelos autores do paper original [2].

Na elaboração do nosso trabalho, utilizamos um modelo BERT pré-treinado para português do Brasil, o BERTimbau. Nele, adicionamos uma nova camada de neurônios não treinados no final do modelo de modo a adaptar o BERTimbau ao problema de classificação binária dos tweets. Para isso, treinamos essa nova camada com os dados do ToLD-Br. Esse processo de modificação do modelo e treinamento com a base de dados é considerado o *fine-tuning* do BERTimbau.

V. Experimentos

Para implementação, foi utilizada a linguagem de programação *Python*, com auxílio das bibliotecas externas `Numpy` e `Pandas` para manipulação e análise de dados. Nas rotinas de pré-processamento e treinamento do NBSVM, utilizamos as bibliotecas `NLTK` e `Scikit-Learn`, sendo a segunda também aplicada no cálculo de métricas. O fine-tuning do BERT foi implementado com os pacotes `PyTorch` e `Hugging Face Transformers`, conforme sugere o tutorial de McCormick et al. [6]. Enfim, para centralizar o desenvolvimento e executar o código em uma GPU (Tesla K80), utilizamos o ambiente *Google Colaboratory*.

A. Pré-processamento

Na Seção III, explicamos como as classes de ofensa são agrupadas em uma única label para tornar o problema em um de classificação binária dos tweets. Feito isso, é preciso pré-processar os dados para alimentar os modelos. Em ambos o primeiro passo consiste em tokenizar os textos, i.e., quebrar os textos em tokens de palavras. Dai em diante o pré-processamento exigido pelas técnicas difere.

Para converter os tokens em uma matriz documento-termo, entrada aceita pelo NBSVM, utilizamos o método `TfidfVectorizer` do `Scikit-Learn`. Esse calcula o valor tf-idf (abreviação do inglês *term frequency–inverse document frequency*, que significa frequência do termo–inverso da frequência nos documentos) de cada termo no texto. O inverso da frequência aumenta o peso das palavras que ocorrem raramente.

Já na entrada do BERT é preciso incluir alguns tokens especiais: o token [SEP] é adicionado ao fim de cada sentença e o token [CLS] prefixado no início, esse último necessário apenas em tarefas de classificação. O BERT espera que todas as sentenças tenham um tamanho fixo: textos maiores são truncados e textos menores são estendidos com tokens [PAD], ignorados no treinamento por máscaras de atenção. Cada token é, então, mapeado a um identificador numérico. Todo esse processo é realizado pelo método `encode_plus` do `BertTokenizer` da biblioteca `Hugging Face Transformers`.

Quanto a divisão do conjunto de dados, dedicamos 90% dos tweets para treino e validação e os outros 10% para teste. O tamanho do conjunto de teste, que contém 2100 tweets, equivale ao tamanho do conjunto avaliado por Leite et al. [5], o que permite que comparemos o desempenho da nossa implementação com o dos autores. Os métodos de separação do `Scikit-Learn` e do `PyTorch` garantem uma divisão balanceada dos dados, no que diz respeito as labels de toxicidade.

B. Treinamento do NBSVM

Um ponto importante a se notar é que, no NBSVM, o SVM com kernel linear pode ser trocado por um modelo semelhante, como o de Regressão Logística. Inclusive, a implementação de Regressão Logística do `Scikit-Learn` utiliza internamente a *Library for Large Linear Classification* (`liblinear`), que suporta ambos os modelos. Na nossa implementação, então, foi utilizado um modelo de Regressão Logística para aprender os *log-count ratios*, e não um SVM.

A performance do NBSVM depende diretamente de alguns hiperparâmetros, definidos tanto no nível de pré-processamento quanto de treinamento. Dentre os hiperparâmetros se destacam: a aplicação, ou não, de stemização dos tokens (redução de palavras flexionadas a sua raiz); o grau de regularização; e as frequências mínima e máxima para um termo ser não ser ignorado na matriz de documento-termo.

Para evitar testar múltiplos valores manualmente, adotamos a técnica de otimização de hiperparâmetros intitulada *grid search* (busca em grade), implementada na classe `GridSearchCV` do `Scikit-Learn`. O modelo é treinado com todas as possíveis combinações de hiperparâmetros definidos em um dicionário, tal que o melhor modelo encontrado é salvo na classe. Cada treinamento para em 1000 épocas. O método do `Scikit-Learn` também aplica validação cruzada, estratégia que ajuda a avaliar se o modelo generaliza bem.

O melhor modelo treinado na busca em grade alcançou uma acurácia de 72% no conjunto de teste, o que indica que o modelo foi razoavelmente eficaz em classificar os tweets. As demais métricas e a matriz de confusão nesse conjunto são apresentadas, respectivamente, na Tabela III e na Figura 2. A porcentagem de falsos negativos é um pouco preocupante, o que é reforçado pelo *recall* baixo de 62% dos tweets ofensivos e pela matriz de confusão. Em uma aplicação real, a não classificação de tweets tóxicos iria tornar ineficaz uma filtragem desse tipo de conteúdo.

C. Fine-tuning do BERTimbau

Seguimos os passos do tutorial de McCormick et al. [6] para realizar o *fine-tuning* de um BERT pré-treinado - o BERTim-

Tabela III
MÉTRICAS DO NBSVM

	Precision	Recall	F1-Score
0	0.73	0.80	0.77
1	0.71	0.62	0.66
Macro Avg	0.72	0.71	0.71
Weighted Avg	0.72	0.72	0.72

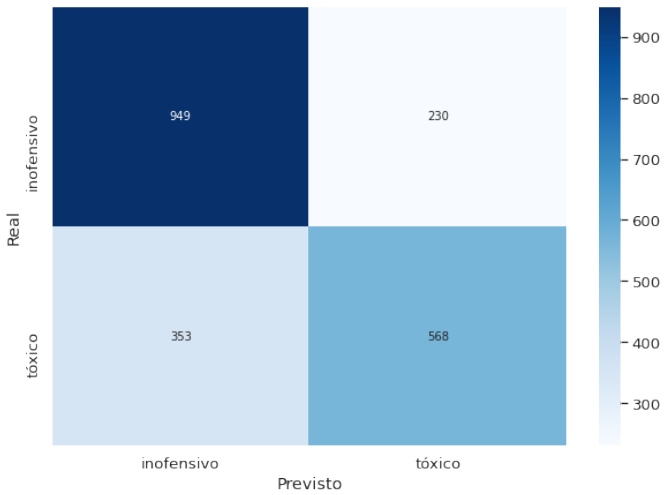

Figura 2. Matriz de confusão do NBSVM.

Tabela IV
MÉTRICAS DO BERTIMBAU

	Precision	Recall	F1-Score
0	0.86	0.76	0.80
1	0.71	0.83	0.76
Macro Avg	0.78	0.79	0.78
Weighted Avg	0.80	0.79	0.79

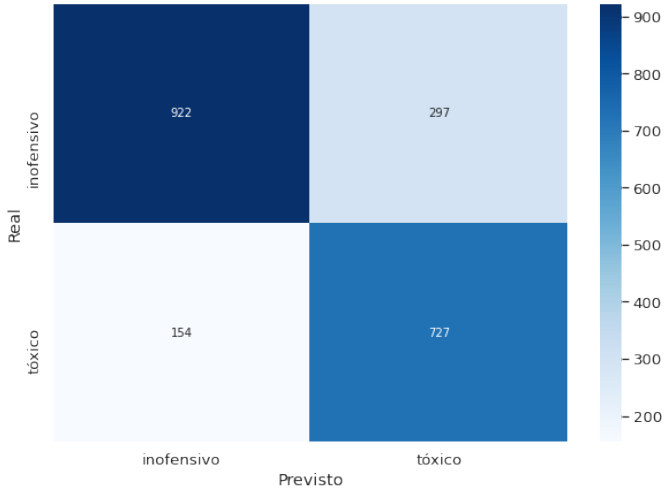

Figura 3. Matriz de confusão do BERTimbau.

bau, no nosso caso - em uma tarefa de classificação binária. A interface `BertForSequenceClassification` da biblioteca `Hugging Face Transformers` inclui no modelo a camada extra de neurônios não treinados usada para a classificação.

Após alguns experimentos iniciais, optamos por treinar apenas 2 épocas, pois foi possível observar que o modelo sofre *overfitting* quando isso é ultrapassado. A RAM do *Google Colaboratory* não suportou treinar *batches* com 32 exemplos, por isso fixamos o *batch size* em 16. Já para a otimização da descida do gradiente, utilizamos o otimizador *Adam* com correção de queda dos pesos, implementado na classe `AdamW` da `Hugging Face Transformers`, e uma taxa de aprendizado de 2×10^{-3}. Com excessão dessa taxa, todos os hiperparâmetros se encaixam nas sugestões dos autores originais do BERT [2].

O loop de treinamento foi implementado manualmente, sendo as rotinas do `PyTorch` de *forward* e *backward propagation* chamadas para cada *batch*. Separamos 10% do conjunto de dados original para, ao fim de cada época, executar uma validação simples. Terminado o treinamento, verificamos que a acurácia alcançada pelo modelo no conjunto de teste é de 79%, consideravelmente maior que a do NBSVM. O que mais chama a atenção, no entanto, é a diferença entre o *recall* dos modelos: o BERTimbau acertou 21% a mais dos tweets tóxicos do que o NBSVM. Na Tabela IV e na Figura 3 encontram-se as outras métricas e a matriz de confusão do BERTimbau. Note que o score F1 está próximo de 80%, o que mostra que há um equilibrio entre a precisão e o *recall* do modelo.

Para comparação, incluímos, na Tabela V, as métricas do melhor modelo de Leite et al. [5], um BERT multilíngue *fine-tuned* com a mesma base de dados (ToLD-Br) que foi usada no *fine-tuning* do BERTimbau. Ambos os modelos foram testados no mesmo conjunto de dados. Pelas métricas é possível observar que o BERTimbau atingiu *weighted averages* de *precision*, *recall* e *F1-score* todas .04 acima dos valores atingidos pelo BERT multilíngue.

Tabela V
MÉTRICAS DO MELHOR MODELO DE LEITE ET AL. [5] (BERT MULTILÍNGUE)

	Precision	Recall	F1-Score
0	0.81	0.69	0.75
1	0.69	0.82	0.75
Macro Avg	0.75	0.75	0.75
Weighted Avg	0.76	0.75	0.75

VI. CONCLUSÃO

Como analisado, o modelo mais complexo mostrou-se mais adequado para resolver o problema em questão. Os melhores resultados que o BERTimbau obteve em relação ao NBSVM reforçam o alto grau de subjetividade envolvido em classificar tweets como tóxicos, tarefa que exige uma compreensão de como a sentença está estruturada, e não somente da frequência das palavras.

Além disso, o fine-tuning do BERTimbau, um modelo BERT monolíngue, alcançou resultados superiores ao fine-tuning do BERT relatado no artigo base, um modelo BERT

multilíngue. Ambos foram avaliados com conjuntos de testes do mesmo tamanho, o que indica que o pré-treinamento do BERTimbau é mais robusto do que o das variantes testadas por Leite et al. [5].

Isto posto, o experimento foi um sucesso, atingindo resultados satisfatórios para o problema abordado. Em trabalhos futuros, acreditamos que o mais interessante seja modelar o problema de diferentes maneiras. É possível, por exemplo, ser mais rígido no agrupamento das labels, executar classificações binárias para cada uma das categorias de toxicidade ou até mesmo tentar rodar uma classificação com múltiplas labels, apesar de Leite et al. [5] desencorajar a última opção. Uma vantagem dessa base de dados em relação a outras mais restritas é que ela permite explorar diferentes problemas para além do inicialmente proposto, todos com seu cenário de interesse.

REFERÊNCIAS

[1] Artstein, R., and Poesio, M. Inter-coder agreement for computational linguistics. *Computational Linguistics 34*, 4 (2008), 555–596.

[2] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[3] Fortuna, P., da Silva, J. R., Wanner, L., Nunes, S., et al. A hierarchically-labeled portuguese hate speech dataset. In *Proceedings of the Third Workshop on Abusive Language Online* (2019), 94–104.

[4] Gate. Twitter collector. https://cloud.gate.ac.uk/shopfront/displayItem/twitter-collector. (Accessed on 01/20/2021).

[5] Leite, J. A., Silva, D. F., Bontcheva, K., and Scarton, C. Toxic language detection in social media for brazilian portuguese: New dataset and multilingual analysis. *arXiv preprint arXiv:2010.04543* (2020).

[6] McCormick, C., and Ryan, N. Bert fine-tuning tutorial with pytorch, 2019.

[7] Pires, T., Schlinger, E., and Garrette, D. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502* (2019).

[8] Souza, F., Nogueira, R., and Lotufo, R. Bertimbau: Pretrained bert models for brazilian portuguese. In *Brazilian Conference on Intelligent Systems*, Springer (2020), 403–417.

[9] Trindade, L. V. P. Mídias sociais e a naturalização de discursos racistas no brasil. *COMUNIDADES, ALGORITMOS E ATIVISMOS DIGITAIS* (2020), 26.

[10] Wulczyn, E., Thain, N., and Dixon, L. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web* (2017), 1391–1399.