CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

O DISCURSO DE ÓDIO HOMOFÓBICO NA REDE SOCIAL X: UMA PROPOSTA DE MÉTODO DE CLASSIFICAÇÃO DE TEXTOS

Vinícius Soares dos Santos

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ, como parte dos requisitos necessários à obtenção do grau de mestre.

Orientadores:
Professor D.Sc. Gustavo Paiva Guedes e Silva

Rio de Janeiro,

Março de 2025

O discurso de ódio homofóbico na rede social X: uma proposta de método de classificação de textos

Qualificação de Mestrado em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ.

Vinícius Soares dos Santos

Aprovada por:

Presidente, Professor D.Sc. Gustavo Paiva Guedes e Silva (orientador)

Professor D.Sc. Eduardo Ogasawara

Professor D.Sc. Ronaldo Ribeiro Goldschmidt

Rio de Janeiro,

Março de 2025

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

S237 Santos, Vinícius Soares dos
O discurso de ódio homofóbico na rede social X: uma proposta de método de classificação de textos / Vinícius Soares dos Santos. — 2025.
95f. + apêndice : il. color. , enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2025.
Bibliografia : f. 88-95
Orientador: Gustavo Paiva Guedes e Silva

1. Discurso de ódio na Internet. 2. Homofobia. 3. Aprendizado do computador. 4. Lexicografia. I. Silva, Gustavo Paiva Guedes e. (Orient.). II. Título.

CDD 342.0853

Elaborada pela bibliotecária Tania Mello – CRB/7 nº 5507/04

DEDICATÓRIA

Dedico esta dissertação de mestrado aos meus amados e estimados pais, Egnaldo José dos Santos e Kátia Lima Soares. A vocês, que sempre acreditaram em mim, pelo direcionamento a uma Educação de qualidade e libertadora, e por todo apoio inabalável ao longo da minha jornada. Sou eternamente grato pelo suporte emocional e amor incondicional que me sustentaram e inspiraram em cada etapa da minha vida. Esta conquista é, também, fruto do esforço e da dedicação de vocês.

"Sei que dois e dois são quatro,
sei que a vida vale a pena,
mesmo que o pão seja caro
e a liberdade pequena."
Ferreira Gullar

AGRADECIMENTOS

Agradeço ao meu orientador, Prof. D.Sc. Gustavo Paiva Guedes e Silva, por todo apoio durante minha trajetória no Mestrado Acadêmico e pela orientação cuidadosa na produção desta dissertação.

Estendo minha gratidão a todo o quadro docente do Programa de Pós-Graduação em Ciência da Computação (PPCIC) do CEFET/RJ, com destaque ao Prof. D.Sc. Eduardo Ogasawara, cuja apresentação na *NVIDIA Rio Info AI Conference* em 2019 me proporcionou o primeiro contato com o programa de pós-graduação, inspirando minha aplicação no ano seguinte.

Aos colegas discentes do PPCIC, sou grato pelo grupo que se formou durante a pandemia de Covid-19, pela rede de apoio criada e pela colaboração que enriqueceu o ensino e a aprendizagem em um período tão desafiador.

Ao meu parceiro de vida, Rodrigo Pinheiro Dantas, registro meu profundo agradecimento por todo o estímulo contínuo, pela presença constante e por criar um ambiente de paz que me permitiu seguir em frente, alcançando meus objetivos acadêmicos e profissionais. Sua parceria foi fundamental para esta conquista.

RESUMO

O discurso de ódio homofóbico na rede social X: uma proposta de método de classificação de textos

Vinícius Soares dos Santos

Orientadores:
Professor D.Sc. Gustavo Paiva Guedes e Silva

Resumo da Dissertação submetida ao Programa de Pós-graduação em Ciência da Computação do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ como parte dos requisitos necessários à obtenção do grau de mestre.

O discurso de ódio homofóbico trata-se da ação ou incitamento ao ódio, à discriminação ou à hostilização de um indivíduo, motivado por um preconceito contra pessoas de orientação sexual lésbicas, gays, bissexuais, transexuais e travestis (LGBT). A homofobia foi criminalizada por meio de ação no Supremo Tribunal Federal (STF) do Brasil. Este trabalho propõe um método para construção de uma classificação do discurso de ódio homofóbico, alinhado aos Objetivos de Desenvolvimento Sustentável (ODS) que enfatiza a promoção da inclusão social, econômica e política de todos, sem distinção de orientação sexual ou qualquer outra condição. Submetida ao Comitê de Ética em Pesquisa (CEP), subordinado ao Ministério da Saúde do Brasil, recebeu aprovação para sua realização sob o certificado 59293922.0.0000.5289. Esta pesquisa busca superar as limitações dos estudos anteriores sobre a detecção de discurso de ódio homofóbico em português, que utilizam léxicos e bases de dados rotuladas incipientes, aprimorando o desempenho dos modelos de classificação. Os objetivos específicos são a criação de um léxico homofóbico LEX-HOMOFENSA-PT e a criação de uma base de dados de textos extraídos de tweets rotulados por humanos TEXT-HOMOFENSA-PT. Ao final, avaliou-se a aplicação de modelos de classificação ao dataset TEXT-HOMOFENSA-PT, classificadores tradicionais, *Logistic Regression* (LR), *Suport Vector Machine* (SVM), *K-Nearest Neighbors* (KNN), *Decision Tree* (DT) e *Naive Bayes* (NB); classificadores ensemble, *Random Forest* (RF), *Adaptive Boosting* (AB) e *Gradient Boosting* (GB); redes neurais artificiais, *Convolutional Neural Network* (CNN) e *Multilayer Perceptron* (MLP); classificadores *Large Language Models* (LLM), *Bidirectional Encoder Representations from Transformers* (BERT), *Robustly Optimized BERT Pretraining Approach* (RoBERTa) e *Generative Pre-trained Transformer* (GPT). O modelo LR obteve o melhor desempenho, com uma Precisão de 85,37%, uma Revocação de 85,30% e F-measure de 85,30%.

Palavras-chave: discurso de ódio; homofobia; aprendizado de máquina; aprendizado profundo; léxico; mineração de texto; classificação de texto

Rio de Janeiro,

Março de 2025

ABSTRACT

O discurso de ódio homofóbico na rede social X: uma proposta de método de classificação de textos

Vinícius Soares dos Santos

Advisors:
Professor D.Sc. Gustavo Paiva Guedes e Silva

Abstract of dissertation submitted to Programa de Pós-graduação em Ciência da Computação - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ as partial fulfillment of the requirements for the degree of master.

Homophobic hate speech refers to the act of inciting hatred, discrimination, or hostility against an individual, motivated by prejudice against people of lesbian, gay, bisexual, transgender, and transvestite (LGBT) sexual orientation. Homophobia was criminalized through a ruling by the STF in Brazil. This study proposes a method for constructing a classification of homophobic hate speech, aligned with the Sustainable Development Goals (SDGs), which emphasize the promotion of social, economic, and political inclusion for all, regardless of sexual orientation or any other condition. Submitted to the Research Ethics Committee (CEP), under the Brazilian Ministry of Health, the study received approval under certificate number 59293922.0.0000.5289. This research aims to overcome the limitations of previous studies on the detection of homophobic hate speech in Portuguese, which rely on rudimentary lexicons and labeled datasets, improving the performance of classification models. The specific objectives are the creation of a homophobic lexicon, LEX-HOMOFENSA-PT, and the development of a labeled dataset of texts extracted from tweets, TEXT-HOMOFENSA-PT. At the end of the study, the application of classification models to the TEXT-HOMOFENSA-PT dataset was evaluated, including traditional classifiers: LR, SVM, KNN, DT, and NB; ensemble classifiers: RF, AB, and GB; artificial neural networks: CNN and MLP; and large language model (LLM) classifiers: BERT, RoBERTa, and GPT. The LR model achieved the best performance, with an accuracy of 85.37%, a recall of 85.30%, and an F-measure of 85.30%.

Key-words: hate speech; homophobia; machine learning; deep learning; lexicon; text mining; text classification

Rio de Janeiro,

Março de 2025

Sumário

I Introdução **15**

I.1 Exemplo motivador 15

I.2 Definição do Problema 17

I.3 Objetivos 18

I.4 Metodologia 18

I.5 Organização da dissertação 19

II Fundamentação Teórica **20**

II.1 Homofobia e o discurso homofóbico nas redes sociais 20

II.2 Processamento de Linguagem Natural 22

II.3 Aprendizado de Máquina 23

II.4 Pré-processamento de Texto e Cálculo de Similaridade 24

II.5 A Tarefa de Classificação de Textos e Modelos de Classificação 25

II.6 Modelos de Fatoração de Matrizes Word-Context 34

II.6.1 Representação Gráfica de Distância de Palavras 36

II.6.2 Redes Neurais 37

II.6.3 Redes Neurais Recorrentes (RNNs) 39

II.6.4 Redes Neurais Convolucionais 41

II.6.5 Validação cruzada: processo de avaliação do modelo 42

II.7 Medidas de avaliação de modelos de aprendizado de máquina 42

III Trabalhos relacionados **45**

III.1 Dataset para análise de discurso de ódio homofóbico 46

III.2 Abordagem lexical e Modelos de aprendizagem de máquina 47

III.3 Síntese dos Trabalhos Relacionados 50

IV Metodologia **56**

IV.1 Método de criação do Léxico homofóbico (MCLH) 57

IV.2 Método de criação dataset *tweets* (MCDT) 59

IV.3 Método de rotulagem textos homofóbicos (MRTH) 60

IV.4 Método de classificação de aprendizado de máquina (MCAM) 61

V Avaliação Experimental 63

V.1 MCLH 63

V.2 MCDT 64

V.3 MRTH 66

V.4 MCAM 69

V.4.1 Análise de Modelos 70

V.4.2 Resumo dos classificadores 81

VI Conclusões 83

VI.1 Trabalhos Futuros 86

Referências 88

Apêndice A - API Twitter V2 96

Apêndice B - Termo de consentimento livre e esclarecido 98

Apêndice C - Léxico homofóbico 99

Lista de Figuras

I.1 Homofobia no X - fonte: S.PAULO [2017] 16

II.1 Exemplo de uma árvore de decisão - Fonte Pedregosa et al. [2011] 27
II.2 Exemplo modelo KNN para diferentes pesos - Fonte Pedregosa et al. [2011] 28
II.3 Exemplo de um Support Vector Machine - Fonte Pedregosa et al. [2011] 29
II.4 Exemplo de uma florestas aleatórias - Fonte Pedregosa et al. [2011] 31
II.5 Representação de Redes Neurais - Fonte Aggarwal [2018] 38
II.6 Representação de Redes Neurais - Fonte Aggarwal [2018] 40
II.7 Processo de validação cruzada. Fonte Pedregosa et al. [2011] 43
II.8 Processo de validação cruzada. Fonte Pedregosa et al. [2011] 43
II.9 Matriz de confusão para problemas de classificação binária 44

IV.1 Metodologia para Construção de Classificadores de Conteúdo Homofóbico (MCCCH) 56
IV.2 Contribuição humana ao léxico homofóbico 58
IV.3 MCLH - Pergunta 1 - Exemplo de termos homofóbicos Hatebase (demais omitidos) 59
IV.4 MCLH - Léxico homofóbico - Pergunta 2 59
IV.5 Método de rotulagem de mensagens textuais homofóbicas - MRMTH 62

V.1 Boxplot Tamanho dos Textos em Palavras - TEXT-HOMOFENSA-PT 65
V.2 Termos mais frequentes - TEXT-HOMOFENSA-PT 66
V.3 Termos mais frequentes 66
V.4 Exemplo de mensagem para rotulação 68
V.5 Frequência dos rótulos - TEXT-HOMOFENSA-PT 68
V.6 Boxplot rótulos por tamanho das frases 69
V.7 Matriz de confusão - Classificador LR 71
V.8 Matriz de confusão - Classificador NB 71
V.9 Matriz de confusão - Classificador SVM 72
V.10 Matriz de confusão - Classificador KNN 73
V.11 Matriz de confusão - Classificador DT 74
V.12 Matriz de confusão - Classificador RF 74

V.13 Matriz de confusão - Classificador AB 75
V.14 Matriz de confusão - Classificador GB 76
V.15 Matriz de confusão - Classificador MLP TF-IDF 77
V.16 Matriz de confusão - Classificador CNN 78
V.17 Matriz de confusão - Classificador BERT 79
V.18 Matriz de confusão - Classificador RoBERTa 80
V.19 Matriz de confusão - Classificador GPT 81

VI.1 E-mail de confirmação da submissão API Twitter v2 - Pesquisador 96
VI.2 E-mail de aprovação da submissão API Twitter v2 - Pesquisador 97
VI.3 Área do desenvolvedor API Twitter v2 - Pesquisador 97
VI.4 Termo de consentimento livre e esclarecido 98

Lista de Tabelas

III.1 Critério de inclusão e exclusão - revisão sistemática 45

III.2 Resumo do levantamento bibliográfico 52

III.3 Continuação da Tabela Resumo do levantamento bibliográfico 53

III.4 Continuação da Tabela Resumo do levantamento bibliográfico 54

III.5 Continuação da Tabela Resumo do levantamento bibliográfico 55

IV.1 Termos homofóbicos em português - fonte: Hatebase [2020] 57

V.1 Análise Descritiva do Tamanho dos Textos em Palavras - TEXT-HOMOFENSA-PT 64

V.2 Tabela resumo dos resultados modelos de classificação 82

Lista de Abreviações

AB	*Adaptive Boosting*	49, 61, 69, 74, 75, 82
API	*Application Programming Interface*	17
BERT	*Bidirectional Encoder Representations from Transformers*	18, 61, 62, 69, 78, 79, 80, 82, 84, 85
BOW	Bag-of-Words	24, 31, 70
CEP	Comitê De Ética Em Pesquisa	19
CMC	Comunicação Mediada Pelo Computador	15
CNN	*Convolutional Neural Network*	18, 61, 62, 69, 76, 78, 82, 84, 85
CNS	Conselho Nacional De Saúde	19
CONEP	Comissão Nacional De Ética Em Pesquisa	19
DT	*Decision Tree*	18, 26, 47, 61, 69, 73, 74, 81, 82, 84
FN	Falsos Negativos	43
FP	Falsos Positivos	43
GB	*Gradient Boosting*	49, 61, 69, 75, 76, 82
GPT	*Generative Pre-trained Transformer*	18, 61, 62, 80, 81, 82, 84, 85
KNN	*K-Nearest Neighbors*	18, 27, 61, 69, 72, 73, 81, 82, 84
LGBT	Lésbicas, Gays, Bissexuais E Transexuais	16, 21, 48, 83
LLM	*Large Language Models*	18, 50, 61, 62, 69, 80, 82, 84, 85, 86
LR	*Logistic Regression*	47, 48, 70, 71, 72, 73, 81, 82, 84, 85
MCAM	Método De Classificação Por Aprendizado De Máquina	56, 61, 63, 69, 84
MCCCH	Metodologia Para Construção De Classificadores De Conteúdo Homofóbico	56, 84
MCDT	Método De Criação Do Dataset Tweets	56, 59, 63, 64, 84
MCLH	Método De Criação Do Léxico Homofóbico	56, 57, 63, 84, 85
MLP	*Multilayer Perceptron*	18, 47, 48, 61, 62, 69, 75, 77, 82, 84, 85
MRTH	Método De Rotulagem De Textos Homofóbicos	56, 60, 63, 65, 66, 84, 85, 86
NB	*Naive Bayes*	18, 46, 47, 48, 49, 71, 82
NLG	Natural Language Generation	22
NLU	Natural Language Understanding	22, 23
PLN	Processamento De Linguagem Natural	20, 22, 24, 51

RF	*Random Forest*	47, 48, 49, 61, 69, 73, 74, 82
ROBERTA	*Robustly Optimized BERT Pretraining Approach*	18, 61, 62, 69, 79, 80, 82, 84, 85
STF	Supremo Tribunal Federal	16, 21, 83
SVM	*Suport Vector Machine*	18, 28, 39, 46, 47, 48, 49, 51, 61, 69, 70, 72, 81, 82, 84
TCLE	Termo De Consentimento Livre E Esclarecido	57, 61
TF-IDF	Termo Frequência-Frequência Inversa Do Documento	25, 47, 48, 70
VN	Verdadeiros Negativos	43
VP	Verdadeiros Positivos	43

Capítulo I Introdução

O advento da *Internet* proporcionou diversas mudanças ao mundo contemporâneo, dentre elas a maneira como a sociedade se relaciona. Para Recuero [2012], ferramentas de Comunicação Mediada pelo Computador (CMC) permitiram que pessoas, instituições ou grupos de pessoas se expressassem e socializassem, por meio da interação e da comunicação entre si. Segundo a autora, as relações em uma rede social virtual são estabelecidas por meio da interação social, caracterizada pela diversidade de trocas de informações que podem abranger aspectos afetivos, pessoais, profissionais, entre outros.

A revista Exame [2022] divulgou uma estimativa apontando que as redes sociais no mundo já chegam a 4,62 bilhões de usuários. Segundo a matéria, esse número tende a chegar rapidamente a 60% da população global, uma vez que as plataformas recebem cerca de um milhão de novos usuários por dia. Dados do jornal de economia Valor Investe [2022] apontavam que, em janeiro do ano de 2022, o X, antigo *Twitter*, possuía mais de 217 milhões de usuários em sua plataforma. Ainda segundo o jornal, a quarta maior base de usuários do X no mundo é de brasileiros, com um total estimado de aproximadamente 19 milhões de usuários ativos no período em questão.

O *X* é definido como um serviço de microblogue, por meio do qual pessoas e empresas podem se comunicar por mensagens [Honey and Herring, 2009]. O significado do termo microblogue é definido pelo tipo de serviço que é oferecido, e que segundo a própria empresa consiste em: "*as pessoas publicam mensagens, que podem conter fotos, vídeos, links e texto. Essas mensagens são publicadas em seu perfil e enviadas a seus seguidores, podendo ser encontradas por meio de busca na rede*" [Twitter, 2020]. A característica que destaca o *X* das outras redes sociais é que as mensagens trocadas limitam-se a 280 caracteres, chamadas pelo microblogue de *tweets*, que permitem que os usuários compartilhem suas "*últimas histórias, ideias, opiniões e notícias em tempo real, escolhendo entre mensagens públicas ou um grupo selecionado*" [Twitter, 2020].

I.1 Exemplo motivador

Segundo Meyer-Pflug [2008], entende-se como discurso de ódio a manifestação de incitação a discriminação racial, sexual, social ou religiosa. Ainda segundo a autora, o discurso do ódio é mais do que uma manifestação de antipatia, ele é indicado como hostilidade, ameaças e agressões verbais. Silva et al. [2011] argumenta que o discurso de ódio é junção de dois elementos principais:

discriminação e externalidade. Para eles, a existência do discurso de ódio exige mudança de ideias do plano mental (abstrato) para o plano fático (concreto), justificando a externalidade.

Para Costa and Nardi [2015], o termo Homofobia é utilizado para conceituar o preconceito e discriminação contra pessoas cuja orientação sexual seja diferente da heterossexual. Os autores citam, no contexto histórico do século XX, a interpretação dominante da psicologia e da psiquiatria, a respeito da homossexualidade, corroborou o preconceito contra homossexuais. Ainda segundo eles, o termo "homossexualismo" foi considerado por muito tempo no manual de diagnóstico de doenças psiquiátricas. Por fim, os autores abordam a homofobia como um preconceito contra a diversidade sexual.

Ao passo que as redes se popularizam, o reflexo da sociedade em torno do discurso de ódio também é percebido. Segundo o portal *UOL Notícias*, um perfil homofóbico chamado "*HomofobiaSIM*" teve rápida ascensão na rede social *X*, contando com mais de 15 mil seguidores e forte incitação. O perfil, como seu próprio nome sugere, incitava violência contra homossexuais. A reportagem ainda destaca que o perfil chegava ao ponto de compartilhar que homossexuais mereciam morrer (*e.g.*, "*Um viadinho [sic] brasileiro foi espancado, aí que está o erro, ele deveria ter sido é morto, acabe com Aids, mate um gay*") [UOL, 2010].

Outra notícia que traz à tona casos de homofobia foi relatada pelo jornal *O Estado de São Paulo*. Um usuário do *X* com mais de 7 milhões de inscritos em suas redes sociais, contratado pelo Governo Federal para divulgação do programa de Reforma do Ensino Médio no ano de 2017, teve mensagens antigas denunciadas por diversos usuários da rede por conter discurso de ódio. Dentre essas mensagens, destaca-se uma de cunho homofóbico, divulgada na mesma matéria jornalística, conforme exibido na Figura I.1.



Figura I.1: Homofobia no X - fonte: S.PAULO [2017]

O STF do Brasil, em 13 de junho de 2019, julgou como crime de racismo casos de agressões contra pessoas de orientação sexual Lésbicas, Gays, Bissexuais e Transexuais (LGBT) [UOL, 2020]. Na ação denominada "*Homofobia e omissão legislativa*", lista-se o julgamento e as alegações da suprema corte. Dentre os trechos da determinação, destaca-se o enquadramento de atos homofóbicos e transfóbicos como formas contemporâneas de racismo [Brasil, 2019].

Os Objetivos de Desenvolvimento Sustentável (ODS) da Agenda 2030 da ONU destacam a importância de combater todas as formas de discriminação, incluindo o discurso de ódio e a homofobia,

como parte do compromisso global com a igualdade e a inclusão [Organização das Nações Unidas, 2015]. A ODS 10, em sua meta 10.2, enfatiza a promoção da inclusão social, econômica e política de todos, sem distinção de raça, gênero, orientação sexual ou qualquer outra condição. Além disso, a meta 10.3 reforça a necessidade de eliminar práticas discriminatórias, o que inclui comportamentos que promovam discursos de ódio.

I.2 Definição do Problema

Em 2017, o X definiu novas regras para combater o abuso, violência e discurso de ódio. Em constante atualização, a conduta de propagação de ódio define que não é permitido "*promover violência, atacar diretamente ou ameaçar outras pessoas com base em raça, etnia, nacionalidade, orientação sexual, sexo, identidade de gênero, religião, idade, deficiência ou doença grave*" [Twitter, 2020].

A aplicação dessas regras, entretanto, pode requerer um grande esforço manual para revisar cada mensagem compartilhada na rede. MacAvaney et al. [2019] destacam que as ferramentas de identificação e classificação automáticas de textos podem acelerar o processo de curadoria dos textos, e ainda que os recursos humanos poderiam ser utilizados em análises mais aprofundadas de temas específicos ainda não automatizados. Desse modo, o *X* poderia impedir que as mensagens de cunho homofóbico fossem compartilhadas ainda no momento da sua postagem, impedindo sua disseminação na rede.

Uma limitação à classificação automática de textos para este contexto de pesquisa é a incipiência de alguns léxicos para discurso de ódio homofóbico no idioma português. Cita-se como exemplo a ferramenta *Hatebase*, um vocabulário multilíngue, mantido por funcionários e voluntários em mais de 90 idiomas e mais de 200 países. Para construção do Hatebase [2020], foi desenvolvido o *Hatebrain*, uma engenharia de processamento de linguagem natural que realiza análises linguísticas em conversas públicas para estimar uma probabilidade de conteúdo com discurso de ódio. Todos os dados estão disponíveis para consulta por meio do seu sítio eletrônico ou sua interface de programação de aplicativos, em inglês, *Application Programming Interface* (API). Atualmente, nessa ferramenta, existem 28 termos relacionados à homofobia no idioma português.

Uma outra limitação ao uso de aprendizado de máquina supervisionado para classificação é a necessidade de um conjunto de dados rotulados. No trabalho de Fortuna et al. [2019b], uma base de dados para o discurso de ódio foi construída a partir da coleta no X para 19 termos de ódio e 10 *hashtags*. Duas pessoas julgaram as mensagens extraídas para os rótulos de "sim" e "não" quanto ao discurso de ódio. Os respondentes, quando julgavam que o conteúdo era discurso de ódio, informavam, na sequência, que tipo de discurso de ódio era, para uma lista contendo "homofobia", "racismo", "religião" e outros. No total, 5.668 *tweets* foram rotulados, porém apenas 365 destes, ou

aproximadamente 6,4% da base de dados, foram para o discurso homofóbico. A amostra de 365 *tweets* ainda é limitada e uma expansão desse conjunto de dados é necessária.

I.3 Objetivos

Dada a problemática relatada na Seção anterior, o objetivo geral da pesquisa é desenvolver um método de identificação do discurso homofóbico em mensagens do *X*. O estudo concentra-se na análise de conteúdos textuais, abordando conceitos e trabalhos relacionados a métodos de classificação automática do discurso de ódio, com ênfase no discurso homofóbico. Diferentemente dos estudos existentes, este trabalho propõe expandir o léxico de termos homofóbicos, atualmente limitado a apenas 28 termos no Hatebase; criar uma base de dados rotulada específica para discurso de ódio homofóbico, suprindo a limitação encontrada no trabalho de de Pelle and Moreira [2017] onde apenas 365 textos foram rotulados para homofobia; e avaliar algoritmos de classificação estado-da-arte para a tarefa de classificação de textos. Para alcançar o objetivo principal, têm-se como objetivos específicos:

- Criação de um léxico de termos homofóbicos.
- Criação de uma base de dados de textos, produzidos por meio da extração da informação dos *tweets* da rede social em questão.
- Rotulagem dos textos a partir do julgamento humano para as classes “sim”, “não” e “não tem certeza”.
- Aplicação de modelos de classificação ao conjunto de tweets rotulados, para: (i) classificadores tradicionais, como SVM, KNN, DT e NB; (ii) classificadores ensemble, como Random Forest, Adaptive Boosting e Gradient Boosting; (iii) redes neurais artificiais, como CNN e MLP; (iv) LLM, como BERT, RoBERTa e GPT.

Para embasar a pesquisa e garantir uma análise abrangente da literatura existente, foi realizada uma revisão sistemática nas plataformas *SCOPUS*, *IEEE XPLORER* e *ACL Anthology* a partir da *string* de busca *(classification OR detection) AND (homophobia OR homophobic)*, resultando em 203 documentos localizados nos acervos das bases científicas. Em seguida, foram definidos critérios de inclusão e exclusão, finalizando com o total de 112 documentos. Técnicas de *snowbowlling* foram consideradas para a análise das referências e citações dos documentos selecionados.

I.4 Metodologia

A presente pesquisa foi submetida à apreciação do *X*, a fim de adquirir os direitos de uso de sua *API Developer V2*, lançada em janeiro de 2021, permitindo a coleta de até 10 milhões de

tweets mensalmente. Segundo o Twitter [2020], além do acesso gratuito a pesquisadores, sua *API* permite a busca por dados históricos, antes restritos a usuários pagantes. A empresa disponibiliza guias técnicos e metodológicos para facilitar a construção das consultas e análises dos conteúdos extraídos.

No dia 22 de setembro de 2021, a empresa aprovou a submissão e concedeu as credenciais de uso. Destaca-se ainda que a justificativa para a escolha dos dados provenientes do *X* foi devido aos exemplos de homofobia relatados na Seção I.1, o acesso gratuito à plataforma e os recursos disponibilizados pela rede social aos pesquisadores para elaboração da pesquisa.

Ressalta-se que a pesquisa envolve seres humanos na criação do léxico homofóbico e na rotulagem dos textos potencialmente homofóbicos. Devido a isso, foi submetido o pedido ao Comitê de Ética em Pesquisa (CEP), coordenado pela Comissão Nacional de Ética em Pesquisa (CONEP), subordinada ao Conselho Nacional de Saúde (CNS) do Ministério da Saúde do Brasil, recebendo a aprovação sob o número de protocolo 59293922.0.0000.5289.

I.5 Organização da dissertação

As próximas seções estão estruturadas da seguinte forma: no Capítulo 2, são apresentadas as fundamentações teóricas; no Capítulo 3, estão elencados os trabalhos relacionados; no Capítulo 4, apresenta-se a metodologia; no Capítulo 5, a avaliação experimental e os resultados; no Capítulo 6, as conclusões finais.

Capítulo II Fundamentação Teórica

O presente capítulo tem como objetivo estabelecer uma base conceitual sólida para a compreensão dos principais temas abordados nesta pesquisa. Inicialmente, realiza-se uma contextualização acerca da homofobia e do discurso homofóbico nas redes sociais, situando o fenômeno no âmbito das interações digitais. Em seguida, apresenta-se o embasamento teórico do Processamento de Linguagem Natural (PLN), com ênfase no léxico e vocabulário. Por fim, são explorados os conceitos da tarefa de classificação de textos, incluindo abordagens do aprendizado de máquina supervisionado, modelos de classificação e métricas de avaliação empregadas na análise e validação cruzada no processo de avaliação dos modelos. Essa estrutura visa fornecer uma compreensão abrangente dos elementos teóricos que sustentam o método proposto.

II.1 Homofobia e o discurso homofóbico nas redes sociais

O termo "Homofobia" é utilizado para conceituar o preconceito e discriminação contra pessoas cuja orientação sexual seja diferente da heterossexual, segundo Costa and Nardi [2015]. Os autores contextualizam que a palavra "homofobia" é um neologismo que foi cunhado pelo psicólogo George Weinberg em 1972, agrupando dois radicais, "semelhante" e "medo", com o objetivo de definir sentimentos negativos em relação a homossexuais, tais como aversão, desprezo, ódio, desconfiança, desconforto ou medo. Os autores citam ainda que, no contexto histórico do século XX, a interpretação dominante da psicologia e da psiquiatria a respeito da homossexualidade enquanto doença, distúrbio ou perversão, corroborou para o preconceito contra homossexuais.

Segundo Meyer-Pflug [2008], entende-se por discurso de ódio a manifestação de incitação à discriminação racial, sexual, social ou religiosa. A autora define o discurso de ódio como mais do que uma manifestação de antipatia; ele é indicado como hostilidade, ameaças e agressões verbais. Silva et al. [2011] argumentam que o discurso de ódio é a junção de dois elementos principais: discriminação e externalidade. Para eles, a existência do discurso de ódio exige mudança de ideias do plano mental (abstrato) para o plano fático (concreto), justificando a externalidade.

Os autores Quadrado and Ferreira [2020] constatam em sua pesquisa que as redes sociais produzem uma espécie de validação do seu ódio que, antes da sua existência, era de difícil propagação. Eles indicam que os casos de crimes cibernéticos de ódio possuem como características o ataque,

principalmente, às minorias na forma de misoginia, xenofobia, racismo, homofobia e outros. Ainda segundo os autores, nas redes sociais, esse tipo de discurso se baseia em estereótipos e estigmas sociais, configurando-se como uma competição na qual a intensidade do conteúdo odioso parece elevar o prestígio e a aceitação do emissor com quem compartilha o mesmo pensamento.

Os autores Silva et al. [2019] analisaram a ação das redes sociais Facebook, Twitter e YouTube, entre os anos de 2015 e 2018 em relação ao discurso de ódio. Eles apontam que, a partir de 2015, as redes sociais tratam da problemática do ódio online pautando suas políticas nos princípios fundamentais da liberdade de expressão, dignidade humana, segurança pessoal e respeito pelo estado democrático de direito. Os autores verificaram que, entre os anos de 2015 e 2016, as redes sociais investiram seus principais esforços para a política das plataformas e educação dos usuários quanto à tolerância, civilidade e respeito. Em relação aos anos de 2017 e 2018, os autores pontuam que as redes sociais abordaram questões tecnológicas no combate ao discurso de ódio, com o desenvolvimento e aperfeiçoamento de recursos de Inteligência Artificial e Aprendizagem de Máquina.

Em permanente modernização, em abril de 2023 a rede social X (antigo Twitter) atualizou sua política de uso da plataforma, no que tange à propagação do discurso de ódio e desumanização. Disponível para consulta em sua página da internet[1], dentre outros pontos que a norteiam, destaca-se a passagem: "*Proibimos a desumanização de um grupo de pessoas, com base na religião, casta, idade, deficiência, doença grave, origem nacional, raça, etnia, sexo, identidade de gênero ou orientação sexual.*" [Twitter, 2020]. A penalidade mais grave imposta pela rede social a quem violar os termos da política de propagação de ódio é a suspensão das contas.

Entretanto, avaliar o discurso de ódio em tempo real é uma tarefa árdua e, por isso, conforme supracitado por Silva et al. [2019], a partir de 2017, ações tecnológicas são desenvolvidas e aperfeiçoadas. Os pesquisadores MacAvaney et al. [2019] destacaram que as ferramentas de identificação e classificação automáticas de textos podem acelerar o processo de curadoria dos textos, e ainda que os recursos humanos poderiam ser utilizados em análises mais aprofundadas de temas específicos ainda não automatizados. Desse modo, o *X* poderia impedir que as mensagens de cunho homofóbico fossem compartilhadas ainda no momento da sua postagem, impedindo sua disseminação na rede.

No Brasil, em 13 de junho de 2019, o discurso de ódio homofóbico é considerado crime. O STF do Brasil enquadrou como crime de racismo casos de agressões contra pessoas de orientação sexual LGBT [UOL, 2020]. Na ação denominada "*Homofobia e omissão legislativa*", lista-se o julgamento e as alegações da suprema corte para embasar a decisão, uma vez que o Estado Brasileiro não dispunha de lei específica para o caso. Dentre os trechos da determinação, destaca-se o enquadramento de atos homofóbicos e transfóbicos como formas contemporâneas de racismo [Brasil, 2019].

[1] `https://help.x.com/pt/rules-and-policies/hateful-conduct-policy`

II.2 Processamento de Linguagem Natural

De acordo com Khurana et al. [2017], uma linguagem pode ser definida como um conjunto de regras ou um conjunto de símbolos, nos quais os símbolos são combinados e utilizados para transmitir ou difundir informações. Com sua origem a partir dos anos de 1940, os autores definem que o PLN é a combinação da inteligência artificial e a linguística, classificadas em duas partes: a Compreensão de Linguagem Natural (Natural Language Understanding - NLU) e a Geração de Linguagem Natural (Natural Language Generation - NLG).

A Natural Language Understanding (NLU) capacita as máquinas a interpretar e processar a linguagem humana, identificando conceitos, entidades, emoções e palavras-chave. Para alcançar essa compreensão, segundo Khurana et al. [2017], a linguística torna-se fundamental, enquanto ciência que estuda o significado, o contexto e as diversas formas da linguagem. Diferentes níveis linguísticos são considerados no processo de NLU citadas pelo autor:

- Fonologia: estudo da organização dos sons na linguagem.
- Morfologia: análise da estrutura e formação das palavras.
- Léxico: compreensão e interpretação de palavras individuais.
- Sintaxe: análise da estrutura gramatical das frases.
- Semântica: interpretação do significado das frases.
- Discurso: análise de textos mais extensos, considerando a relação entre frases.
- Pragmática: interpretação baseada no contexto e conhecimento de mundo.

Por outro lado, ainda segundo os autores, a Natural Language Generation (NLG) refere-se ao processo de criação de textos compreensíveis a partir de representações internas, sendo o oposto da Compreensão de Linguagem Natural (NLU). Esse processo envolve quatro fases principais:

- Identificação de Objetivos: Determina o conteúdo a ser comunicado.
- Planejamento de Conteúdo: Organiza as informações de acordo com a estrutura linguística adequada.
- Seleção de Recursos Linguísticos: Escolha de palavras, expressões e estruturas gramaticais mais apropriadas.
- Realização Textual: Transformação das informações selecionadas em frases ou parágrafos completos.

No contexto da NLU, segundo Vilela [1997], o léxico é descrito como uma codificação da realidade que reflete o conhecimento compartilhado por uma comunidade linguística, podendo ser visto tanto de uma perspectiva cognitiva quanto comunicativa. O vocabulário é uma subdivisão do léxico, representando as palavras que realmente são usadas em um contexto específico, seja de um autor, um texto ou uma área de conhecimento. A diferença entre léxico e vocabulário é apresentada pelo autor como uma distinção entre "parte" e "todo"; o léxico compreende todas as palavras existentes de uma língua, enquanto o vocabulário se refere ao conjunto de termos efetivamente utilizados em determinado tempo e lugar.

II.3 Aprendizado de Máquina

O Aprendizado de máquina é uma área da Inteligência Artificial, cujo objetivo é desenvolver técnicas computacionais e a construção de sistemas capazes de adquirir conhecimento de forma automática Monard and Baranauskas [2003]. Os diversos sistemas de aprendizado de máquina possuem características particulares e comuns que possibilitam a classificação quanto à linguagem, modo ou paradigma de aprendizado adotado.

Segundo Ricci et al. [2010], modelos de aprendizagem de máquina supervisionada são construídos a partir de uma entrada de dados rotulados, assim sendo possível avaliar a saída predita com os dados reais. Ainda segundo o autor, os modelos supervisionados são divididos em regressão e classificação, no qual a regressão busca encontrar como uma variável evolui em relação a outras e a classificação busca explicar uma variável categórica.

As variáveis categóricas dos modelos de aprendizado de máquina de classificação são, segundo o autor:

- binárias: apenas duas categorias, representadas por "ocorrência de um evento = 1" ou "não ocorrência de um evento = 0";
- faixas: como níveis de renda, faixas de idade, escalas e outros;
- variáveis puramente categóricas não ordenadas:o gênero, raça, sexo, cor, religião e outros.

De acordo com Aggarwal [2018], a extração de informações úteis a partir de textos por meio de algoritmos estatísticos é conhecida como mineração de texto, análise de texto ou aprendizado de máquina aplicado a textos. Em sua obra, o autor destaca o aumento da relevância das análises textuais com o advento da Internet. Segundo ele, as mídias sociais se destacam como uma fonte rica de textos devido à sua natureza aberta, permitindo que qualquer pessoa compartilhe conteúdo. O que torna essas postagens únicas é o uso frequente de acrônimos curtos e informais, o que demanda técnicas específicas de análise para interpretar e extrair informações relevantes.

O autor Aggarwal [2018] sinaliza a existência de duas principais formas de representação de texto usadas nesse contexto:

- Texto como Bag-of-Words (BoW): Nessa abordagem, a ordem das palavras é ignorada, e o documento é representado por uma matriz multidimensional esparsa baseada na frequência dos termos. Essa representação é amplamente usada em tarefas como classificação, modelagem de tópicos e sistemas de recomendação.

- Texto como conjunto de sequências: as palavras são mantidas na ordem original, geralmente dentro dos limites de sentenças ou parágrafos. Essa abordagem permite uma interpretação semântica mais profunda, sendo utilizada em tarefas de PLN.

A maioria dos modelos de Aprendizado de Máquina para Textos utiliza a representação BoW, em que palavras são tratadas como dimensões e seus valores correspondem às frequências nas quais aparecem.

II.4 Pré-processamento de Texto e Cálculo de Similaridade

Segundo Aggarwal [2018], o pré-processamento de texto é essencial para transformar dados não estruturados em uma representação estruturada e multidimensional. Dentre as etapas do pré-processamento, o autor destaca:

- Extração de Texto: separação de conteúdo relevante de anúncios ou seções irrelevantes em páginas da web.

- Remoção de Stop-Words: eliminação de palavras comuns como artigos e preposições para reduzir ruído nos dados.

- Stemming e Normalização: consolidação de palavras com a mesma raiz, como “sinking” e “sank”, em um único termo básico. Normalizar o texto para todas as letras minúsculas (capitalização) e a remoção de pontuação conforme o contexto.

- Normalização de Frequência: aplicação da técnica TF-IDF, que pondera palavras com base na sua frequência relativa nos documentos.

Os autores Kusner and Kolkin [2015] definem a frequência do termo no documento d como a frequência que representa o número de instâncias de um determinado termo t. Enquanto isso, para eles, a frequência do documento f_d avalia o significado do texto para toda a coleção do corpus. No documento d, TF é o contador de frequência para um termo t, enquanto f_d é o número de ocorrências no conjunto de documentos D do termo t.

$$TF(t,d) = \frac{f_d(t)}{\max\limits_{t \in d} f_d(t)} \tag{II.1}$$

Para o autor, a frequência inversa do documento IDF avalia o quão relevante é o termo, de modo que o mais comum é considerado menos significativo.

$$IDF(t) = \ln\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right) \tag{II.2}$$

Assim, Termo Frequência-Frequência Inversa do Documento (TF-IDF) é definido por

$$TF\text{-}IDF = TF(t) \times IDF(t) \tag{II.3}$$

De acordo com Aggarwal [2018], o cálculo de similaridade entre textos é bastante sensível ao tamanho dos documentos. A distância Euclidiana, apesar de amplamente utilizada em representações multidimensionais, pode gerar resultados distorcidos se não houver normalização prévia. Para contornar esse problema, é comum a aplicação da similaridade cosseno, que normaliza os documentos antes de calcular o produto escalar entre os vetores.

Ainda segundo Aggarwal [2018], o coeficiente de Jaccard (J) é outra medida de similaridade, especialmente utilizada em representações booleanas de texto. Ele é definido pela razão entre os termos comuns a dois conjuntos (A e B) e os termos distintos na união desses conjuntos, apresentando valores que variam entre 0 e 1, assim como a similaridade cosseno.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{II.4}$$

Embora a distância de Jaccard seja mais adequada para textos representados em forma booleana, no caso de TF-IDF, a similaridade cosseno é geralmente preferida, pois tende a produzir resultados semelhantes com maior eficiência.

II.5 A Tarefa de Classificação de Textos e Modelos de Classificação

Os autores Manning et al. [2008] descreveram em seu livro que a tarefa de Classificação de Textos busca determinar a qual classe, ou classes, um determinado texto pertence. Segundo eles, dada uma descrição $d \in X$ de um documento, onde X é o espaço de documentos e um conjunto fixo de classes $C = \{c_1, c_2, \ldots, c_t\}$, também chamadas de categorias ou rótulos. Os autores ilustram que, dado um conjunto de dados de treinamento D de documentos rotulados (d, c), onde $(d, c) \in X \times C$. Como exemplo aplicado ao contexto desta pesquisa, pode-se citar:

$$\langle d, c\rangle = (\text{Todo gay merece morrer}, \text{Homofobia})$$

para o documento de uma frase "Todo gay merece morrer" e a classe "Homofobia".

Usando um método de aprendizado ou algoritmo de aprendizado, deseja-se então definir um classificador ou função de classificação γ que mapeia documentos para classes:

$$\gamma : X \to C$$

Esse tipo de aprendizado é chamado de aprendizado supervisionado, denotado por T, sendo $T(D) = \gamma$. O método de aprendizado T toma o conjunto de dados de treinamento D como entrada e retorna a função de classificação aprendida γ.

Classificadores Rule-Based, ou baseados em regras, associam condições em subconjuntos de atributos a rótulos de classe. O antecedente de uma regra define a presença de palavras em um documento, e o consequente corresponde ao rótulo da classe. A previsão é feita combinando as regras cujos antecedentes correspondem à instância de teste.

O classificador *Naïve Bayes* é uma versão supervisionada dos modelos de mistura usados em *clustering*. Ele assume que os dados são gerados por uma combinação de k componentes, onde cada classe possui uma distribuição específica. Os parâmetros dessas distribuições são estimados para maximizar a probabilidade das instâncias de treino. Com base nessas probabilidades, o modelo prevê a classe mais provável de uma nova instância. O termo "naïve" refere-se à suposição simplificadora de independência entre os atributos.

As árvores de decisão particionam os dados hierarquicamente com base em condições sobre atributos, alocando documentos pertencentes a cada classe em um único nó. Divisões podem ser *univariadas*, utilizando um único atributo, ou multivariadas, com múltiplos atributos. O processo é repetido até que os nós contenham instâncias de uma única classe (nós-folha). A classe majoritária no nó folha é usada para prever rótulos de novos dados [Aggarwal, 2018].

As DT, ou árvores de decisão, são um método de aprendizado supervisionado não paramétrico usado para classificação e regressão [Pedregosa et al., 2011]. Para os autores, o objetivo é criar um modelo que preveja o valor de uma variável de destino aprendendo regras de decisão simples inferidas a partir dos recursos dos dados.

Segundo Pedregosa et al. [2011], uma árvore pode ser vista como uma aproximação constante por partes, segundo a Figura II.4, por exemplo, as árvores de decisão aprendem com os dados para um conjunto de regras de decisão, chamada *if-then-else*. Quanto mais profunda a árvore, mais complexas são as regras de decisão e mais adequado é o modelo.

Segundo Aggarwal [2018], os classificadores k-nearest neighbor (KNN), também conhecidos como



Figura II.1: Exemplo de uma árvore de decisão - Fonte Pedregosa et al. [2011]

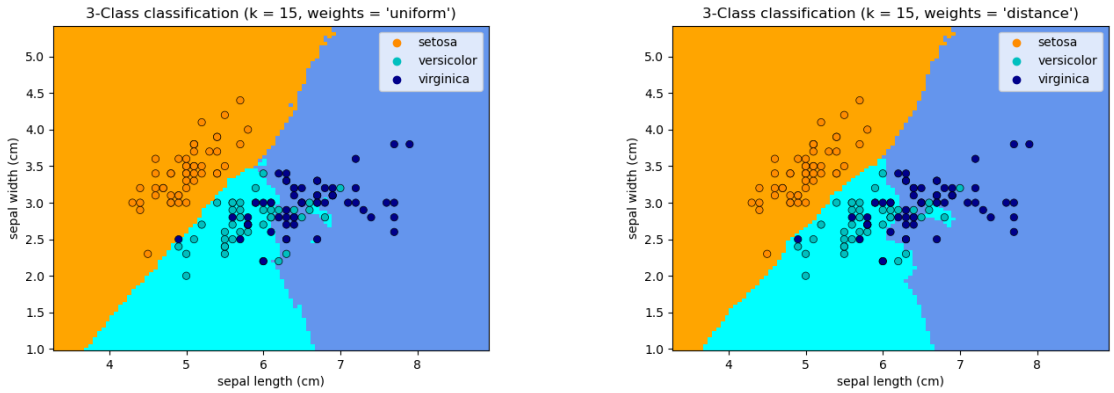
instance-based learners ou lazy learners, funcionam armazenando as instâncias de treinamento e classificando novos dados com base nos k vizinhos mais próximos. Diferente de métodos que realizam treinamento antecipado (eager learners), o KNN adia todo o processamento para o momento da classificação.

Embora o KNN tenha comportamento teoricamente ótimo com quantidades infinitas de dados, na prática, ele é superado por métodos que realizam treinamento prévio em conjuntos de dados finitos. Ainda assim, sua relevância teórica persiste, pois algoritmos de alto desempenho, como random forests e support vector machines (SVMs), podem ser vistos como variantes otimizadas do KNN [Aggarwal, 2018].

Segundo Pedregosa et al. [2011], a classificação baseada em KNN, ou seja, k-vizinhos mais próximos, é um aprendizado baseado em instâncias. A classificação é calculada a partir de uma votação majoritária simples; a um determinado ponto, é atribuída à classe de dados que tem mais representantes dentro dos k-vizinhos mais próximos do ponto.

Ainda segundo os autores, no contexto de métodos de classificação baseados em vizinhança, a escolha da estratégia de ponderação dos vizinhos pode influenciar significativamente o resultado, como observado na Figura II.2. A Figura ilustra o modelo de classificação KNN para $k = 15$ considerando pesos uniformes e de distância para 3 classes (setosa, versicolor e virgínica) do conjunto de dados flor de íris do autor Fisher [1936].

Para Pedregosa et al. [2011], quando o peso "uniforme" é utilizado, todos os pontos dentro da



(a) KNN - Peso Uniforme

(b) KNN - Peso Distância

Figura II.2: Exemplo modelo KNN para diferentes pesos - Fonte Pedregosa et al. [2011]

vizinhança recebem o mesmo peso na decisão, sem considerar a distância até o ponto de consulta. Já ao optar por "distância", o peso de cada vizinho é inversamente proporcional à sua distância em relação ao ponto consultado, fazendo com que os vizinhos mais próximos tenham maior influência na classificação do que aqueles mais distantes. Esse método pode ser especialmente útil em cenários onde a proximidade dos exemplos é um fator determinante para a tomada de decisão.

As Support Vector Machines (SVMs) possuem uma interpretação geométrica, baseada na separação por margem dos pontos em classes. O conceito central é criar dois hiperplanos paralelos simétricos ao redor da fronteira de decisão, garantindo que a maioria dos pontos fique do lado correto. Embora muitas abordagens apresentem o SVM por meio de uma visão geométrica, a interpretação baseada em otimização regularizada é considerada mais útil, pois conecta o SVM a outros modelos lineares, como a classificação por mínimos quadrados [Aggarwal, 2018].

Segundo Pedregosa et al. [2011], as classes SVC, NuSVC e LinearSVC são utilizadas para realizar classificação binária e multiclasse, e embora sejam métodos similares, elas diferem em parâmetros e formulações matemáticas. SVC e NuSVC suportam diferentes tipos de kernels, como linear, RBF e polinomial, oferecendo flexibilidade na separação dos dados. LinearSVC é uma implementação mais rápida voltada para kernels lineares, mas não suporta todos os atributos de SVC e NuSVC. A escolha entre esses métodos depende da aplicação e dos requisitos de performance, com LinearSVC sendo mais eficiente para grandes volumes de dados lineares. A Figura II.3 ilustra as abordagens do SVC, NuSVC e LinearSVC.

De acordo com Aggarwal [2018], a Regressão Logística pertence à família dos modelos lineares generalizados, com uma interpretação probabilística natural. Ela é projetada para lidar com variáveis dependentes binárias, mas pode ser estendida para variáveis ordinais, categóricas e de contagem. A regressão logística assume que a variável-alvo $y_i \in \{-1, +1\}$ é gerada a partir de uma distribuição de Bernoulli, parametrizada por uma função sigmoidal de $W \cdot X_i$. A função de predição

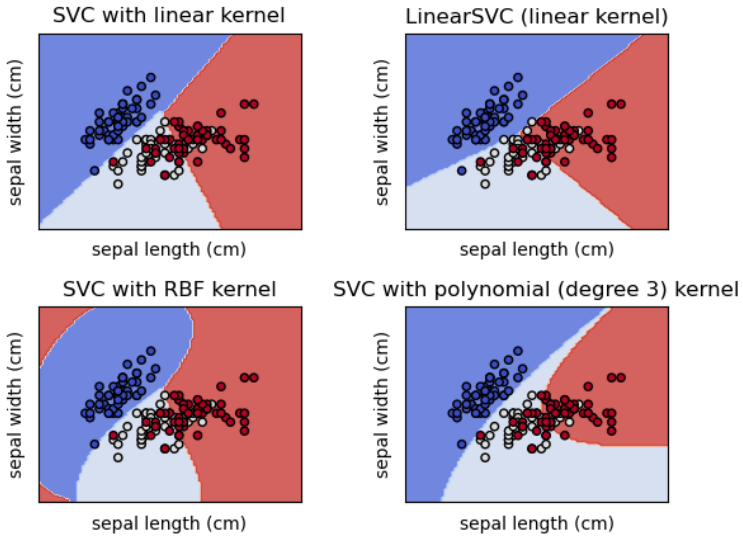


Figura II.3: Exemplo de um Support Vector Machine - Fonte Pedregosa et al. [2011]

é dada por:

$$P(F(Z) = 1) = \frac{1}{1 + \exp(-W \cdot Z)} \tag{II.5}$$

De forma mais geral, para qualquer valor $y \in \{-1, +1\}$, a probabilidade é:

$$P(F(Z) = y) = \frac{1}{1 + \exp(-y(W \cdot Z))}. \tag{II.6}$$

O objetivo da regressão logística é maximizar a verossimilhança ($\mathcal{L}$) dos dados, que para um conjunto de treino com n pares (X_i, y_i) é dada por:

$$\mathcal{L}(\text{Training Data}|W) = \prod_{i=1}^{n} P(F(X_i) = y_i) = \prod_{i=1}^{n} \frac{1}{1 + \exp(-y_i(W \cdot X_i))}. \tag{II.7}$$

Para resolver este problema, minimiza-se o log-verossimilhança negativa ($\mathcal{LC}$), que é expresso como:

$$\mathcal{LC} = \sum_{i=1}^{n} \log[1 + \exp(-y_i(W \cdot X_i))]. \tag{II.8}$$

Após a adição do termo de regularização, essa função torna-se equivalente à função objetivo da regressão logística, mostrando que o método é essencialmente um algoritmo de minimização da log-verossimilhança negativa.

De acordo com Aggarwal [2018], métodos de ensembles são meta-algoritmos que utilizam um modelo base, aplicando-o repetidamente em diferentes versões dos dados ou em variantes do mesmo

modelo. Os resultados obtidos são então combinados para gerar uma previsão mais precisa. *Bagging* e *subsampling* são métodos utilizados para reduzir a variância em algoritmos ensemble; ambos envolvem a criação de múltiplos modelos com base em amostras diferentes dos dados de treino e combinam suas predições para gerar um resultado mais robusto. Entretanto, uma desvantagem apontada é que as amostras geradas podem conter correlações (*bagging*) ou perda de padrões importantes (*subsampling*).

O *boosting* é uma técnica de ensemble em que pesos são atribuídos a cada instância de treino e ajustados iterativamente com base no desempenho dos classificadores. A ideia central é dar maior peso às instâncias mal classificadas, buscando corrigir os erros em iterações futuras e reduzir o viés do modelo. Os classificadores subsequentes dependem dos resultados dos modelos anteriores, criando uma combinação ponderada que melhora a precisão geral. O algoritmo mais conhecido de *boosting* é o AdaBoost, que funciona associando pesos iguais a todas as instâncias no início, ajustando-os a cada rodada de treinamento, onde instâncias mal classificadas têm seus pesos aumentados e instâncias corretamente classificadas têm seus pesos reduzidos [Aggarwal, 2018].

O *boosting* é mais sensível a dados ruidosos, pois assume que erros de classificação resultam do viés perto da fronteira de decisão, porém podem ser causados por erros de rotulação no conjunto de dados. Esse ruído é intrínseco aos dados, não ao modelo. Nessas situações, o *boosting* tende a superajustar o classificador a partes de baixa qualidade dos dados, prejudicando o desempenho. No entanto, quando os dados não contêm excesso de ruído, o *boosting* geralmente supera o *bagging* em termos de precisão [Aggarwal, 2018].

Os métodos ensemble de Random Forest, ou florestas aleatórias, são técnicas de perturbação e combinação especificamente projetadas para árvores de decisão. Segundo Pedregosa et al. [2011], um conjunto diversificado de classificadores é criado pela introdução de aleatoriedade na construção do classificador.

Ainda segundo os autores, em florestas aleatórias, cada árvore no ensemble é construída a partir de uma amostra com substituição para o conjunto de treinamento. Ao dividir cada nó durante a construção de uma árvore, a melhor divisão é encontrada em todos os recursos de um subconjunto aleatório, conforme ilustrado na Figura II.4. As árvores de decisão individuais normalmente exibem alta variância e tendem a se ajustar demais, enquanto a aleatoriedade injetada nas florestas produz árvores de decisão com a redução da variância, resultando em um modelo que melhor se adapta aos dados.

Segundo Aggarwal [2018], a necessidade de utilizar representações semânticas de alta dimensão surge do fato de que métodos tradicionais, como a representação "bag-of-words", não capturam as complexidades linguísticas e a estrutura sequencial dos textos. O autor enumera duas limitações para justificar o uso de representações semânticas:

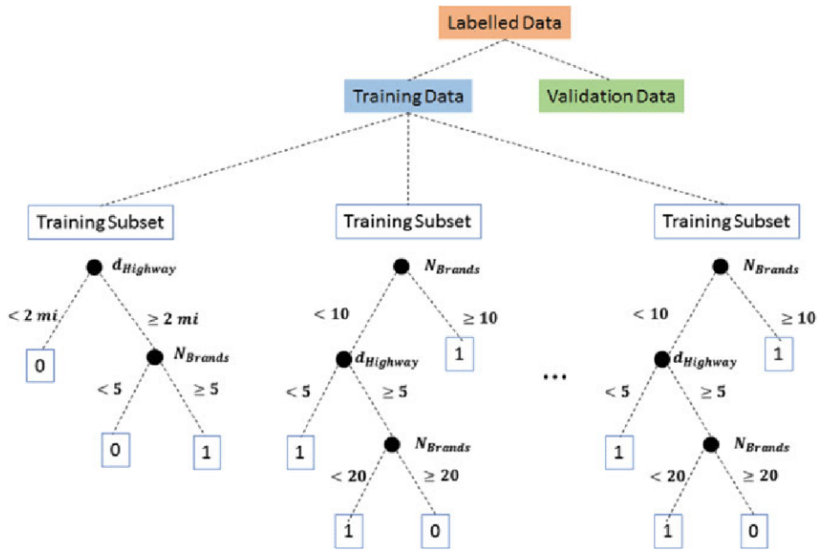


Figura II.4: Exemplo de uma florestas aleatórias - Fonte Pedregosa et al. [2011]

1. Razões centradas nos dados: os segmentos de texto podem ser pequenos, como em microblogs e tweets. Nestes casos, não há informações suficientes dentro da representação "bag-of-words" para fazer inferências.

2. Razões centradas na aplicação: Muitas aplicações, como sumarização de textos, extração de informações, mineração de opiniões e resposta a perguntas, requerem entendimento semântico e este só pode ser obtido tratando as frases como sequências.

O autor argumenta que a ordem das palavras transmite um significado semântico que não pode ser captado pela representação BoW. Por exemplo, as frases "O gato perseguiu o rato" e "O rato perseguiu o gato" são idênticas do ponto de vista dessa representação, mas elas possuem significados diferentes, evidenciando a limitação da abordagem em compreender o contexto de uma sequência de palavras. Para superar essas limitações, o autor propõe duas abordagens principais: métodos específicos da linguagem e métodos independentes da linguagem.

A abordagem específica da linguagem, ou *language-specific*, utiliza, segundo Aggarwal [2018], características linguísticas diretamente extraídas das línguas, como as partes do discurso e as regras gramaticais. Esses métodos, embora eficazes, requerem um conhecimento específico da língua em que está sendo aplicado o método, o que os limita em generalização para outros idiomas. O autor destaca que a abordagem independente da linguagem, ou *Language-independent*, busca representar os textos por meio de modelos estatísticos que não dependem de regras linguísticas específicas. Nesse contexto, métodos como unigramas, bigramas e n-gramas, bem como as redes neurais, são utilizados para aprender padrões a partir de grandes volumes de dados textuais.

Ainda segundo o autor, dentro das abordagens independentes da língua, o uso de redes neurais tem-se mostrado altamente eficaz. Modelos como Word2Vec e Doc2Vec são capazes de aprender re-

presentações semânticas, permitindo que palavras com significados semelhantes sejam representadas em um espaço vetorial próximo, independentemente da língua. Esse aprendizado de representações de palavras, frases e até documentos inteiros se dá a partir da análise estatística da sequência de palavras, sem a necessidade de intervenções manuais ou regras explícitas, tornando os modelos mais flexíveis e escaláveis.

Modelos de linguagem estatística atribuem uma probabilidade a uma sequência de palavras. Para Aggarwal [2018], dada uma sequência de palavras $w_1, w_2, \ldots, w_m$, o modelo de linguagem estima sua probabilidade $P(w_1, w_2, \ldots, w_m)$. Este modelo é utilizado para prever a probabilidade de ocorrência de sequências de palavras, considerando que palavras gramaticalmente corretas e que ocorrem com alta frequência em um conjunto de dados tendem a ter altas probabilidades. Esses modelos são amplamente utilizados em tarefas como reconhecimento de fala, tradução automática, marcação de partes do discurso, entre outros. A equação básica para o cálculo dessa probabilidade é dada por:

$$P(w_1, \ldots, w_m) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1, w_2) \ldots P(w_m|w_1, \ldots, w_{m-1}) \tag{II.9}$$

Essa equação ilustra como a probabilidade de uma sequência de palavras pode ser decomposta em uma série de probabilidades condicionais.

O trabalho de Aggarwal [2018] destaca que a principal dificuldade na utilização de modelos de linguagem estatística é a estimativa das probabilidades condicionais $P(w_i|w_1, \ldots, w_{i-1})$, especialmente em contextos de dados escassos. Em grandes corpora de texto, muitas combinações de palavras são raras ou inexistem, o que dificulta a estimativa precisa de suas probabilidades. Um modelo de n-grama (por exemplo, bigramas e trigramas) é utilizado para superar esse problema, ao capturar dependências entre palavras em uma sequência limitada, sem considerar toda a sequência. No caso de bigramas, a estimativa é dada por:

$$P(w_i|w_1, \ldots, w_{i-1}) \approx P(w_i|w_{i-1}) \tag{II.10}$$

Isso reduz a complexidade do problema, mas também pode perder informações importantes sobre dependências mais longas.

Ainda segundo o autor, em modelos de linguagem com n-grama, a suavização é uma técnica fundamental para lidar com a escassez de dados. Um exemplo é a suavização de Laplace, que adiciona uma constante β para suavizar as estimativas de probabilidade de transições raras ou inexistentes. A fórmula com suavização de Laplace para o bigrama é dada por:

$$P(w_i|w_{i-1}) = \frac{\text{Count}(w_{i-1}, w_i) + \beta}{\sum_{w_i} \text{Count}(w_{i-1}, w_i) + \beta \cdot |V|} \tag{II.11}$$

Onde β é o parâmetro de suavização e $|V|$ é o tamanho do vocabulário. Isso assegura que todas as transições possíveis tenham uma probabilidade maior que zero, mesmo que não tenham ocorrido no conjunto de treinamento.

Em seu trabalho, Aggarwal [2018] argumenta que os modelos de skip-gram são utilizados para lidar com a escassez de dados e melhorar a captura de contexto ao prever palavras em uma sequência, considerando o contexto ao redor. Esse modelo é útil para representar palavras como vetores, possibilitando a criação de *embeddings* que capturam as relações semânticas entre elas. *Embeddings* são representações numéricas de dados, utilizadas para transformar informações textuais, imagens ou outros tipos de dados categóricos em vetores contínuos de dimensão fixa. A equação para o modelo skip-gram é dada por:

$$S(k, n) = \{w_{i-k} w_{i-k+1} \dots w_{i+k}\} \tag{II.12}$$

O modelo skip-gram permite que palavras distantes na sequência sejam consideradas, enriquecendo a representação semântica ao prever o contexto das palavras de forma mais flexível.

Existem diferentes abordagens para criar modelos de linguagem segundo o autor, como:

- Modelos Baseados em Kernel: Utilizam a similaridade entre frases ou documentos para criar representações multidimensionais.
- Modelos Semânticos de Distribuição: Usam a contagem de palavras para criar embeddings, utilizando a matriz de co-ocorrência de palavras.
- Modelos de Redes Neurais Contextuais: Usam redes neurais para aprender representações de palavras em um contexto específico, como modelos word2vec.
- Modelos de Redes Neurais Recorrentes: Focados em embeddings de frases, esses modelos utilizam redes neurais recorrentes para capturar a estrutura sequencial dos textos.

Conforme apontado por Aggarwal [2018], os métodos Kernel são comumente utilizados em modelos centrados em sequências, como o aprendizado de máquina em texto. Esses métodos extraem embebimentos de texto em nível de sentença ou documento, com o intuito de capturar semelhanças entre sentenças e documentos, que podem então ser processados para computar representações vetoriais. Como exemplo, para a construção de uma matriz de similaridade S, baseada na comparação entre sentenças ou documentos, usa-se uma decomposição matricial que resulta na expressão $S \approx UU^T$, onde U é uma matriz de embebimentos de $N \times K$. Aqui, N representa o número de sentenças e K a dimensão do espaço de embebimento.

Além disso, segundo o autor, a matriz de similaridade S pode ser fatorada de forma simétrica utilizando a matriz U, como mostrado na equação (10.7), ou através da fatoração $S \approx Q\Sigma^2 Q^T$ con-

forme exemplificado na equação (10.8). Isso pode ser feito aplicando uma transformação ortogonal à matriz U para manter a integridade dos fatores simétricos, utilizando um vetor de características Q e uma matriz diagonal Σ.

Os métodos kernel podem ser aplicados de diferentes maneiras, dependendo da natureza da representação dos dados e da tarefa de aprendizagem. A seguir estão algumas abordagens principais elencadas porAggarwal [2018]:

- Métodos baseados diretamente em n-grams e skip-grams: Esta abordagem busca enriquecer a representação vetorial de texto com n-grams e skip-grams, calculando a similaridade entre pares de documentos a partir da comparação dessas representações. Essa técnica envolve a decomposição da matriz de similaridade S em uma fatoração simétrica que resulta em um embebimento de documentos, como ilustrado na equação $S \approx UU^T$.

- Métodos baseados em subsequências de strings (String Subsequence Kernels): Esses métodos, discutidos em detalhe no Capítulo 3, são capazes de lidar com a omissão de palavras ao usar um fator de decaimento para permitir que um skip-gram seja aprendido. Isso resulta em uma técnica robusta que incorpora conhecimento linguístico no modelo de embebimento.

- Métodos baseados em gramáticas específicas de linguagens: Outra possibilidade é incorporar diretamente conhecimento linguístico no kernel, criando árvores de análise baseadas em gramáticas de constituintes. Essa abordagem é usada para melhorar a extração de informações, aproveitando métodos como a convolução de árvores (tree kernels).

Em todos esses casos, a técnica de fatoração simétrica se destaca, permitindo representar documentos ou sentenças de maneira eficiente e extrair relações importantes através de diferentes representações vetoriais. Tais métodos são amplamente utilizados em tarefas de aprendizado em sequência e bastante eficazes quando aplicados em conjunto com modelos de redes neurais e outros modelos de aprendizado de máquina.

II.6 Modelos de Fatoração de Matrizes Word-Context

Os modelos de fatoração de matrizes de palavras-contexto computam uma matriz baseada nas contagens dos elementos textuais para cada palavra-alvo [Aggarwal, 2018]. A ideia central é criar uma representação que captura o contexto das palavras em relação à sua coocorrência no texto. Nesse modelo, o contexto de uma palavra-alvo é geralmente definido em termos de uma janela de palavras ao redor dela, e o tamanho da janela pode variar entre 2 e 10, dependendo do tamanho dos dados de treinamento.

Segundo Aggarwal [2018], a forma mais simples de realizar a fatoração é utilizando uma matriz de contagens de coocorrência de palavras, uma abordagem chamada modelo HAL (Hyperspace Analogue to Language). Nesse modelo, a coocorrência é calculada considerando as palavras que aparecem dentro de uma janela ao redor de uma palavra-alvo. Para cada palavra i, se c_{ij} representa o número de vezes que a palavra j aparece no contexto de i, a matriz C de coocorrência é gerada. Essa matriz pode ser fatorada nas matrizes U e V, conforme apresentado pela equação:

$$C \approx UV^T \tag{10.9}$$

Onde U contém as representações de p-dimensões das palavras e V contém as representações dos contextos.

Conforme apontado por Aggarwal [2018], o modelo de fatoração de matriz PPMI (Positive Pointwise Mutual Information) é um método que considera a informação mútua entre as palavras. Nesse modelo, as entradas da matriz de coocorrência são ajustadas com um valor de regularização, visando à redução da influência de palavras muito frequentes. A equação para a construção da matriz PPMI é dada por:

$$PPMI(i,j) = \max\left(\log\left(\frac{N \cdot c_{ij}}{f_i \cdot f_j}\right) - \log(k), 0\right) \tag{10.15}$$

Onde c_{ij} é a coocorrência de palavras, f_i e f_j são as frequências das palavras no corpus, e k é um parâmetro que pode ser ajustado para controlar a esparsidade da matriz.

Além das contagens de palavras, o trabalho de Aggarwal [2018] destaca que os modelos de fatoração de matrizes também podem incorporar informações sintáticas e outras características. A matriz C pode não ser quadrada, e pode representar outras características contextuais além de palavras, como entidades ou partes do discurso. Uma das vantagens desses modelos é a flexibilidade para incorporar diferentes tipos de informações no processo de aprendizado.

O autor cita ainda que o modelo GloVe (Global Vectors for Word Representation) é uma variação da fatoração de matrizes de palavras-contexto que utiliza uma normalização logarítmica para atenuar o impacto de palavras frequentes. Esse modelo propõe duas modificações principais:

- A entrada c_{ij} é substituída por $\log(1 + c_{ij})$ para reduzir a influência de palavras muito frequentes.
- O erro da entrada c_{ij} na função de otimização é ponderado de acordo com o número máximo de co-ocorrências M e o parâmetro α.

A função de erro otimizada é dada por:

$$\text{Minimizar } J = \sum_i \sum_j \text{Weight}(i,j) \left[\log(1 + c_{ij}) - (b_i + b_j - u_i^T v_j)^2\right] \tag{10.12}$$

Onde u_i e v_j representam os embeddings das palavras e contextos, respectivamente.

Vários problemas de pós-processamento são comuns a todos os métodos de incorporação de palavras, incluindo os métodos de redes neurais. Um problema recorrente, para o autor, é como combinar as matrizes U e V para criar um embedding único, especialmente quando essas matrizes podem ser normalizadas de maneiras diferentes. A fatoração da matriz C pode ser ajustada para incluir uma combinação das duas matrizes, garantindo um embedding único e robusto.

II.6.1 Representação Gráfica de Distância de Palavras

Segundo Aggarwal [2018], a abordagem de Representação Gráfica de Distâncias de Palavras propõe o uso de grafos para capturar relações semânticas entre palavras, levando em consideração não apenas a proximidade das palavras, mas também a ordem delas em uma sequência. Essa técnica se destaca por permitir a criação de representações mais refinadas em relação aos métodos tradicionais de janela de contexto, como os modelos de n-grama. O grafo de distância, em particular, é criado a partir das sentenças individuais, onde cada nó no grafo representa uma palavra e as arestas entre os nós são formadas com base na proximidade das palavras no contexto da sentença. Cada palavra é conectada a outras palavras quando elas aparecem próximas em uma sentença, com o peso das arestas sendo ajustado conforme a distância entre as palavras.

A representação gráfica pode ser configurada com diferentes ordens k, onde um valor de $k = 1$ corresponde a um grafo de distância de bigramas e $k = 2$ pode ser interpretado como um trigrama. A flexibilidade do modelo permite a modulação da granularidade da relação entre as palavras, ajustando o valor de k, que controla a quantidade de palavras consideradas ao criar as arestas entre os nós. Isso permite a criação de grafos de distâncias mais ou menos complexos, dependendo da necessidade da tarefa.

Uma inovação importante dos grafos de distância, destacada pelo autor, é a introdução de um parâmetro de decadência λ, que ajusta a contribuição de palavras mais distantes para a definição das arestas. Quando $\lambda = 1$, o grafo é considerado não decaído e a distância entre as palavras é tratada de forma uniforme. Contudo, ao adicionar o fator de decadência, palavras próximas recebem um peso maior, o que resulta em uma representação mais precisa dos contextos. A equação associada a essa modulação de pesos é dada por:

$$c_{ij} = \sum_{s=1}^{t} \lambda^{r_s+1} \tag{II.13}$$

Onde r_s representa o número de saltos entre as palavras i e j, e λ é o parâmetro de decadência

que ajusta o peso dado à proximidade das palavras.

Além disso, Aggarwal [2018] destaca que os grafos de distância podem ser agregados a nível de documento, criando um grafo único que representa o contexto global de um corpus. Esse processo de agregação pode ser seguido por uma fatoração do grafo de distância, utilizando métodos de fatoração de matrizes para obter representações de palavras que refletem mais precisamente as relações de proximidade e ordem entre elas. As técnicas de mineração de grafos também podem ser aplicadas diretamente nesses grafos de distância, permitindo explorar conexões mais complexas, como a identificação de agrupamentos ou padrões dentro do texto.

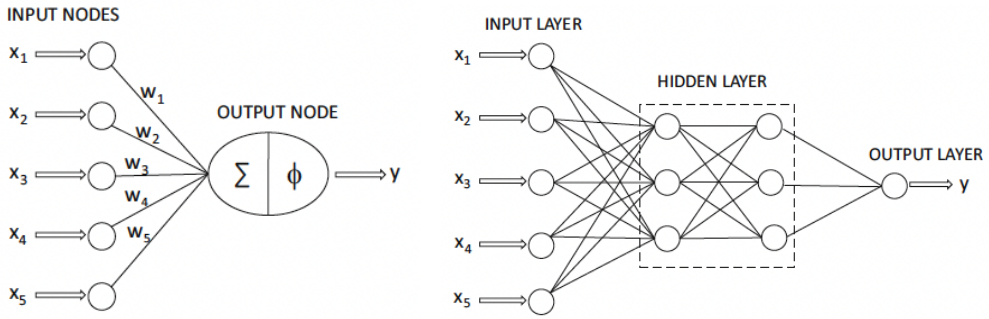
II.6.2 Redes Neurais

Para Aggarwal [2018], Redes Neurais são modelos de simulação do sistema nervoso humano, compostos por células chamadas neurônios. Neurônios biológicos estão interconectados em pontos de contato, denominados sinapses. O aprendizado nos organismos vivos é realizado ao alterar a força dessas conexões entre neurônios. Isso é refletido nas redes neurais artificiais, onde as conexões entre neurônios são modeladas como pesos, que são ajustados em resposta a estímulos externos. Este processo de ajuste de pesos pode ser visto como uma simulação do processo biológico de aprendizado, com o objetivo de aprender representações adequadas dos dados de entrada.

Segundo o autor, os neurônios nas redes neurais artificiais podem ser considerados unidades de computação que recebem entradas, realizam cálculos e enviam as saídas para outros neurônios. A computação realizada em um neurônio é afetada pelos pesos das conexões de entrada. O cálculo no neurônio é escalado pelos pesos associados a cada entrada, que podem ser ajustados para melhorar o desempenho do modelo. Assim como as conexões sinápticas no cérebro, os pesos das redes neurais artificiais são ajustados durante o processo de aprendizado, usando dados de treinamento, para que o modelo faça previsões mais precisas.

Ainda segundo o autor, existem diversas arquiteturas de redes neurais, sendo uma das mais simples o *perceptron*. Redes com muitas camadas, também chamadas de *deep learning*, têm-se mostrado poderosas, pois são capazes de lidar com uma grande variedade de tipos de dados e problemas de aprendizado. Um *perceptron* é um modelo básico, com uma camada de entrada e uma saída binária, enquanto as redes mais complexas, como redes neurais recorrentes, podem ser usadas para tarefas mais sofisticadas, como modelagem de sequência. A Figura II.5 ilustra o modelo básico Perceptron e o modelo Multi-layer, com 3 camadas.

No modelo *perceptron* da Figura II.5, as variáveis de entrada do modelo são x_1, x_2, x_3, x_4, x_5, os pesos sinápticos $(w_1, w_2, w_3, w_4, w_5)$ são os coeficientes associados a cada entrada, responsáveis por determinar a influência de cada x_i na saída. O nodo de saída (*Output Node*) é responsável pelo processamento dos sinais de entrada e pela tomada de decisão, onde a função de soma $(\sum)$ é a soma



(a) Perceptron (b) Three-layer feedforward

Figura II.5: Representação de Redes Neurais - Fonte Aggarwal [2018]

ponderada dos valores de entrada; a função de ativação (ϕ) aplica uma transformação sobre a soma ponderada para determinar a saída y.

Em relação à rede neural feedforward de três camadas da Figura II.5, a camada de entrada (*Input Layer*) é representada pelos nós x_1, x_2, x_3, x_4, x_5, que correspondem às variáveis de entrada do modelo. Cada nó recebe um valor de entrada e o transmite para a próxima camada. A camada oculta (*Hidden Layer*) é constituída por neurônios intermediários responsáveis pelo processamento dos sinais recebidos da camada de entrada. Cada neurônio aplica uma combinação ponderada dos valores de entrada, seguida por uma função de ativação não linear, permitindo que a rede capture padrões complexos nos dados. A camada de saída (*Output Layer*) é responsável por produzir o resultado final da rede. O valor y gerado depende das ativações dos neurônios da camada oculta e pode representar uma classe (para classificação) ou um valor contínuo (para regressão).

Conforme apontado por Aggarwal [2018], a função de ativação do *perceptron* é uma função de sinal, que mapeia as saídas para um valor binário, como -1 ou 1. No caso de múltiplas classes, como na classificação de palavras ou em outros tipos de problemas, a função de ativação *softmax* é frequentemente utilizada para calcular a probabilidade de cada classe. A função de perda mais comum para a rede neural é a *cross-entropy loss*, que mede a diferença entre as previsões feitas pela rede e os valores reais. Essa função de perda é minimizada durante o treinamento para otimizar os pesos da rede e melhorar sua acurácia.

Ainda segundo o autor, a atualização dos pesos em redes neurais é frequentemente realizada através do método de descenso de gradiente estocástico (SGD). Esse método ajusta os pesos da rede com base no erro observado em uma instância de treinamento, seguindo a regra de atualização:

$$W \leftarrow W + \alpha E(X)X \tag{II.14}$$

onde α é a taxa de aprendizado e $E(X)$ é o erro para uma instância de entrada X. A rede percorre repetidamente os dados de treinamento, atualizando os pesos para reduzir o erro.

O autor afirma que a escolha da função de ativação e da função de perda depende do tipo de tarefa e do tipo de dados. Para tarefas de classificação binária, a função de ativação *sigmoide* é uma escolha comum, enquanto para múltiplas classes, a função *softmax* é amplamente utilizada. A função de perda, como a *cross-entropy*, é otimizada para fazer com que o modelo aprenda a prever as classes corretas com alta probabilidade [Aggarwal, 2018].

O *perceptron* tem uma forte semelhança com o modelo de máquinas de vetores de suporte (SVM), especialmente quando a atualização dos pesos do *perceptron* é comparada com a do SVM. Ambos os modelos buscam otimizar uma função de perda semelhante, com a principal diferença sendo que o SVM utiliza uma margem maior para classificações mais precisas, enquanto o *perceptron* foca apenas na minimização do erro para cada instância de treinamento [Aggarwal, 2018].

II.6.3 Redes Neurais Recorrentes (RNNs)

Segundo Aggarwal [2018], Redes neurais recorrentes (RNNs) utilizam laços dentro da arquitetura da rede para modelar dependências temporais. As redes recorrentes recebem uma sequência de entradas e produzem uma sequência de saídas. Esses modelos são particularmente úteis para aprendizado de sequência-a-sequência, como exemplificado nos seguintes cenários:

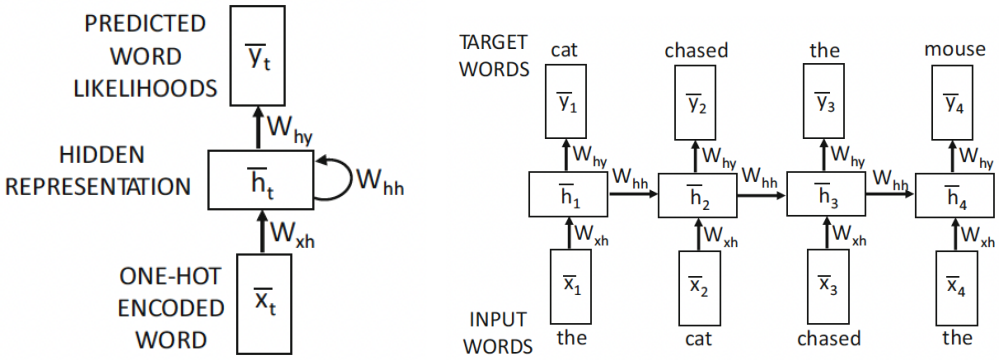
- A entrada pode ser uma sequência de palavras, e a saída pode ser a mesma sequência de palavras deslocada por 1, como em um modelo clássico de linguagem, onde tentamos prever a próxima palavra com base no histórico sequencial de palavras.
- A entrada pode ser uma sentença em um idioma, e a saída pode ser a sentença correspondente em outro idioma, como em tarefas de tradução automática.
- A entrada pode ser uma sequência (por exemplo, uma sentença), e a saída pode ser um vetor de probabilidades de classe, acionado no final da sentença.

O modelo básico de uma RNN é ilustrado na Figura II.6 (a), com a presença de um laço, que permite ao estado oculto mudar após cada entrada na sequência. Este modelo só funciona com sequências de tamanho finito, e é necessário "desenrolar" o laço para transformá-lo em uma rede feedforward. Este processo de "desenrolar" é mostrado na Figura II.6 (b), e os pesos das diferentes camadas temporais são compartilhados.

O autor ilustra que, dada uma sequência de palavras, sua codificação one-hot é alimentada na rede uma a uma, como mostrado na Figura II.6 (a). A saída da rede, no contexto de modelagem de linguagem, será um vetor de probabilidades da próxima palavra na sequência.

Para Aggarwal [2018], a RNN pode ser formalizada matematicamente pela seguinte equação:

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t) \tag{II.15}$$



(a) RNN (b) Time-layered representation

Figura II.6: Representação de Redes Neurais - Fonte Aggarwal [2018]

onde $\mathbf{h}_t$ é o estado oculto no tempo t, e $\mathbf{x}_t$ é o vetor de entrada no mesmo instante. O estado oculto evolui ao longo do tempo à medida que as entradas são alimentadas na rede.

A equação de atualização do estado oculto pode ser expandida como:

$$\mathbf{h}_t = \tanh(W_{xh}\mathbf{x}_t + W_{hh}\mathbf{h}_{t-1}) \tag{II.16}$$

onde W_{xh} é a matriz de pesos entre a entrada e o estado oculto, e W_{hh} é a matriz de pesos entre os estados ocultos em diferentes instantes de tempo.

Para aprender as probabilidades de saída, um vetor de probabilidades $\mathbf{y}_t$ é calculado:

$$\mathbf{y}_t = W_{hy}\mathbf{h}_t \tag{II.17}$$

onde W_{hy} é a matriz de pesos entre o estado oculto e a saída.

Em um modelo de linguagem clássico, uma RNN pode ser treinada para prever a próxima palavra em uma sequência, dado o histórico de palavras anteriores. Como exemplo:

O gato perseguiu o rato.

Dado o início “O”, a rede aprenderá a prever a palavra “gato”, e assim por diante até a palavra “rato”. A RNN, ao processar a sequência uma palavra por vez, ajusta seus estados ocultos e gera uma distribuição de probabilidade para a próxima palavra. Isso é feito com a rede compartilhando os mesmos pesos para cada palavra na sequência.

As RNNs podem ser treinadas usando o algoritmo de retropropagação no tempo (BPTT). No caso da modelagem de linguagem, o objetivo é maximizar a probabilidade das palavras previstas,

ou minimizar a função de perda, que pode ser representada por:

$$L = -\sum_{t} \log P(y_t | x_{1:t})$$

onde y_t é a palavra prevista no tempo t e $x_{1:t}$ é a sequência de entradas até o tempo t.

II.6.4 Redes Neurais Convolucionais

Conforme estudo de Aggarwal [2018], embora as redes neurais convolucionais (CNNs) sejam mais comumente usadas em dados de imagem, elas se tornaram cada vez mais populares para processamento de textos. A entrada para a rede neural convolucional contém uma sequência de palavras, correspondente a uma sentença ou unidade sequencial. Contudo, as palavras não são representadas em codificação one-hot, mas sim por meio de embeddings word2vec.

A operação convolucional é realizada por meio de filtros, que são matrizes bidimensionais de tamanho $F \times d$ contendo pesos treináveis, onde $F \leq L$, e d é a dimensionalidade do embedding word2vec. Para entradas com comprimento menor que L, é possível usar padding, preenchendo com valores de "dummy". O valor de d também é referido como a profundidade da camada de entrada, correspondente ao número de canais da entrada.

A operação convolucional é expressa como:

$$h_{sr} = \sum_{i=1}^{F} \sum_{j=1}^{d} f_{ij} x_{s+i,j} \quad \forall s \in \{1 \dots L - F + 1\}, \forall r \in \{1 \dots k\}$$

onde x_{ij} representa o elemento (i, j)-ésimo da entrada, com i referindo-se à posição da palavra e j ao canal. Os filtros são aplicados para criar uma ativação que será processada pela próxima camada da rede.

Uma operação importante que é comum em redes neurais convolucionais é o *max-pooling*, que também pode ser referido como *max-pooling over time*. A operação max-pooling é aplicada a cada canal de uma camada da rede separadamente. A operação de pooling divide o mapa de entrada em segmentos de tamanho $l/p \times k$, em que p regula a complexidade do embedding.

$$c_{ij} = \sum_{s=1}^{t} \lambda^{r+1}$$

Onde λ é um parâmetro que controla o fator de decaimento do peso entre os elementos do gráfico de distâncias, sendo ajustado para minimizar o impacto de palavras em posições distantes.

Após a aplicação da operação convolucional, ativação e max-pooling, um número de camadas totalmente conectadas é utilizado. Estas camadas são alimentadas com o vetor r-dimensional resultante do max-pooling, e esse vetor é então alimentado na camada totalmente conectada. A entrada

dessa camada corresponde a uma combinação das representações aprendidas das palavras da sentença, sendo usada para predizer a classificação do texto.

Segundo o autor, redes neurais convolucionais possuem a vantagem de serem altamente paralelizáveis, diferentemente das redes neurais recorrentes, o que as torna muito adequadas para o treinamento em grandes conjuntos de dados. O modelo TextCNN, por exemplo, tem sido amplamente usado em aplicações de análise de sentimentos, mostrando boa precisão ao lidar com grandes quantidades de dados textuais.

Para a implementação, a entrada da rede consiste nas palavras de uma sentença, onde cada palavra é representada pelo seu embedding word2vec. O resultado da operação convolucional e do max-pooling é alimentado em camadas totalmente conectadas, que realizam a classificação do texto. Esta abordagem tem mostrado ser eficaz em muitos cenários de processamento de linguagem natural (NLP), principalmente na análise de sentimentos [Aggarwal, 2018].

Para o autor, as redes neurais convolucionais para texto oferecem uma poderosa abordagem para modelar dependências temporais e hierárquicas nas palavras de uma sentença. Sua capacidade de paralelizar o treinamento e aprender representações contextuais é particularmente útil em aplicações que envolvem grandes volumes de dados textuais.

II.6.5 Validação cruzada: processo de avaliação do modelo

Segundo Pedregosa et al. [2011], treinar os parâmetros de uma função de previsão e testá-la nos mesmos dados é um erro metodológico, pois assim o modelo apenas repetiria os rótulos aprendidos nas amostras, mas não conseguiria prever com a mesma avaliação sobre os dados não vistos. Os autores chamam essa situação de *overfitting*, ou sobreajuste, e, segundo eles, para evitar isso, uma prática comum ao realizar um experimento de aprendizado de máquina supervisionado é separar parte dos dados disponíveis como um conjunto de testes e de treinamento. Um fluxograma de validação cruzada no treinamento do modelo pode ser ilustrado pela Figura II.7, indicando que melhores parâmetros podem ser determinados por técnicas de busca em grade ($GridSearchCV$).

Ainda de acordo com os autores, o processo de validação cruzada utiliza várias partições ou divisões (em inglês, *k-folds*) do conjunto de treinamento com o objetivo de identificar os melhores parâmetros. Como exemplo do processo, pode-se ilustrar o procedimento de validação cruzada por meio da Figura II.8.

II.7 Medidas de avaliação de modelos de aprendizado de máquina

Segundo Hossin and M.N [2015], as métricas de avaliação têm por objetivo alcançar o modelo ideal durante o treinamento de classificação de dados. Com o intuito de abordar os principais conceitos envolvendo as medidas de avaliação, os autores destacam a matriz de confusão, as medidas

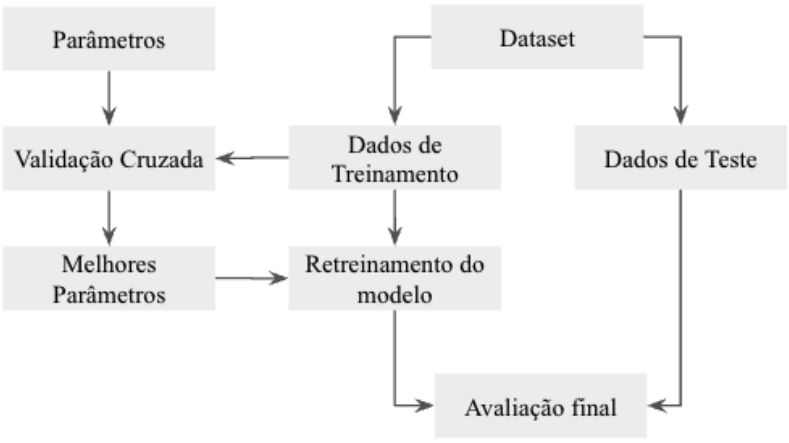


Figura II.7: Processo de validação cruzada. Fonte Pedregosa et al. [2011]

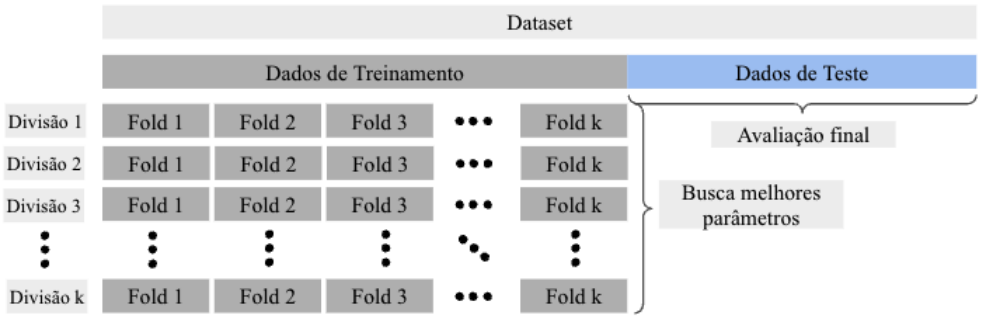


Figura II.8: Processo de validação cruzada. Fonte Pedregosa et al. [2011]

de precisão, revocação e F1-measure.

A matriz de confusão é uma representação dos valores reais e dos valores preditos pelo modelo [Hossin and M.N, 2015]. Segundo os autores, os conceitos da matriz são tais que:

- Verdadeiros Positivos (VP): são os valores reais verdadeiros e o modelo os classificou como sendo verdadeiros;
- Verdadeiros Negativos (VN): são os valores reais falsos e o modelo os classificou como sendo falsos;
- Falsos Positivos (FP): são os valores reais falsos e o modelo os classificou como sendo verdadeiros;
- Falsos Negativos (FN): são os valores reais verdadeiros e o modelo os classificou como sendo falsos;

Com o objetivo de ilustrar uma matriz de confusão para classe binária, adotam-se os conceitos relacionados por Hossin and M.N [2015] por meio da Figura II.9:

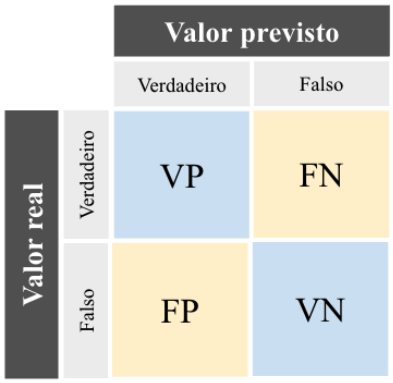


Figura II.9: Matriz de confusão para problemas de classificação binária

A acurácia mede a proporção de predições corretas sobre o número total de instâncias avaliadas [Ricci et al., 2010]. A equação II.18 representa o racional para a acurácia.

$$\text{Acurácia} = \frac{\text{VP} + \text{VN}}{\text{Total}} \tag{II.18}$$

A precisão avalia a fração de previsões positivas do modelo que correspondem a verdadeiros positivos [Ricci et al., 2010]. A equação II.19 ilustra o racional do cálculo da precisão.

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}} \tag{II.19}$$

A revocação avalia a proporção de instâncias positivas corretamente classificadas pelo modelo. A equação II.20 aborda o racional do cálculo da revocação.

$$\text{Revocação} = \frac{\text{VP}}{\text{VP} + \text{FN}} \tag{II.20}$$

A F-measure combina precisão e revocação por meio da média harmônica, proporcionando uma avaliação equilibrada do modelo. A Equação da métrica II.21 é dada por:

$$F_1 = \frac{2 \times \text{ Precisão } \times \text{ Revocação}}{\text{Precisão} + \text{Revocação}} \tag{II.21}$$

Capítulo III Trabalhos relacionados

Neste capítulo, abordam-se alguns trabalhos relacionados à problemática do presente estudo. Inicialmente, descreve-se a abordagem lexical nos modelos de aprendizado de máquina para métodos de classificação de textos. Em seguida, são apresentados alguns estudos orientados a modelos de aprendizado de máquina supervisionados, como modelos tradicionais, modelos ensemble e modelos de aprendizagem profunda. Ao término, um resumo das métricas de avaliação e abordagens é apresentado.

A revisão sistemática é reportada nesse capítulo, com o objetivo de identificar os principais trabalhos relacionados. A string de busca empregada foi *(classification OR detection) AND (homophobia OR homophobic)*, considerando todos os trabalhos escritos nas línguas inglesa e portuguesa, na lista de acervo das bases científicas SCOPUS, IEEE Xplore e ACL Anthology. Ao todo, 203 documentos foram encontrados para a chave em questão, e após isso, cada documento foi classificado seguindo um critério de inclusão e exclusão para textos relacionados ao objetivo do presente estudo, conforme III.1.

Critérios	**Descrição**
CI-MCDOG	Critério de inclusão - O texto refere-se a métodos ou modelos de classificação ou detecção textual do discurso de ódio para classes gerais
CI-MCDOH	Critério de inclusão - O texto refere-se a métodos ou modelos de classificação ou detecção textual do discurso de ódio para a classe Homofobia
CE-NMCDO	Critério de exclusão - O texto não refere-se a métodos ou modelos de classificação ou detecção textual do discurso de ódio

Tabela III.1: Critério de inclusão e exclusão - revisão sistemática

Dos resultados da busca, 112 documentos foram considerados como inclusão, sendo 59 documentos como CI-MCDOG e 53 documentos como CI-MCDOH. O passo seguinte utilizado na seleção dos documentos foi avaliar a significância dos trabalhos, segundo critérios: (a) resultados dos modelos avaliados pela métrica F1-measure; (b) método de classificação e características textuais semelhantes ao presente estudo.

Após a seleção final, foi aplicada a técnica de snowballing para análise das referências e citações dos documentos selecionados, onde foi possível identificar trabalhos relacionados a um conjunto de dados para o discurso homofóbico em português e trabalhos relacionados à aplicação da classificação de textos em português no X (Twitter).

III.1 Dataset para análise de discurso de ódio homofóbico

Segundo de Pelle and Moreira [2017], o aprendizado supervisionado para classificação de conteúdo de discurso de ódio depende da existência de conjuntos de dados rotulados. Devido à ausência de um conjunto de dados disponíveis para o contexto abordado em português, os autores propuseram a criação de um, a partir da coleta dos comentários do portal de notícias G1[1]. Eles coletaram as informações de diversas notícias do portal e identificaram que 90% delas continham ao menos um comentário com discurso de ódio. Dois datasets em português foram gerados a partir dos julgamentos de três pessoas, o primeiro contendo 1.250 registros, no qual o rótulo foi definido quando ao menos duas pessoas concordavam, chamado OFFCOMBR-2, o segundo contendo 1.033 comentários com julgamentos unânimes, chamado OFFCOMBR-3. Os julgadores respondiam se o conteúdo era de discurso de ódio e qual tipo de discurso identificaram (e.g. racismo, religião, blasfêmia, homofobia e outros). Os autores analisaram os rótulos aplicados a cada conjunto de dados, evidenciando que apenas 14 comentários foram julgados como homofóbicos. Ao final, os autores aplicaram dois modelos de classificação a ambos os datasets. Eles observaram que os modelos SVM e NB obtiveram, respectivamente, em 82% e 79% para a métrica F1-measure aplicada ao dataset OFFCOMBR-3. Ambos os datasets podem ser consultados pelo repositório do autor, OFFCOMBR[2].

Utilizando como referência o trabalho feito por de Pelle and Moreira [2017], Fortuna et al. [2019a] expandiram o trabalho de construção de um dataset em português para o discurso de ódio. Eles arbitraram 29 perfis do Twitter e a partir deles fizeram uma busca por 19 termos e 10 *hashtags* para o discurso de ódio. Os autores filtraram por *tweets* em português, entre 8 e 9 de março de 2017 e contendo ao menos 3 palavras. Além disso, eliminaram repetições e retweets para evitar duplicações, removeram html, tags e outros conteúdos não textuais. Após a coleta, eles submeteram as mensagens extraídas ao julgamento de 2 pessoas e utilizaram a métrica Cohen's Kappa para avaliar a concordância entre eles para as classes de discurso de ódio (e.g., homofobia, racismo, religião, imigrantes e outros). No total, o dataset contém 5.668 *tweets*, sendo apenas 365 para o discurso de ódio homofóbico. A base de dados se encontra disponível para livre consulta.

A incipiência de datasets é uma limitação que busca-se resolver com o presente estudo. Mesmo com as contribuições feitas por de Pelle and Moreira [2017] e Fortuna et al. [2019a], poucos são os textos coletados com o rótulo para homofobia, fazendo-se necessário a coleta de mais amostras, abordando diferentes tipos de termos homofóbicos.

[1] `g1.globo.com`
[2] `https://github.com/rogersdepelle/OffComBR`

III.2 Abordagem lexical e Modelos de aprendizagem de máquina

Uma maneira clássica de se identificar o discurso de ódio é usar uma abordagem baseada em palavras-chave, conforme descrito em MacAvaney et al. [2019]. Para os autores, com base em uma ontologia ou dicionário, textos que contenham potencial discurso de ódio podem ser identificados. Um exemplo considerado pelos autores é o vocabulário *Hatebase*. Eles ponderam que apenas o uso do termo presente no texto não é suficiente para definir a presença do discurso de ódio. Além disso, destacam que a utilização dessa abordagem exclusivamente não permite identificar discurso de ódio em textos que não contenham quaisquer palavras-chave de incitação ao ódio, como é o caso do uso de linguagem figurada. Vale destacar que podem ser encontrados 28 termos homofóbicos, em português, no *Hatebase*.

Coutinho and Malheiros [2020] abordaram o tema da detecção de mensagens homofóbicas em português, no *Twitter*, usando análise de sentimentos. Os autores construíram a base de dados consultando a interface do *Twitter* com quatro termos de cunho homofóbico (e.g., *"sapatão", "viado", "traveco" e "gay"*). Após isso, vetorizaram as mensagens utilizando TF-IDF e treinaram um modelo de Regressão Logística aplicados ao TASH-PT, um banco de dados para análise de sentimentos na língua portuguesa com dados coletados do *Twitter*. Os autores relatam que para a construção do TASH-PT foram avaliados 2.787 *tweets*, sendo 888 positivos, 881 negativos e 1.018 neutros. Para validação do modelo, os autores fizeram uso de cinquenta e sete julgadores humanos. Para avaliar a qualidade do modelo de classificação adotado, tais julgadores responderam se cerca de 160 mensagens classificadas pela Regressão Logística eram ou não mensagens homofóbicas. Os resultados observados foram: acurácia de 61,48%, precisão de 66,67%, revocação de 62,16% e *F-Measure* de 64,33%.

Davidson et al. [2017] coletaram dados do Twitter em inglês, a partir do léxico extraído do *Hatebase* para o discurso de ódio em geral. No total, uma amostra de 25 mil tweets foi rotulada por funcionários da empresa *CrowdFlower* (CF), empresa especializada em rotulagem. Apenas 5% dos tweets foram considerados discurso de ódio; eles utilizaram uma métrica de concordância chamada *intercoder-agreement*. Os autores avaliaram diversas abordagens de classificação, como LR, NB, DT, RF e SVM. Cada modelo foi testado utilizando validação cruzada com 5 partições e utilizaram como *features* bigramas, unigramas e trigramas, com pesos calculados por TF-IDF. Ao final, os autores concluíram que o melhor modelo foi a LR com regularização L2, tendo precisão de 91%, revocação de 90% e F1-measure de 90%.

Quatro modelos de aprendizado de máquina para classificação foram estudados em Silva and Roman [2020]. Os autores utilizaram a base de dados em português criada por Fortuna et al. [2019b] de tweets extraídos para o discurso de ódio. Os algoritmos analisados foram NB, SVM, LR e MLP.

Os autores utilizaram como recursos *BOW*, TF-IDF e n-gramas. Para avaliação dos modelos, eles fizeram validação cruzada para dez partições, obtendo como melhor modelo o SVM, com *F-measure* de 71,3%.

Em Burnap and Williams [2015], utilizou-se a abordagem híbrida para um classificador de aprendizado de máquina supervisionado e modelagem estatística aplicados a um conjunto de dados de mensagens do *Twitter*. Eles propuseram uma combinação de classificadores probabilísticos, *rule-based* e *spatial-based*, com um meta-classificador conjunto, *voted ensemble* com máxima probabilidade. Segundo os autores, o meta-classificador utilizou como base os modelos *Bayesian Logistic Regression* (BLR), *Random Forest Decision Tree* (RFDT) e *Support Vector Machine* (SVM). Os resultados obtidos foram: precisão de 95%, revocação de 95% e *F-measure* de 95%.

No estudo de Vandersmissen [2012], foi projetado dois métodos de aprendizado de máquina supervisionado para o conjunto de dados "Dutch Netlog Corpus", a partir da coleta de dados textuais da rede social *Netlog* para língua holandesa. O autor utilizou como rótulos, conteúdos do discurso de ódio racista ou sexual, incluindo homofobia. Ao todo, 535 mensagens foram classificadas manualmente pelo autor para o discurso de ódio, sendo 173 racistas e 362 sexuais. O primeiro classificador aplicado ao conjunto de dados foi Naive Bayes, tendo um desempenho ruim e atingindo uma precisão de 9% e revocação de 86%. O segundo classificador avaliado foi SVM, obtendo 69% de precisão e 62% de revocação. Com o objetivo de otimizar os resultados aferidos, o autor adotou uma abordagem híbrida combinando um classificador semântico com SVM, atingindo precisão de 100%, revocação de 79% e F-measure de 88,3%.

Saha et al. [2019] utilizaram como proposta de estudo casos de estresse da comunidade LGBT por meio dos dados extraídos da plataforma *REDDIT* para a língua inglesa. Os autores coletaram 12.645 mensagens da rede entre janeiro de 2016 e maio de 2018; após isso, aleatorizaram 347 conteúdos a serem rotulados pelos próprios, obtendo um coeficiente Cohen de Kappa de 0,91. Os autores avaliaram alguns modelos de classificação adotando validação cruzada para 5 partições. Os modelos foram NB, LR, RF, SVM e MLP. Ao final, o modelo de melhor performance foi o Multilayer Perceptron (MLP), com precisão de 75%, revocação de 74% e F1-measure de 75%.

Chakravarthi et al. [2022] propuseram um modelo baseado em mBERT para a tarefa de classificar textos com conteúdo homofóbico. A aplicação do modelo envolveu um código misturado multilíngue, ou *multilingual code-mixed*, de línguas indianas (dravídicas) para o inglês. O autor avaliou diversos modelos multilíngues de linguagem, abordando a ampliação de dados por meio de pseudolabeling, transliterando o texto com código misturado para a língua principal. O modelo com melhor desempenho foi o mBERT, com um F1-score de 66,4%.

Observa-se no estudo conduzido por Karayigit et al. [2022], no qual um conjunto de dados Homophobic-Abusive Turkish Comments (HATC) foi coletado da rede social Instagram na língua

turca, com o objetivo de treinar modelos de detecção de Homofobia. Segundo o autor, o conjunto de dados HATC foi balanceado utilizando o método de reamostragem, resultando em duas formas do conjunto de dados: resHATC e HATC original. Para a extração de características, o autor empregou técnicas como GLoVe word embedding, unigram e TF-IDF. O modelo mBERT foi comparado com modelos baseados em *Deep Learning*, como Long-Short Term Memory, Bidirectional Long-Short Term Memory e Gated Recurrent Unit, classificadores de Aprendizado de Máquina Tradicional, como SVM, NB, RF e classificadores ensemble, como AB, eXtreme GB e GB para determinar o melhor modelo. O modelo de melhor desempenho no conjunto de dados HATC foi o mBERT, alcançando um F1-score de 82,6%, enquanto para o conjunto de dados reamostrado (resHATC), o melhor modelo foi o mBERT com um F1-score de 80,9%.

Os estudos de Kumar et al. [2021], Cohn et al. [2015] e Ninalga [2023] foram aplicados à língua inglesa e, além disso, o estudo de Ninalga [2023] também aborda a aplicação em línguas indianas (Dravídicas). Os trabalhos de Akhtar et al. [2019], Goffredo et al. [2022], Ponnusamy et al. [2023] e Musto et al. [2019] são realizados na língua italiana. Finalmente, o estudo de Vásquez et al. [2023] é realizado na língua espanhola (México).

O modelo de classificação de texto com melhor desempenho para a língua italiana é citado em Musto et al. [2019]. O autor utiliza o modelo HanSEL, alcançando uma pontuação de F1 de 0.840. Para a língua espanhola, o melhor modelo de classificação de texto é citado em Vásquez et al. [2023], onde a aplicação do modelo beto-cased superou os outros, alcançando uma pontuação de F1 de 0.860. Em relação às línguas inglesa e dravídica, os modelos que apresentaram o melhor desempenho estão detalhados no trabalho de Ninalga [2023]: o modelo Bernice+LangPAINT obteve as maiores pontuações de F1-measure para inglês, hindi, malaiala e tâmil, com 94%, 93,2%, 93% e 89,5%, respectivamente.

Os melhores modelos identificados para cada língua empregaram abordagens distintas, destacando a influência da língua na seleção do modelo. Em Musto et al. [2019], o autor explora técnicas de Análise de Sentimentos (i.e., SentipolC e HanSEL) e Lexicons: HateMap lexicon e lexicon completo (HateMap+Hurtlex) para o italiano. O algoritmo HanSEL, conforme explicado pelo autor, utiliza uma rede neural profunda C-BiLSTM com uma camada de entrada de embeddings de palavras, enquanto o SentiPolC é um algoritmo de análise de sentimentos encarregado de atribuir o sentimento correto aos tweets.

Em Vásquez et al. [2023], o autor propõe a criação de um Corpus anotado em espanhol Mexicano para detecção de LGBT+fobia. Para isso, o autor selecionou anotadores candidatos das comunidades heterossexuais e LGBT+. Esses anotadores rotularam 11.000 tweets como "LGBT+fóbicos", "Não LGBT+fóbicos" ou "irrelevantes para a comunidade LGBT+".

Em Ninalga [2023], foi aplicado um algoritmo de machine learning Bernice, um modelo de lingua-

gem treinado exclusivamente com dados do Twitter em múltiplas línguas. Os autores apresentam o Language-PAINT, um método para modelar conjuntamente o conhecimento *multilingual* (M-L) e *language-specific* (L-S) por meio de interpolação de pesos.

Estudos aplicados ao banco de dados do YouTube tiveram como objetivo abordar a tarefa compartilhada no Workshop realizada em três eventos: *Workshop on Language Technology for Equality, Diversity, Inclusion* (LT-EDI-2022[3], LT-EDI-2023[4] e LT-EDI-2024[5]). Os autores submeteram seus trabalhos baseados no conjunto de dados publicado em Chakravarthi et al. [2021] para a rede social YouTube, a partir dos comentários feitos por usuários nos vídeos publicados na referida rede.

O desafio lançado por esses respectivos eventos foi centrado em textos rotulados como Homofobia, para textos em inglês e línguas indianas (Dravídicas). Os participantes foram incumbidos de criar um modelo de classificação textual que pudesse alcançar métricas de sucesso ótimas, com foco particular no F1-score. Com base nas submissões recebidas, um ranking foi compilado para cada evento, apresentando o modelo de melhor desempenho juntamente com suas métricas de sucesso correspondentes.

Os autores utilizaram diversos modelos, incluindo o modelo Multilingual Bidirectional Encoder Representations from Transformers (mBERT), Redes Neurais, Perceptron Multicamadas (MLP), Descida Estocástica do Gradiente (SGD), algoritmos tradicionais de aprendizado de máquina (por exemplo, Máquinas de Vetores de Suporte, Regressão Logística e Random Forest), e um modelo de LLM, o Generative Pre-trained Transformer 2 (GPT-2). No campo da extração de características, as técnicas predominantes incluem GLoVe word embedding, fastText, TF-IDF, word-2-vec, n-grams, e Perceptron Multicamadas.

O estudo de Andrew [2023], utilizando o modelo GPT-2, apresentou um desempenho consideravelmente alto para o idioma inglês, com um F-measure de 90%, embora tenha sido baixo para as línguas dravídicas. Entre os trabalhos relacionados, este foi o único estudo que utilizou um LLM.

III.3 Síntese dos Trabalhos Relacionados

A revisão sistemática evidenciou a incipiência de estudos para o discurso de ódio homofóbico para a língua portuguesa. As buscas realizadas por Coutinho and Malheiros [2020] e Fortuna et al. [2019a] na etapa de coleta de dados basearam-se em um número restrito de termos lexicais, utilizando 4 e 19 termos, respectivamente. Ainda nesse contexto, o *Hatebase* é o único que dispõe de termos e expressões homofóbicos em português, com 28 termos. A presente pesquisa tem por objetivo específico criar um léxico homofóbico, abordando uma maior variedade de termos ou expressões em relação *Hatebase*.

[3] `https://sites.google.com/view/lt-edi-2022/home`
[4] `https://sites.google.com/view/lt-edi-2023/home`
[5] `https://sites.google.com/view/lt-edi-2024/home`

Os trabalhos relacionados ao PLN permitem ilustrar as técnicas utilizadas nos mais diversos contextos de discurso de ódio, em diversas línguas. A Tabela III.2 tem por objetivo resumir as informações sobre as abordagens utilizadas pelos autores dos trabalhos relacionados, bem como a Base de Dados utilizada, a Língua (idioma) de abordagem, os modelos e as métricas de avaliação.

Uma análise das línguas abordadas nos trabalhos elencados na Tabela III.2 revela uma clara predominância de artigos em Inglês (42,27%) e em Línguas Indianas (Dravídicas) (41,24%). As outras línguas, como Espanhol (7,22%), Italiano (4,12%), Português (3,09%), Holandês (1,03%) e Turco (1,03%), possuem representatividade significativamente menor, destacando uma concentração de estudos em contextos globais e específicos da Índia.

O trabalho de de Pelle and Moreira [2017] construiu um dataset com base no site de notícias G1, conforme mencionado na Seção III.1. Após a anotação, o autor aplicou o modelo de classificação SVM e obteve o melhor resultado para a métrica F-measure, com 82,0%. A presente pesquisa utiliza desse trabalho para comparar os resultados obtidos.

Diferentemente dos estudos existentes, este trabalho propõe expandir o léxico de termos homofóbicos, atualmente limitado a apenas 28 termos no Hatebase; criar uma base de dados rotulada específica para discurso de ódio homofóbico, suprindo a limitação encontrada no trabalho de de Pelle and Moreira [2017] onde apenas 365 textos foram rotulados para homofobia; e avaliar algoritmos de classificação estado-da-arte para a tarefa de classificação de textos.

Autor e obra	Base de dados	Língua	Modelo	Precisão	Revocação	F-measure
Burnap and Williams [2015]. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making.	X ou Twitter	Inglês	Classificadores probabilíticos, rule-based e spatial-based, com meta-classificador conjunto, voted ensemble e máxima probabilidade.	95,0%	95,0%	95,0%
Coutinho and Malheiros [2020]. Detecção de mensagens homofóbicas em português no twitter usando análise de sentimentos.	X ou Twitter	Português	Regressão Logística aplicados ao TASH-PT	66,7%	62,2%	64,3%
Davidson et al. [2017]. Automated hate speech detection and the problem of offensive language	X ou Twitter	Inglês	Regressão Logística com regularização L2	91,0%	90,0%	90,0%
de Pelle and Moreira [2017]. Offensive comments in the brazilian web: a dataset and baseline results	Site de Notícias G1	Português	Linear SVM	-	-	82,0%
Saha et al. [2019]. The language of LGBTQ+ minority stress experiences on social media	Reddit	Inglês	Multilayer Perceptron (MLP)	75,0%	74,0%	75,0%
Silva and Roman [2020]. Hate Speech detection in portuguese with naive bays, svm, MLP e Logistic Regression.	X ou Twitter	Português	Abordagem SVM combinada com BOW	-	-	71,3%
Vandersmissen [2012]. Automated detection of offensive language behavior on social networking sites.	Dutch Netlog Corpus	Holandês	Abordagem híbrida combinando um classificador semântico com SVM	100,0%	79,0%	88,3%
Cohn et al. [2015]. AttitudeBuzz: Using Social Media Data to Localize Complex Attitudes	X ou Twitter	Inglês	SVM	-	-	Não informado
Musto et al. [2019]. HateChecker: A tool to automatically detect hater users in online social networks	X ou Twitter	Italiano	HanSEL	-	-	84,0%
Akhtar et al. [2019]. A New Measure of Polarization in the Annotation of Hate Speech	X ou Twitter	Italiano	SVM+P-maxfilter +replication	-	-	33,7%
Kumar et al. [2021]. A Deep Multi-modal Neural Network for the Identification of Hate Speech from Social Media	X ou Twitter	Inglês	Deep multi-moda System VGG-16+CNN	-	-	57,0%
Karayigit et al. [2022]. Homophobic and Hate Speech Detection Using Multilingual-BERT Model on Turkish Social Media	Instagram	Turco	M-BERT	-	-	HATC: 82,6% resHATC: 80,9%
Goffredo et al. [2022]. Counter-TWIT: An Italian Corpus for Online Counterspeech in Ecological Contexts	X ou Twitter	Italiano	AlBERTo	-	-	74,2%
Vásquez et al. [2023]. HOMO-MEX: A Mexican Spanish Annotated Corpus for LGBT+phobia Detection on Twitter	X ou Twitter	Espanhol	beto-cased	-	-	86,0%
Ponnusamy et al. [2023]. Team_Tamil at HODI: Few-Shot Learning for Detecting Homotransphobia in Italian Language	X ou Twitter	Italiano	twitter-xlm-roberta base-sentiment-finetunned Few-Shot (400)	-	-	74,0%
Ninalga [2023]. Cordyceps@LT-EDI: Patching Language-Specific Homophobia/-Transphobia Classifiers with a Multilingual Understanding	X ou Twitter	Inglês e Línguas Indianas (Dravídicas)	Bernice+LangPAINT(LP) Bernice+M-L	-	-	Inglês (LP): 94,0% Hindi (LP): 93,2% Malayalam (LP): 93,0% Spanish (M-L): 93,2% Tamil (LP): 89,5%
Varsha et al. [2022]. Sentiment Analysis and Homophobia detection of YouTube comments in Code-Mixed Dravidian Languages using machine learning and Transformer models	YouTube	Inglês e Línguas Indianas (Dravídicas)	BERT Multilayer Perceptron	-	-	Malayalam: 95% Tamil: 88% Tamil-Inglês: 86%
Upadhyay et al. [2022]. Sammaan@LT-EDI-ACL2022: Ensembled Transformers Against Homophobia and Transphobia	YouTube	Inglês e Línguas Indianas (Dravídicas)	Ensemble IndicBERT	-	-	Inglês (Ensemble): 49% Tamil (IndicBERT): 55% Tamil-Inglês (IndicBERT): 35%
Ugursandi and Kumar [2022]. Sentiment Analysis and Homophobia Detection of YouTube Comments	YouTube	Inglês e Línguas Indianas (Dravídicas)	6-layer deep learning	-	-	Malayalam-Englsih: 98% Tamil-Inglês: 54%
Swaminathan et al. [2022]. SSNCSE_NLP@LT-EDI-ACL2022: Homophobia/Transphobia Detection in Multiple Languages using SVM Classifiers and BERT-based Transformers	YouTube	Inglês e Línguas Indianas (Dravídicas)	SVM-GloVe SVM-TF-IDF LaBSE	-	-	Inglês (SVM-GloVe): 93% Tamil (SVM-TF-IDF): 75% Tamil-Inglês (LaBSE): 87%
Sivaprasath et al. [2022]. Homophobia, Transphobia Detection in Tamil, Malayalam, Inglês Languages using Logistic Regression and Code-Mixed Data using AWD-LSTM	YouTube	Inglês e Línguas Indianas (Dravídicas)	Logistic Regression AWD-LSTM	-	-	Inglês (Logistic Regression): 42% Tamil (Logistic Regression): 31% Tamil-Englsih (AWD-LSTM): 33% Malayalam (Logistic Regression): 97%

Tabela III.2: Resumo do levantamento bibliográfico

Autor e obra	Base de dados	Língua	Modelo	Precisão	Revocação	F-measure
Singh and Motlicek [2022]. IDIAP Submission@LT-EDI-ACL2022: Homophobia/Transphobia Detection in social media comments	YouTube	Inglês e Línguas Indianas (Dravídicas)	XLM ROBERTA	-	-	Tamil: 40% Tamil-Inglês: 85% Inglês: 89%
Saumya et al. [2022]. Sentiment and Homophobia Detection on YouTube using Ensemble Machine Learning Techniques	YouTube	Inglês e Línguas Indianas (Dravídicas)	Stacking Ensemble (SE) Passive Aggressive (PA) Gradient Boosting (GB)	-	-	Tamil (SE): 76% Inglês (SE): 98% Malayalam (PA): 93% Tamil-Inglês (GB): 89%
Samuel et al. [2022]. A Sequential DNN for Sentiment Analysis of Dravidian Code-Mixed Language Comments on YouTube	YouTube	Línguas Indianas (Dravídicas)	Sequential DNN LSTM layer	-	-	Tamil-Inglês: 20% Malayalam-Inglês: 48% Kannada-Inglês: 47%
Rajalakshmi et al. [2022]. DLRG@DravidianLangTech-ACL2022: Abusive Comment Detection in Tamil using Multilingual Transformer Models	YouTube	Inglês e Línguas Indianas (Dravídicas)	RF BiLSTM-fastText	-	-	Tamil (BiLSTM-fastText): 70% Tamil-Inglês (RF): 78%
Pranith et al. [2022]. Homophobia and Transphobia Detection of Youtube Comments in Code-Mixed Dravidian Languages using Deep learning	YouTube	Inglês e Línguas Indianas (Dravídicas)	IndicBERT LaBSE	-	-	Inglês (LaBSE): 46% Malayalam (IndicBERT): 54% Tamil-Inglês (IndicBERT): 39% Tamil (IndicBERT): 28%
Nozza [2022]. Nozza@LT-EDI-ACL2022: Ensemble Modeling for Homophobia and Transphobia Detection	YouTube	Inglês e Línguas Indianas (Dravídicas)	Majority Voting Ensemble	-	-	49%
Nilsson et al. [2022]. Leveraging Sentiment Data for the Detection of Homophobic/Transphobic Content in a Multi-Task, Multi-Lingual Setting Using Transformers	YouTube	Inglês e Línguas Indianas (Dravídicas)	XLM-RoBERTA	-	-	Inglês: 49% Malayalam: 70% Tamil: 33% Tamil-Inglês: 55%
Muti et al. [2022]. LeaningTower@LT-EDI-ACL2022: When Hope and Hate Collide	YouTube	Inglês	BERT	-	-	46%
Maimaitituoheti et al. [2022]. ABLIMET @LT-EDI-ACL2022: A RoBERTa based Approach for Homophobia/Transphobia Detection in Social Media	YouTube	Inglês e Línguas Indianas (Dravídicas)	RoBERTa	-	-	Inglês: 57% Tamil: 75% Tamil-Inglês: 53%
Hegde and Shashirekha [2022]. Leveraging Dynamic Meta Embedding for Sentiment Analysis and Detection of Homophobic/-Transphobic Content in Code-mixed Dravidian Languages	YouTube	Inglês e Línguas Indianas (Dravídicas)	LSTM	-	-	Inglês: 74% Malayalam: 37% Tamil: 36% Tamil-Inglês: 58%
García-Díaz et al. [2022]. UMUTeam@LT-EDI-ACL2022: Detecting homophobic and transphobic comments in Tamil	YouTube	Inglês e Línguas Indianas (Dravídicas)	BERT	-	-	Inglês: 45% Tamil: 82% Tamil-Inglês: 58%
Fadil and Lavanya [2022]. Sentiment Analysis of YouTube comments in Dravidian Code-Mixed Language using Deep Neural Network	YouTube	Línguas Indianas (Dravídicas)	DNN	-	-	Tamil: 42% Malayalam: 52% Kannada: 48%
Chanda et al. [2022]. Sentiment Analysis and Homophobia detection of Code-Mixed Dravidian Languages leveraging pre-trained model and word-level language tag	YouTube	Inglês e Línguas Indianas (Dravídicas)	mBERT word-level language	-	-	Inglês: 34% Tamil: 29% Tamil-Inglês: 33% Malayalam: 43%
Chakravarthi et al. [2022]. How can we detect Homophobia and Transphobia? Experiments in a multilingual code-mixed setting for social media governance	YouTube	Inglês e Línguas Indianas (Dravídicas)	mBERT	-	-	66%
Bharathi and Varsha [2022]. SSNCSE NLP@ TamilNLP-ACL2022: Transformer based approach for detection of abusive comment for Tamil language	YouTube	Inglês e Línguas Indianas (Dravídicas)	bert-base-uncased	-	-	Tamil: 96% Tamil-Inglês: 59%
Bhandari and Goyal [2022]. bitsa_nlp@LT-EDI-ACL2022: Leveraging Pretrained Language Models for Detecting Homophobia and Transphobia in Social Media Comments	YouTube	Inglês e Línguas Indianas (Dravídicas)	mBERT	-	-	Inglês: 42% Tamil: 64% Tamil-Inglês: 58%
Balaji and Chinmaya [2022]. A Study on Sentimental Analysis, Homophobia-Transphobia Detection for Dravidian Languages	YouTube	Inglês e Línguas Indianas (Dravídicas)	LightGBM	-	-	Inglês: 49% Malayalam: 94% Tamil: 23% Tamil-Inglês: 32%
Ashraf et al. [2022]. NAYEL @LT-EDI-ACL2022: Homophobia/Transphobia Detection for Equality, Diversity, and Inclusion using SVM	YouTube	Inglês e Línguas Indianas (Dravídicas)	SVM-TF-IDF	-	-	Inglês: 39% Tamil: 84% Tamil-Inglês: 51%

Tabela III.3: Continuação da Tabela Resumo do levantamento bibliográfico

Autor e obra	Base de dados	Língua	Modelo	Precisão	Revocação	F-measure
Wong et al. [2023]. cantnlp@LT-EDI-2023: Homophobia/Transphobia Detection in Social Media Comments using Spatio-Temporally Retrained Language Models	YouTube	Inglês e Línguas Indianas (Dravídicas)	XLM-RoBERTa Retrained Script-Mixed	-	-	Inglês (Retrained): 94% Hindi (Script-Mixed): 97% Malayalam (Script-Mixed): 94% Spanish (Retrained): 00% Tamil (Script-Mixed): 93%
Vaidhegi et al. [2023]. SSNTech2@LT-EDI-RANLP-2023: Homophobia/-Transphobia Detection in Social Media Comments Using Linear Classification Techniques	YouTube	Inglês	SGD	-	-	57%
Shwetha et al. [2023]. Tercet@LT-EDI: Homophobia/Transphobia Detection in social media comment	YouTube	Inglês	SVM with LaBSE	-	-	95%
Sharma et al. [2023]. Detection of Homophobia and Transphobia in Malayalam and Tamil: Exploring Deep Learning Methods	YouTube	Línguas Indianas (Dravídicas)	IndicBERT	-	-	Malayalam: 86% Tamil: 77%
Shanmugavadivel et al. [2023]. KEC AI NLP@DravidianLangTech:Abusive Comment Detection in Tamil Language using Machine Learning Techniques	YouTube	Línguas Indianas (Dravídicas)	Support Vector Classifier	-	-	99%
Sheik et al. [2023]. Mitigating Abusive Comment Detection in Tamil Text: A Data Augmentation Approach with Transformer Model	YouTube	Línguas Indianas (Dravídicas)	MuRIL data augmentation	-	-	47%
Lande et al. [2023]. KaustubhSharedTask@LT-EDI 2023: Homophobia-Transphobia Detection in Social Media Comments with NLPAUG-driven Data Augmentation	YouTube	Inglês e Línguas Indianas (Dravídicas)	Word2Vec BiLSTM Data Augmentation	-	-	43%
Hegde et al. [2023]. MUCS@LT-EDI2023: Homophobic/Transphobic Content Detection in Social Media Text using mBERT	YouTube	Inglês, Espanhol e Línguas Indianas (Dravídicas)	mBERT Oversampling	-	-	Inglês: 91% Tamil: 91% Malayalam: 95% Hindi: 94% Spanish: 81%
Chakravarthi [2023]. Detection of homophobia and transphobia in YouTube comments	YouTube	Inglês e Línguas Indianas (Dravídicas)	LR BERT SVM BERT	-	-	Inglês (LR BERT): 47% Tamil (SVM BERT): 80% Tamil-Inglês (SVM BERT): 50%
Andrew [2023]. JudithJeyafreeda@LT-EDI: Using GPT model for recognition of Homophobia/Transphobia detection from social media	YouTube	Inglês, Espanhol e Línguas Indianas (Dravídicas)	GPT2	-	-	Inglês: 90% Tamil: 27% Malayalam: 25% Hindi: 02% Spanish: 00%
Anand Kumar [2023]. Multilingual Models for Sentiment and Abusive Language Detection for Dravidian Languages	YouTube	Inglês e Línguas Indianas (Dravídicas)	TF-IDF LSTM	-	-	Tamil: 35% Tulu: 51% Telugu: 64% Tamil-Inglês: 51%
Yadav et al. [2024]. dkit@LT-EDI-2024: Detecting Homophobia and Transphobia in Inglês Social Media Comments	YouTube	Inglês	BERT	-	-	49%
Wong and Durward [2024]. cantnlp@LT-EDI-2024: Automatic Detection of Anti-LGBTQ+ Hate Speech in Under-resourced Languages	YouTube	Inglês, Espanhol e Línguas Indianas (Dravídicas)	XLM-RoBERTa	-	-	Inglês: 32% Spanish: 50% Gujarati: 96% Hindi: 33% Kannada: 94% Malayalam: 78% Marathi: 43% Tamil: 56% Tulu: 45% Telugu: 97%
Sonali et al. [2024]. MUCS@LT-EDI-2024: Learning Approaches to Empower Homophobic/Transphobic Comment Identification	YouTube	Inglês, Espanhol e Línguas Indianas (Dravídicas)	Homo$_E nsemble$ Homo$_T L$ Homo$_p robfuse$ Homo$_F SL$	-	-	Telugu (Homo$_E nsemble$) : 95% Inglês (Homo$_T L$) : 49% Marathi (Homo$_T L$) : 53% Hindi (Homo$_T L$) : 45% Kannada (Homo$_T L$) : 94% Gujarati (Homo$_T L$) : 95% Tamil (Homo$_p robfuse$) : 86% Malayalam (Homo$_p robfuse$) : 87% Spanish (Homo$_p robfuse$) : 53% Tulu (Homo$_F SL$) : 62%

Tabela III.4: Continuação da Tabela Resumo do levantamento bibliográfico

Autor e obra	Base de dados	Língua	Modelo	Precisão	Revocação	F-measure
Sharma et al. [2024]. Abusive comment detection in Tamil using deep learning	YouTube	Línguas Indianas (Dravídicas)	LSTM RNN	-	-	Telugu-Inglês (LSTM): 65% Tamil-Inglês (LSTM): 26% Tamil (RNN): 22%
Shanmugavadivel et al. [2024]. KEC-AI-NLP@LT-EDI-2024:Homophobia and Transphobia Detection in Social Media Comments using Machine Learning	YouTube	Inglês	Random Forest (RF)	-	-	34%
Manukonda and Kodali [2024]. byteLLM@LT-EDI-2024: Homophobia/Transphobia Detection in Social Media Comments - Custom Subword Tokenization with Subword2Vec and BiLSTM	YouTube	Línguas Indianas (Dravídicas)	BiLSTM subword2Vec	-	-	Telugu: 96% Malayalam: 89% Kannada: 92% Tamil: 80%
Kumaresan et al. [2024]. Dataset for Identification of Homophobia and Transphobia for Telugu, Kannada, and Gujarati	YouTube	Línguas Indianas (Dravídicas)	BERT	-	-	Telugu: 96% Kannada: 92% Gujarati: 95%
Goswami et al. [2024]. MasonTigers@LT-EDI-2024: An Ensemble Approach Towards Detecting Homophobia and Transphobia in Social Media Comments	YouTube	Inglês, Espanhol e Línguas Indianas (Dravídicas)	XLM-R	-	-	Inglês: 32% Spanish: 55% Gujarati: 94% Hindu: 33% Kannada: 95% Malayalam: 51% Marata: 49% Tamil: 49% Tutlu: 42% Telugu: 97%
Arora et al. [2024]. MEnTr@LT-EDI-2024: Multilingual Ensemble of Transformer Models for Homophobia/Transphobia Detection	YouTube	Inglês, Espanhol e Línguas Indianas (Dravídicas)	XLM-RoBERTa	-	-	Tulu: 71% Spanish: 58% Telugu: 96% Gujarati: 96% Marata: 49% Tâmil: 75% Hindi: 33% Canará: 94% Inglês: 41% Malaiala: 74%
Allan et al. [2024]. Quartet@LT-EDI 2024: Support Vector Machine Based Approach For Homophobia/Transphobia Detection In Social Media Comments	YouTube	Línguas Indianas (Dravídicas)	SVM	-	-	Malayalam: 88% Telugu: 94% Inglês: 96% Marathi: 78% Kannada: 93% Tamil: 77% Gujarati: 94% Hindi: 47%

Tabela III.5: Continuação da Tabela Resumo do levantamento bibliográfico

Capítulo IV Metodologia

Esta seção detalha a Metodologia para Construção de Classificadores de Conteúdo Homofóbico (MCCCH), apresentando a sequência das etapas necessárias para atingir o objetivo principal e os objetivos específicos. O processo inicia-se com a criação do léxico homofóbico, seguida pela construção do dataset de mensagens do *X*. Em seguida, ocorre a rotulagem dos textos quanto ao conteúdo homofóbico e, por fim, a classificação das mensagens utilizando aprendizado de máquina.

A Figura IV.1 ilustra esse fluxo, no qual cada método processa os dados e gera um novo conjunto de dados como saída, servindo de entrada para a etapa seguinte. A única exceção ocorre na última etapa, cujo objetivo é a classificação final dos textos.

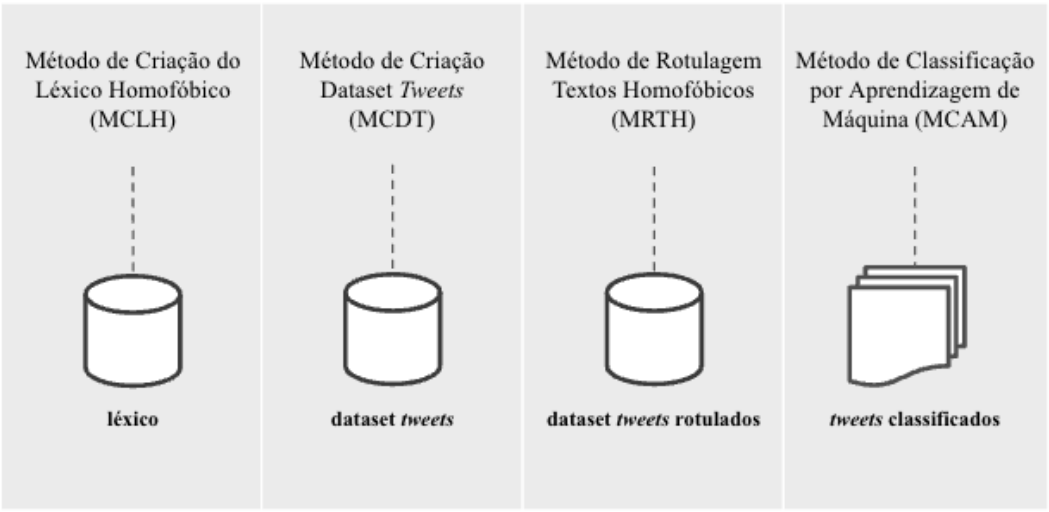


Figura IV.1: Metodologia para Construção de Classificadores de Conteúdo Homofóbico (MCCCH)

O Método de criação do léxico homofóbico (MCLH) é o passo referente ao processo de geração do léxico, o Método de criação do dataset tweets (MCDT) é a etapa de geração do dataset de *tweets*. O Método de Rotulagem de Textos Homofóbicos (MRTH) é o estágio de geração do dataset de *tweets* rotulados. Por fim, o Método de Classificação por Aprendizado de Máquina (MCAM) é o passo da classificação dos tweets quanto à homofobia.

IV.1 Método de criação do Léxico homofóbico (MCLH)

Uma limitação à classificação automática de textos homofóbicos é a incipiência de um léxico. Como descrito no Capítulo I, a plataforma *Hatebase* dispõe de apenas 28 termos ou expressões relacionadas à homofobia, no idioma português.

O MCLH visa romper com tal limitação, por meio da opinião de seres humanos, incrementando termos ou expressões homofóbicas ao léxico. Inicialmente, cada participante é apresentado ao Termo de Consentimento Livre e Esclarecido (TCLE). Após isso, pergunta-se para cada uma das pessoas se estas conhecem os termos ou expressões homofóbicas presentes no *Hatebase*, conforme Tabela IV.1,

Termos Homofóbicos
Baitola (Variante de Boiola)
Bambi
Bambis (Plural de Bambi)
Biba
Bicha
Bichinha (Variante de Bicha)
Bichona (Variante de Bicha)
Boiola
Boiolas
Desviada (Variante de Desviado)
Desviadas (Plural de Desviada)
Desviado
Desviados (Variante de Desviado)
Escorregar No Quiabo
Fufa
Fufas (Plural de Fufa)
Maricas
Mulherzinha
Mulherzinhas (Plural de Mulherzinha)
Paneleira
Paneleiras (Plural de Paneleira)
Paneleiro
Paneleiros (Plural de Paneleiro)
Rabeta
Rabetas (Plural de Rabeta)
Sapatão
Sapatões (Plural de Sapatão)
Viado

Tabela IV.1: Termos homofóbicos em português - fonte: Hatebase [2020]

O intuito de exibir a lista em questão, é fazer com que as pessoas reflitam sobre o que já se têm conhecimento *a priori*. Na sequência, ocorre a colaboração dos respondentes à criação do léxico homofóbico. Os participantes são perguntados se eles têm conhecimento de outros termos ou expressões não listados. Todo o processo é ilustrado, detalhadamente, por meio da Figura IV.2.

Os passos do fluxograma ilustrado na Figura IV.2 são:

1. Início: O processo se inicia.

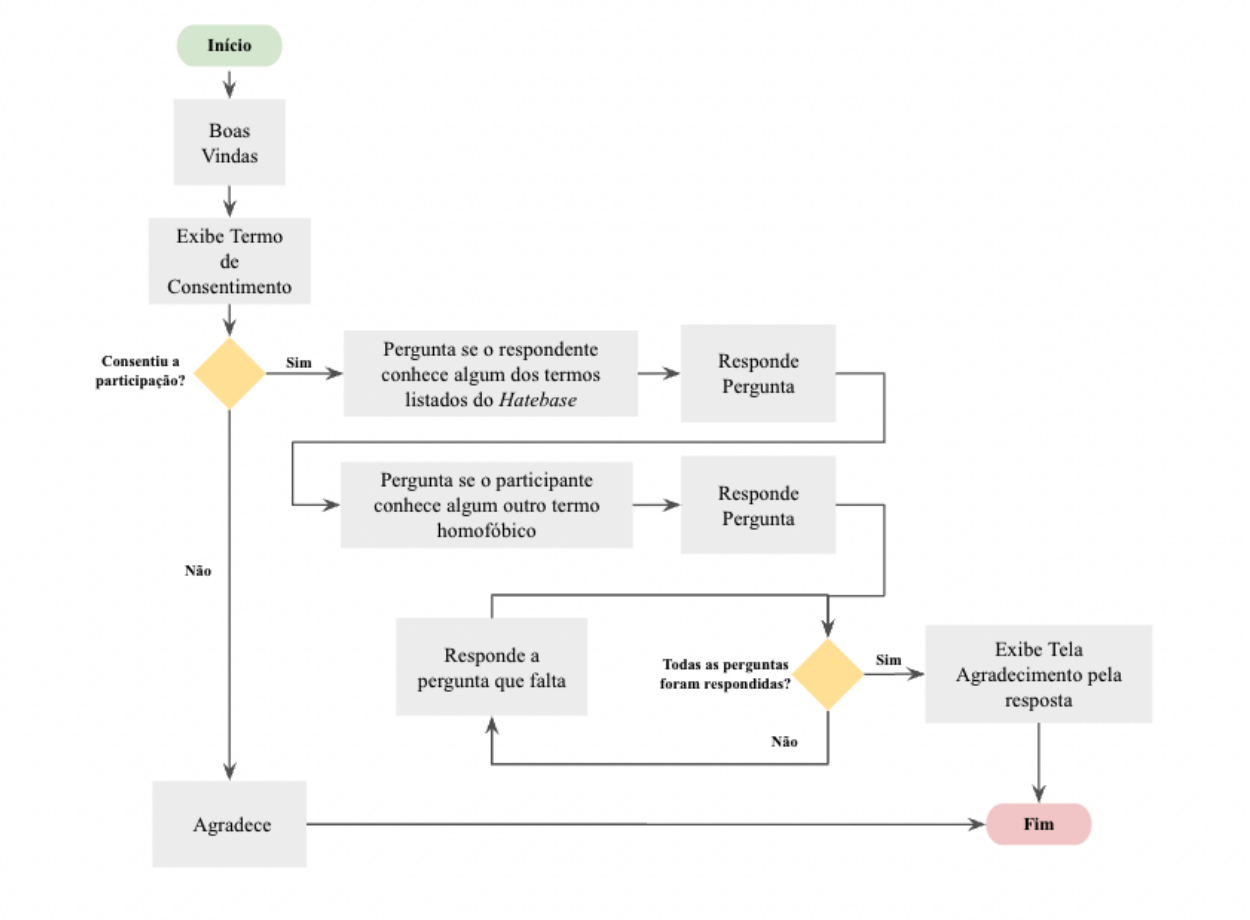


Figura IV.2: Contribuição humana ao léxico homofóbico

2. Boas-vindas: O participante recebe uma mensagem de boas-vindas.
3. Exibe Termo de Consentimento: O sistema apresenta o termo de consentimento ao participante.
4. Decisão – Consentiu a participação? Se "Não", o sistema agradece e encerra o processo. Se "Sim", o participante segue para as perguntas.
5. Pergunta se o participante conhece algum termo listado no *Hatebase*.
6. Responde a pergunta.
7. Pergunta se o participante conhece algum outro termo homofóbico.
8. Responde a pergunta.
9. Verifica se todas as perguntas foram respondidas. Se "Não", o sistema retorna para que o participante responda a pergunta que falta. Se "Sim", o sistema exibe a tela de agradecimento.
10. Exibe Tela de Agradecimento pela resposta.

11. Fim: O processo é concluído.

O formulário de pesquisa utilizado para coleta das informações foi o *Google Forms*. A plataforma é gratuita, sem limitação de respondentes, simples e, além disso, permite baixar os dados coletados e encerrar a pesquisa ao fim do período. A primeira pergunta refere-se aos termos e expressões conhecidos *a priori*, a resposta era obrigatória para "*Conheço*" e "*Não conheço*". A pergunta consiste no seguinte texto: "*Dos termos homofóbicos listados abaixo, marque os que você conhece e os que não conhece*", conforme a Figura IV.3.

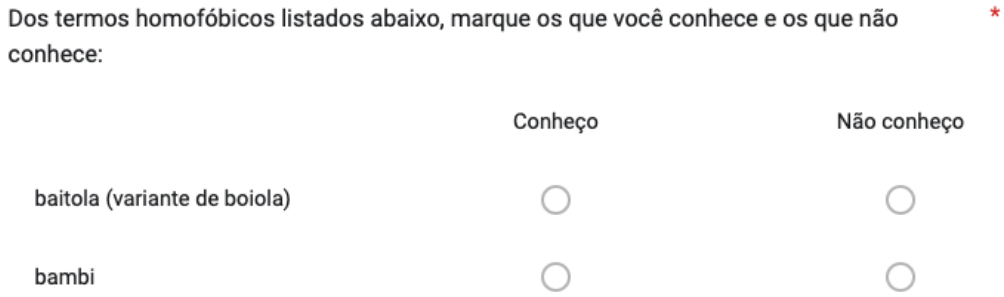


Figura IV.3: MCLH - Pergunta 1 - Exemplo de termos homofóbicos Hatebase (demais omitidos)

A segunda pergunta refere-se à contribuição incremental ao léxico homofóbico, a resposta é obrigatória e pergunta-se: "*Você conhece algum outro termo, expressão homofóbica? Escreva aqui, separado por vírgula*", conforme ilustrado na Figura IV.4.



Figura IV.4: MCLH - Léxico homofóbico - Pergunta 2

IV.2 Método de criação dataset *tweets* (MCDT)

O MCDT efetiva consultas ao *Twitter API V2*, recurso que permite acesso aos dados históricos públicos. Conforme relatado por Twitter [2020], por meio de acesso à API, os pesquisadores dispõem de um conjunto de dados mais precisos, completos e imparciais.

A pesquisa inicia-se com a construção da Base de Dados, representada por uma coleção de *tweets*. Esta etapa do processo prevê o acesso aos dados e, para isso, foi utilizada a *API* do *Twitter*, em `https://developer.twitter.com/en/support/twitter-api/v2`.

A consulta feita a *API* do *Twitter* utilizou como recurso computacional a linguagem de programação Python [1]. Além disso, duas bibliotecas foram tomadas como construção do algoritmo de busca, são elas: *tweepy*[2] e *pandas*[3].

A biblioteca *Tweepy* permite configurar a conexão com a *API* por meio das chaves de acesso, facilitando a extração de dados. Já a biblioteca *Pandas* possibilita a organização e estruturação do banco de dados, permitindo a incorporação dos textos extraídos durante a consulta.

O algoritmo de busca foi desenvolvido na plataforma *Google Colab*[4], permitindo que todo o processamento dos dados ocorra em um ambiente colaborativo e baseado em nuvem, garantindo maior acessibilidade e escalabilidade.

Os seguintes critérios foram estabelecidos na construção do algoritmo de busca:

- Consultas por mensagens no período de 01/01/2019 a 30/06/2022 semestralmente a fim de evitar bloqueio do *Twitter* por excesso de requisição;
- Buscas por textos em português e escritos no Brasil;
- Não consideração de retweets ou textos compartilhados, evitando duplicidade do mesmo conteúdo;
- Para evitar que os termos estivessem contidos no nome de algum perfil (i.e. viadoescroto), considerou-se buscas que não contivessem menções;
- Para cada termo ou expressão homofóbica desenvolvida na Seção IV.1, foi arbitrada uma amostra de 30 *tweets*.

Após a coleta dos *tweets*, os dados passam por um pré-processamento para garantir sua adequação à fase de rotulagem. Inicialmente, são removidos conteúdos não textuais, como URLs, emojis, imagens, hashtags e múltiplos espaços. Em seguida, os textos são agrupados para a eliminação de duplicatas. Por fim, registros com consultas vazias ou ausência de dados são desconsiderados.

IV.3 Método de rotulagem textos homofóbicos (MRTH)

Conforme discutido no Capítulo III, a escassez de *datasets* representa uma limitação que este estudo busca superar. Embora de Pelle and Moreira [2017] e Fortuna et al. [2019a] tenham feito contribuições relevantes, o número de textos rotulados como homofóbicos ainda é muito reduzido, destacando a necessidade de coletar mais amostras específicas para a língua portuguesa. O MRTH

[1] `https://www.python.org/`
[2] `https://www.tweepy.org/`
[3] `https://pandas.pydata.org/`
[4] `https://research.google.com/colaboratory/intl/pt-BR/faq.html`

visa submeter ao julgamento humano os textos coletados de mensagens extraídas do *Twitter*. Para tal, os seguintes rótulos são avaliados no método: “sim”, “não” e “não tem certeza”.

O formulário de pesquisa foi desenvolvido utilizando a plataforma *PCIBEX*, acessível em `https://farm.pcibex.net/`. A escolha dessa ferramenta se justifica por suas características únicas, como ser totalmente gratuita, de código aberto, colaborativa e projetada para a condução de experimentos *online*, garantindo flexibilidade e acessibilidade no desenho do estudo.

A Figura IV.5 apresenta um fluxograma detalhado do processo de interação do participante na pesquisa, contribuindo para a visualização e entendimento de todas as etapas do processo. Os passos do fluxograma ilustrado são:

1. Início do processo.
2. Exibição da mensagem de boas-vindas ao participante.
3. Apresentação do termo de consentimento TCLE.
4. Decisão: O participante consente em participar? Se a resposta for “Não”, o sistema agradece e o processo termina. Se a resposta for “Sim”, o participante segue para o sorteio das perguntas.
5. Sorteio de um grupo de perguntas.
6. Sorteio de uma pergunta dentro do grupo selecionado.
7. Exibição da pergunta ao participante.
8. O participante responde à pergunta.
9. Verificação: O participante já respondeu a 30 questões? Se “Não”, o fluxo retorna ao sorteio de uma nova pergunta. Se “Sim”, o sistema exibe uma tela de agradecimento pela resposta.
10. O processo é finalizado.

IV.4 Método de classificação de aprendizado de máquina (MCAM)

O MCAM utiliza o dataset de tweets rotulados quanto ao conteúdo homofóbico descrito na Seção IV.3. A tarefa de classificação de textos envolve a aplicação de algumas técnicas descritas no Capítulo II. Os modelos de classificação aplicados ao conjunto de dados são: (i) classificadores tradicionais, como SVM, KNN, DT e Naive Bayes; (ii) classificadores ensemble RF, GB e AB; (iii) redes neurais artificiais CNN e MLP; (iv) LLM, como BERT, RoBERTa e GPT. Os classificadores aplicados ao conjunto de dados são avaliados quanto às métricas descritas no Capítulo II (e.g., precisão, revocação e F1-measure).

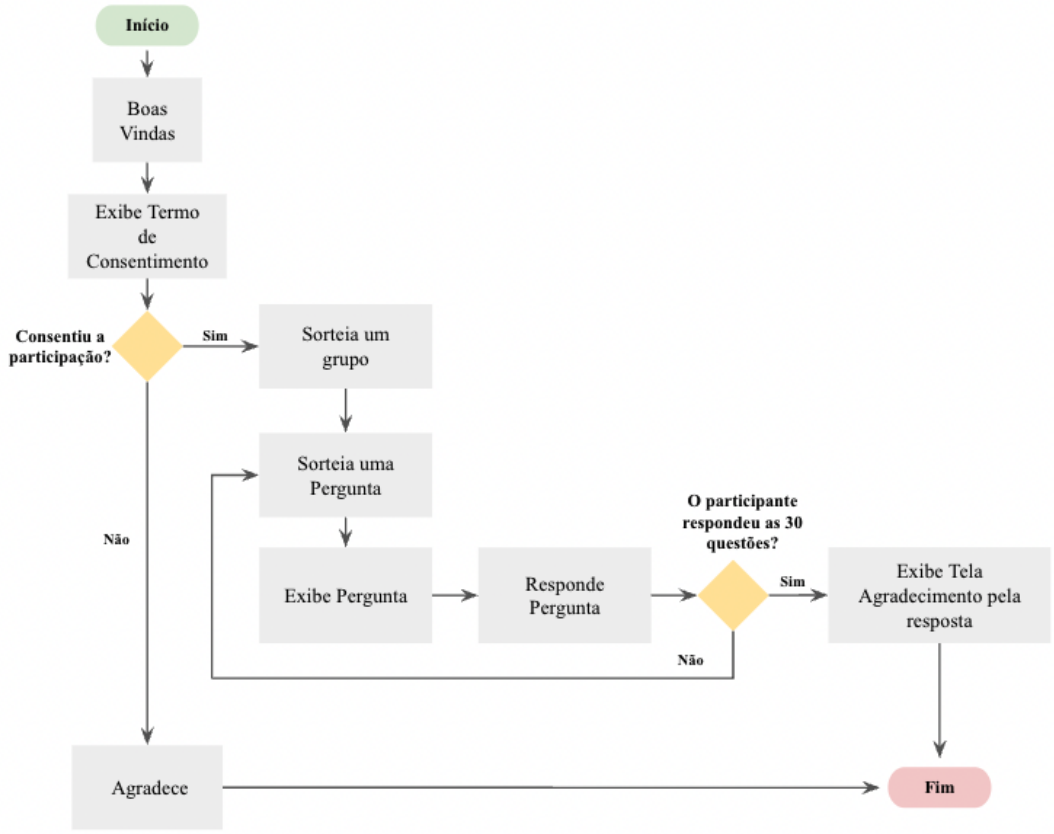


Figura IV.5: Método de rotulagem de mensagens textuais homofóbicas - MRMTH

O desenvolvimento desta etapa do processo foi realizado utilizando a linguagem de programação *Python*, por meio da plataforma *Google Colab*. Para a extração dos atributos, empregaram-se as bibliotecas *CountVectorizer* e *TfidfTransformer*, ambas integradas ao pacote *sklearn*[5]. Os métodos de *Deep Learning*, incluindo Redes Neurais MLP, CNN, e LLM BERT e RoBERTa, foram implementados com o suporte das bibliotecas *tensorflow*[6] e *transformers*.

Adicionalmente, a otimização dos parâmetros dos modelos foi realizada utilizando o *GridSearchCV*[7] para os métodos tradicionais e de comitê, enquanto o *Keras Tuner* foi empregado para ajustar as redes neurais e os modelos baseados em LLM. Por fim, a qualidade dos classificadores foi analisada por meio da matriz de confusão, validação cruzada e das métricas de precisão, revocação e F1-measure, assegurando uma avaliação detalhada e consistente dos resultados.

A implementação do modelo de classificação baseado no GPT utiliza a biblioteca OpenAI API. O modelo recebe instruções no formato de um *prompt* estruturado com uma descrição clara da tarefa. Assim como os demais classificadores, para verificar a qualidade da classificação, foram utilizadas a matriz de confusão e as métricas de precisão, revocação e F1-measure.

[5] `https://scikit-learn.org/stable/index.html`
[6] `https://www.tensorflow.org/?hl=pt-br`
[7] `https://scikit-learn.org/stable/modules/grid_search.html`

Capítulo V Avaliação Experimental

O Capítulo de Avaliação Experimental tem como objetivo detalhar os experimentos realizados, seguindo a metodologia apresentada no Capítulo IV para cada um dos métodos descritos. Inicialmente, são apresentados o MCLH e suas particularidades. Em seguida, abordam-se o MCDT e suas especificidades. Na sequência, discute-se o MRTH e suas características. Por fim, é exposto o MCAM, acompanhado da análise dos resultados obtidos para cada modelo de classificação, proporcionando uma visão completa da aplicação da metodologia.

V.1 MCLH

A incipiência de um léxico homofóbico faz com que seja necessário abranger termos e expressões que não foram contemplados na plataforma *Hatebase*. Conforme descrito no Capítulo IV, uma pesquisa de opinião com seres humanos foi proposta para resolver tal limitação.

A pesquisa foi disponibilizada por meio da url `https://forms.gle/R2pcz7x1RdpH8kii8`, no período entre os dias 01 de julho de 2022 e 10 de julho de 2022. O recrutamento dos participantes ocorreu utilizando as mídias sociais, como *Facebook*, *Whatsapp* e *Linkedin*, além de *e-mail* aos grupos de coordenadores do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - Cefet/RJ. Ao término do período de pesquisa, obteve-se um total de 487 participantes que responderam ao questionário. No entanto, devido às restrições estabelecidas pelo Termo de Consentimento Livre e Esclarecido (TCLE) e à aprovação do estudo pelo Comitê de Ética em Pesquisa, não foram coletados dados demográficos sobre os respondentes nesta etapa da pesquisa.

A análise dos dados coletados permitiu a construção da base necessária para o estudo. Para isso, os seguintes passos foram considerados:

- Separar os termos ou expressões identificados na pergunta 2, utilizando vírgulas como delimitadores;
- Contabilizar a frequência de ocorrência de cada termo extraído;
- Filtrar os termos que apresentarem mais de uma ocorrência, considerando sua relevância;
- Consolidar a base de dados final a partir dos termos filtrados.

No total, foram coletados 136 termos homofóbicos durante a pesquisa com os respondentes. Após a aplicação das etapas de análise, 54 termos ou expressões homofóbicas foram selecionados para compor o Léxico LEX-HOMOFENSA-PT, incluindo aqueles provenientes do *Hatebase*, conforme detalhado no Apêndice C. Esse processo permitiu um aumento de mais de 90% na quantidade de termos homofóbicos em relação à base original do *Hatebase*.

V.2 MCDT

O MCDT utiliza a base de léxico homofóbico desenvolvida na Seção V.1. Conforme a metodologia descrita no Capítulo IV, a coleta dos dados ocorreu pelo acesso a *API* do *X* com o uso da biblioteca *tweepy* da linguagem de programação *Python*.

O primeiro passo para essa tarefa de recuperação de textos foi o cadastro de acesso à plataforma *X* com perfil de *developer* (desenvolvedor) para ser possível conectar-se a *API*. Após a conexão, utilizamos a biblioteca *tweepy* em uma função definida no *Python* com as requisições à rede social para cada um dos termos homofóbicos, respeitando os critérios estabelecidos na metodologia. Ao final do processamento, o dataset de *tweets* foi construído a partir dos resultados obtidos das buscas, utilizando-se uma amostra de 30 textos para cada termo da base de léxico LEX-HOMOFENSA-PT, totalizando 1.620 *tweets* coletados TEXT-HOMOFENSA-PT.

Após a etapa de tratamento dos textos do dataset TEXT-HOMOFENSA-PT, é possível analisar informações relevantes para essa pesquisa. A primeira análise proposta é a descritiva em relação ao tamanho dos textos. Tais informações são fundamentais para compreender as principais estatísticas relacionadas aos textos recuperados e que, segundo Aggarwal [2018], são insumo para aplicação dos modelos de classificação. A Tabela V.1 dispõe das informações de total de textos, média, desvio padrão, mínimo, máximo e os quartis. Pode-se verificar que, em média, os textos do dataset possuem aproximadamente 19 termos.

Estatísticas	Valores
Média	18,81
Desvio Padrão	12,57
Mínimo	2
1º Quartil	9
Mediana	15
3º Quartil	25
Máximo	57

Tabela V.1: Análise Descritiva do Tamanho dos Textos em Palavras - TEXT-HOMOFENSA-PT

O boxplot apresentado na Figura V.1 fornece uma análise visual do tamanho dos textos do dataset. O gráfico ilustra a variação e dispersão dos tamanhos dos textos, sendo útil para identificar padrões ou anomalias na análise de dados textuais. A linha tracejada vermelha indica a média dos

tamanhos de texto, aproximadamente 18,81. A distribuição é inclinada positivamente, com um valor central (mediana) de 15, ligeiramente menor que o valor da média. Também é possível observar a presença de *outliers* à direita, indicando que existem textos com tamanhos significativamente maiores do que a maioria dos dados.

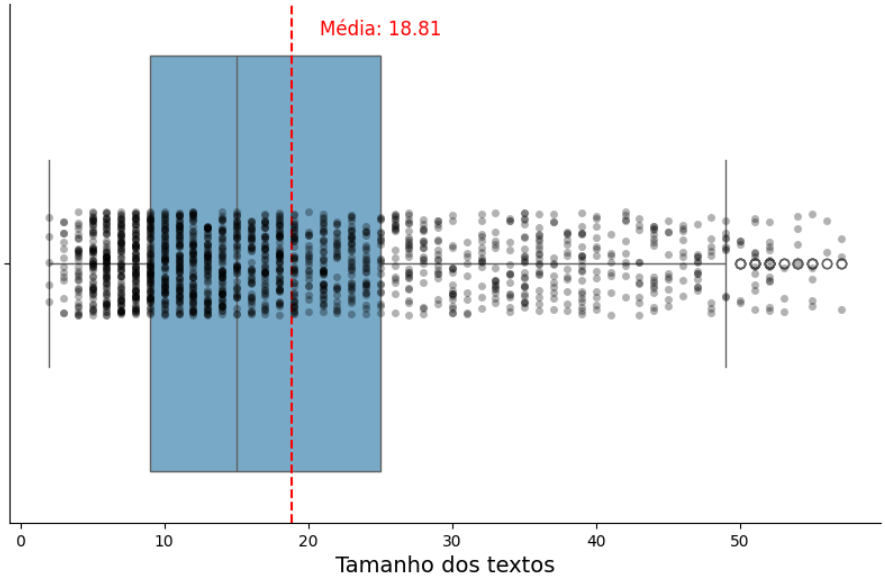


Figura V.1: Boxplot Tamanho dos Textos em Palavras - TEXT-HOMOFENSA-PT

A segunda análise tem como objetivo avaliar a frequência das palavras no dataset TEXT-HOMOFENSA-PT, identificando aquelas que aparecem com maior recorrência nos textos. A Figura V.2 apresenta os 20 termos mais frequentes, destacando que "vai", "cara" e "rosca" são os mais prevalentes, com frequências de 96%, 72% e 63%, respectivamente. Esses dados indicam que tais palavras estão presentes na maioria dos *tweets* analisados. Além disso, a Figura V.2 evidencia a presença de termos do léxico LEX-HOMOFENSA-PT, como esperado, já que os textos foram obtidos com base nesse léxico. Também é possível observar a ocorrência de palavras de baixo calão, o que pode contribuir para o desempenho do MRTH, uma vez que essas palavras frequentemente qualificam ou adjetivam o sujeito dos textos coletados.

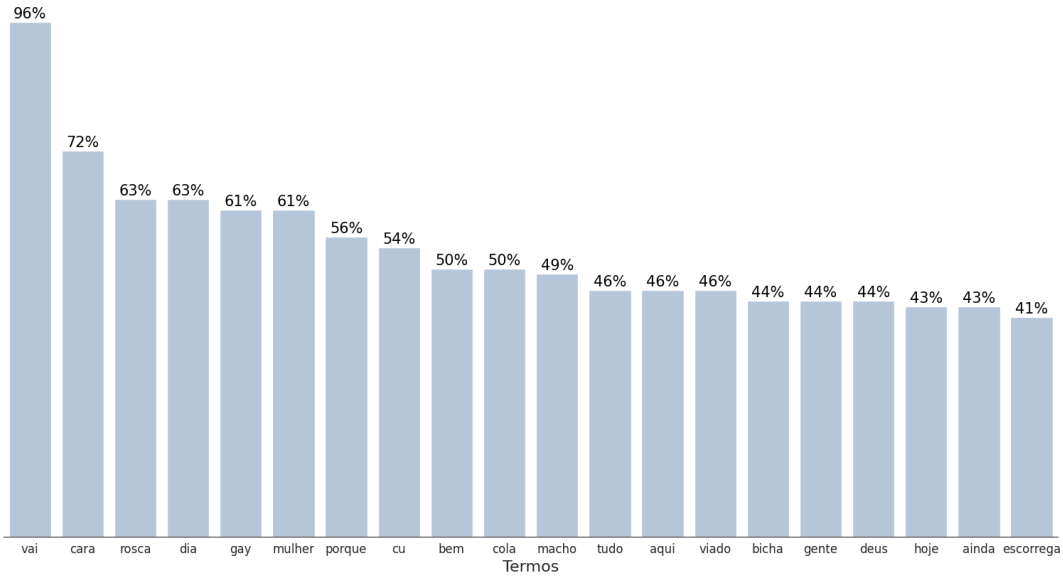


Figura V.2: Termos mais frequentes - TEXT-HOMOFENSA-PT

Uma nuvem de palavras é útil para avaliar, de maneira geral, a frequência de palavras extraídas dos *tweets*. A Figura V.3 representa as frequências relativas a cada palavra, onde o tamanho de cada palavra é a representação da sua presença nos textos; quanto maior, mais recorrentes em relação às demais. Como mostrado na Figura V.2, grande parcela das palavras mais frequentes está fortemente associada a termos homofóbicos, evidenciando a relação dos textos com os termos buscados.



Figura V.3: Termos mais frequentes

V.3 MRTH

A não existência de uma base de dados rotuladas para o discurso de ódio homofóbico é uma limitação a ser superada por meio do MRTH. Para a etapa do MRTH, um total de 1.620 textos foram coletados do X para cada um dos 54 termos ou expressões homofóbicas presentes no conjunto de dados do LEX-HOMOFENSA-PT, descrito na Seção V.2.

O questionário de coleta dos rótulos foi construído na plataforma *PCIBEX*, de acordo com

a metodologia proposta, e ficou disponível na url `https://farm.pcibex.net/p/ugwObt/` entre os dias 03 e 18 de julho de 2023. Para facilitar o processo e dividir o esforço dos julgadores por dia, foram enviados 14 questionários, com 120 textos para os primeiros 13 e 60 textos para o último. Essa medida se fez necessária para evitar que os participantes abandonassem os questionários sem finalizá-los; assim, foi possível diminuir o tempo diário e consolidar o total necessário.

Três respondentes (juízes) foram convidados a participar, de forma anônima, respeitando o Termo de Consentimento. O grupo de juízes foi composto por três indivíduos com perfis demográficos distintos: uma mulher heterossexual com mais de 60 anos, um homem heterossexual com idade entre 21 e 25 anos e um homem gay com idade entre 36 e 40 anos. Cada juiz se comprometeu a responder todas as 1.620 perguntas do questionário, de acordo com a divisão proposta nos questionários, entre os dias da pesquisa. Os disparos dos questionários ocorreram às 9h, por meio de envio de mensagem com a url da pesquisa na mídia social *Whatsapp*, onde os mesmos poderiam responder até às 21h, de acordo com o exemplo abaixo:

> "Olá, você está recebendo o Questionário 12, disponível em `https://farm.pcibex.net/p/MlyYYu/`, cuja validade expira hoje 15/07/23 às 21h.
>
> Reserve um momento tranquilo para responder.
> Ao todo, as perguntas levam menos de 5 minutos.
> Lembre-se de responder dentro do prazo, para que a pesquisa tenha validade.
>
> Qualquer dúvida, entrar em contato direto pelo Whatsapp.
> Obrigado por sua participação.
>
> Vinícius Soares dos Santos
> Mestrando em Ciência da Computação
> PPCIC - CEFET/RJ"

Os participantes julgaram cada *tweet* extraído, respondendo à pergunta: "Na sua opinião, o conteúdo acima é homofóbico?", com as opções de resposta "sim", "não" e "não tem certeza", conforme exemplo da Figura V.4. Ao responderem cada pergunta, os usuários eram destinados à próxima até a conclusão do questionário. Quando os julgadores finalizavam todo o questionário, a plataforma *PCIBEX* consolidava todas as respostas em um dataset com o nome do questionário e uma chave numérica de identificação do respondente, anonimamente.

Ao término da pesquisa, foi realizada uma validação para garantir que todos os julgadores responderam às 1.620 perguntas. Após a confirmação dessa etapa, o dataset de *tweets* rotulados foi construído a partir do agrupamento dos 14 datasets gerados pela plataforma *PCIBEX*. O rótulo final de cada *tweet* foi definido com base na convergência da maioria dos julgadores, isto é, quando dois ou mais julgamentos eram concordantes. Nos casos de empate, o rótulo foi atribuído como "Não

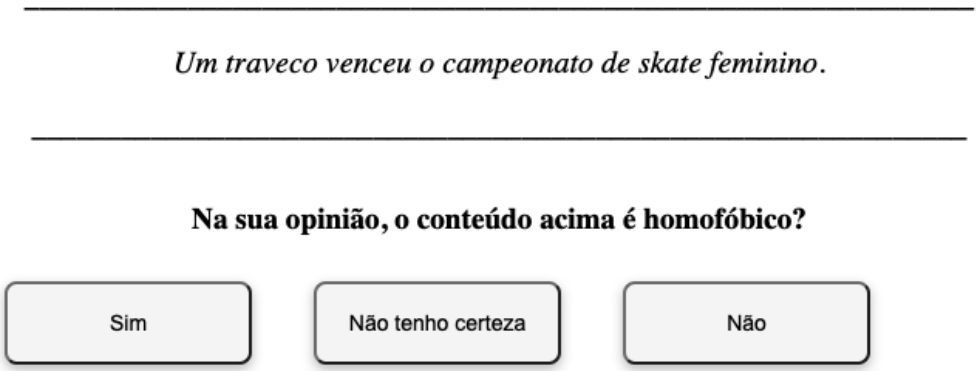


Figura V.4: Exemplo de mensagem para rotulação

tenho certeza", indicando uma discordância sobre a presença de homofobia no texto. O resultado final, com os 1.620 textos rotulados pelos três julgadores, está apresentado na Figura V.5.

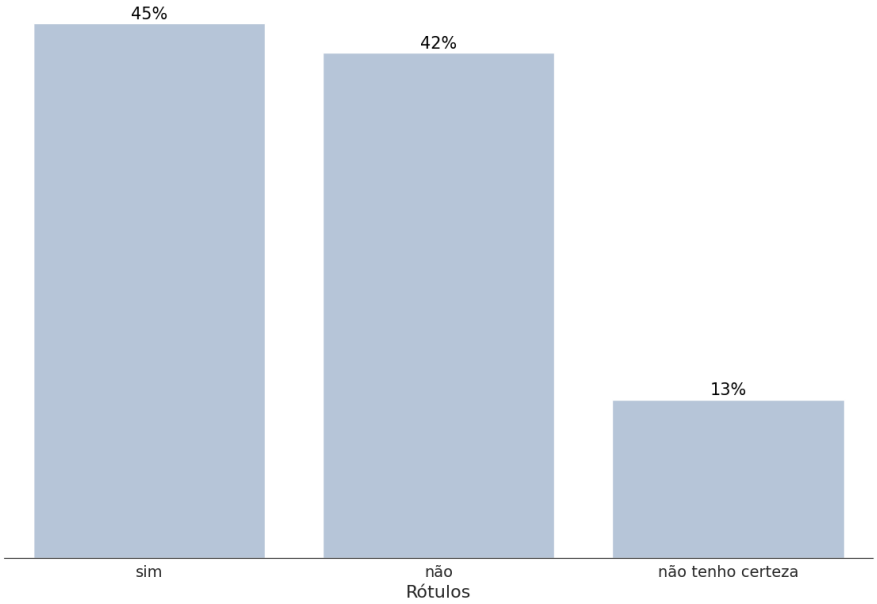


Figura V.5: Frequência dos rótulos - TEXT-HOMOFENSA-PT

Observa-se pela Figura V.5 que, aproximadamente 87% dos *tweets* rotulados puderam ser considerados no julgamento, sendo o rótulo "sim" para 45% (ou 723 tweets) e o rótulo "não" para 42% (ou 683 tweets). Os outros 13% foram considerados como dúvida no julgamento, por não ter certeza sobre o julgamento ou pelo empate entre os três respondentes. Como uma medida estatística de concordância entre múltiplos avaliadores (juízes) em tarefas de categorização, foi realizado o cálculo do coeficiente *Kappa de Fleiss*, obtendo o resultado de 0,2654. O valor do coeficiente de Kappa de Fleiss indica uma baixa concordância entre os avaliadores, demonstrando a dificuldade do julgamento dos textos recuperados. O rótulo da convergência do julgamento foi associado ao dataset

TEXT-HOMOFENSA-PT.

Uma análise boxplot para o tamanho dos textos e o rótulo dos julgadores foi construída a fim de avaliar a distribuição dos dados. A Figura V.6 indica que a mediana da distribuição do tamanho dos textos para o rótulo "sim" é menor que os demais rótulos, embora contendo a maior presença de *outliers*. A mediana do tamanho dos textos para o rótulo "não" é maior que os demais, o que pode sugerir que quanto maior o tamanho dos textos, maiores as chances de ele não ser um discurso de ódio homofóbico. Entretanto, com a visualização lado a lado dos boxplots da Figura V.6, é possível verificar que as distribuições dos tamanhos dos textos são muito próximas para cada rótulo, não sendo possível afirmar que, estatisticamente, há uma diferença significativa em relação ao tamanho dos textos para os rótulos em questão.

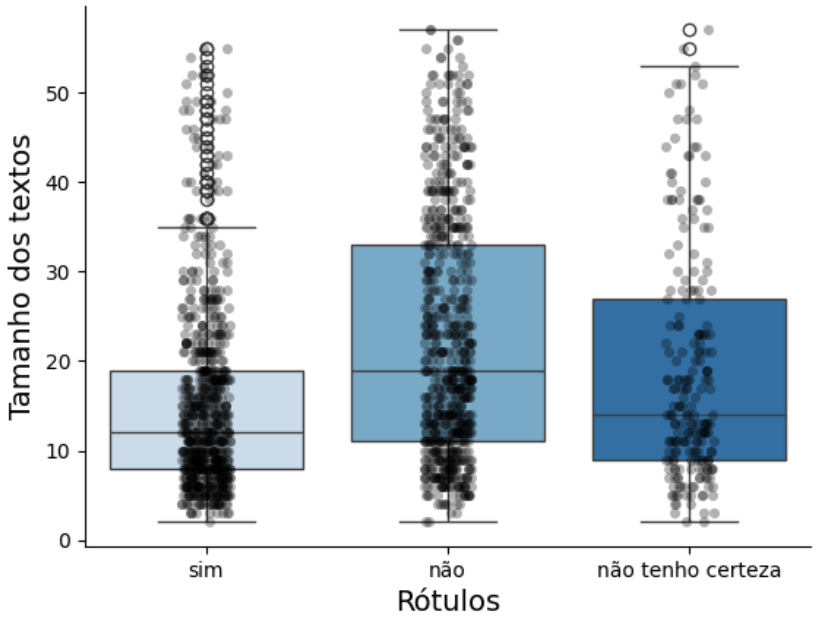


Figura V.6: Boxplot rótulos por tamanho das frases

V.4 MCAM

O MCAM tem como objetivo realizar a classificação de textos por meio da aplicação de modelos de aprendizado de máquina, conforme detalhado no Capítulo IV. Para isso, foram utilizados os rótulos "sim" e "não" do dataset TEXT-HOMOFENSA-PT, que contém 723 (51%) e 683 (49%) *tweets*, respectivamente, apresentando um equilíbrio entre as classes. A tarefa de classificação de textos envolve a aplicação de técnicas descritas no Capítulo II. Os modelos de classificação aplicados ao dataset TEXT-HOMOFENSA-PT incluem: (i) classificadores tradicionais, como SVM, KNN, DT e Naive Bayes; (ii) classificadores *ensemble*, como RF, GB e AB; (iii) redes neurais artificiais, como CNN e MLP; e (iv) modelos de linguagem de grande escala (LLM), como BERT e RoBERTa.

A representação vetorial utilizada é baseada no TF-IDF, aplicada após a transformação inicial com BoW usando a função *CountVectorizer*. Os dados do dataset TEXT-HOMOFENSA-PT foram separados em conjuntos de treinamento e teste, 70% e 30%, respectivamente. O particionamento do conjunto de dados foi feito utilizando a função *train_ test_ split* da biblioteca *scikit-learn*.

Inicialmente, são ajustados os melhores hiperparâmetros dos modelos, utilizando a função GridSearchCV, considerando 5 partições. Em seguida, por meio da validação cruzada, calcula-se a média e o desvio padrão das métricas de desempenho F-measure, precisão e revocação no conjunto de treinamento (70%), garantindo uma análise mais precisa da capacidade de generalização dos modelos. Após essa etapa, o modelo treinado, ajustado com os melhores hiperparâmetros e avaliado por métricas de desempenho no conjunto de treinamento, é aplicado ao conjunto de teste (30%). Esse procedimento permite estimar as classes para novos dados e verificar a capacidade de generalização do modelo.

A avaliação no conjunto de teste utiliza as mesmas métricas adotadas no treinamento, possibilitando uma comparação direta entre as fases de desenvolvimento e validação do modelo. A matriz de confusão é gerada para ilustrar a relação entre as classes reais e previstas. Nesse contexto, a classe "0" corresponde a "não" e a classe "1" corresponde a "sim" para o rótulo de conteúdo homofóbico.

V.4.1 Análise de Modelos

Logistic Regression (LR)

O ajuste dos parâmetros do modelo de classificação LR apresentou os melhores parâmetros para $C = 1$, inverso da força de regularização (positivo), e solver = "liblinear", algoritmo a ser utilizado no problema de otimização, onde para conjuntos de dados pequenos, segundo Pedregosa et al. [2011] o "liblinear" é uma boa escolha. Estes parâmetros foram avaliados a partir do GridSearchCV com: $C \in \{0.1, 1, 10\}$.

Após aplicar a validação cruzada para o conjunto de treinamento, obteve-se uma precisão de 86,79%, com um desvio padrão de 3,15%. Para a revocação, o valor foi de 86,43%, com um desvio padrão de 3,12%. O F-measure foi de 86,50%, com um desvio padrão de 3,13%. A Figura V.7 ilustra a matriz de confusão para os dados de teste, após a validação cruzada e os hiperparâmetros ajustados.

O desempenho apresentado é considerado muito bom para o conjunto de teste, com uma precisão de 85,37%, indicando a capacidade do modelo de identificar corretamente as instâncias positivas. A revocação foi de 85,30%, refletindo a boa cobertura do modelo, enquanto o F-measure de 85,30% sugere um bom equilíbrio entre precisão e revocação. Em relação ao trabalho de de Pelle and Moreira [2017], embora para o conjunto de dados distintos, verifica-se que a avaliação do modelo LR para a métrica F1-measure foi superior ao modelo SVM abordado pelo autor, 82,0%, para comentários

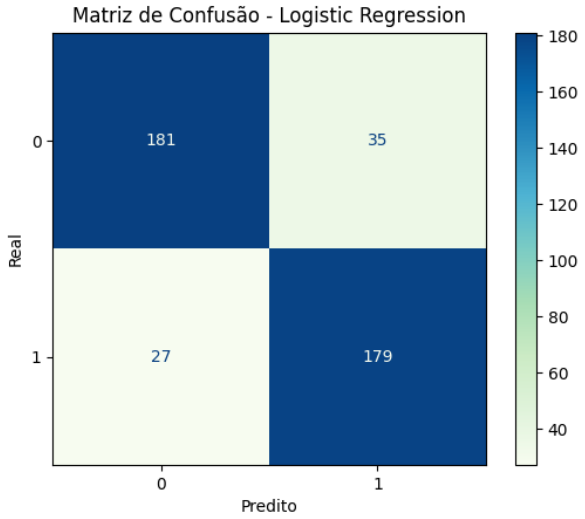


Figura V.7: Matriz de confusão - Classificador LR

extraídos do portal de notícias G1.

Naive Bayes (NB)

O modelo NB utilizou como parâmetro de suavização o valor de $var_smoothing = 1e-06$, um dos valores avaliados no GridSearchCV, que incluiu $var_smoothing \in \{1e-9, 1e-8, 1e-7, 1e-6\}$. Durante a validação cruzada para o conjunto de treinamento, o modelo alcançou uma precisão de 77,02%, com desvio padrão de 1,60%. A revocação foi de 76,70%, e o desvio padrão de 1,65%. O F1-measure ficou em 76,76%, com um desvio padrão de 1,65%. A Figura V.8 ilustra a matriz de confusão para os dados de teste, com os parâmetros ajustados.

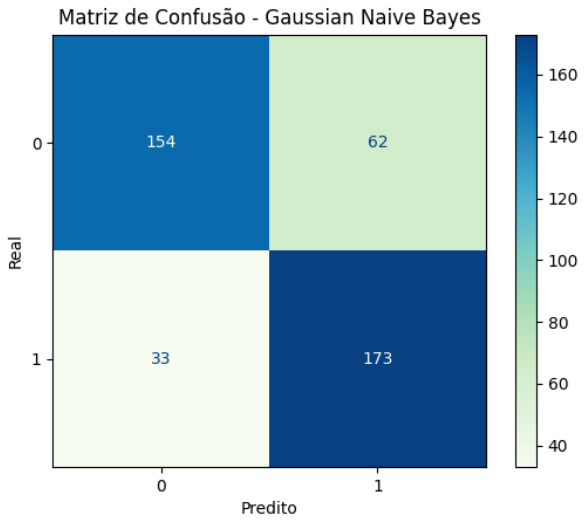


Figura V.8: Matriz de confusão - Classificador NB

O desempenho foi considerado consistente para o conjunto de teste, com uma precisão de 78,09%

e uma revocação de 77,49%. O F-measure foi de 77,42%, refletindo um desempenho equilibrado, porém com desempenho inferior ao da LR.

Support Vector Machine (SVM)

O modelo Support Vector Machine obteve os melhores parâmetros $C = 0.5$, $gamma = scale$ e $kernel =' rbf'$, avaliados a partir dos parâmetros no GridSearchCV: $C \in \{0.1, 0.5, 1.0, 1.5, 2.0\}$, $kernel \in \{'linear','poly','rbf','sigmoid'\}$ e $gamma \in \{'scale','auto'\}$.

A validação cruzada para o conjunto de treinamento revelou uma precisão de 86,74%, com desvio padrão de 2,15%. A revocação foi de 86,50%, com desvio padrão de 2,17%. O F-measure foi de 86,52%, com um desvio padrão de 2,17%. A Figura V.9 ilustra a matriz de confusão para os dados de teste, com os parâmetros ajustados.

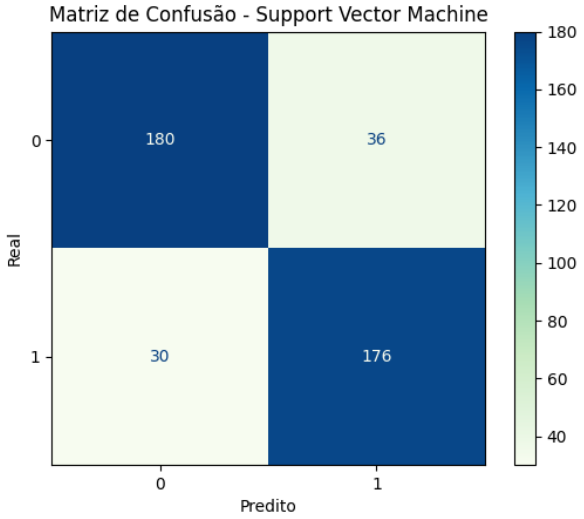


Figura V.9: Matriz de confusão - Classificador SVM

O desempenho do modelo foi considerado bom para o conjunto de teste, muito próximo ao LR, com uma precisão de 84,40% e uma revocação de 84,36%, o que resultou em um F-measure de 84,36%.

K-Nearest Neighbors (KNN)

O modelo k-nearest neighbors obteve melhores parâmetros $algorithm =$ "auto", $metric =$ "euclidean", $neighbors = 15$, $p = 1$ e $weight =$ "distance". Os parâmetros avaliados no GridSearchCV foram $n_neighbors \in \{3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31\}$, $weights \in \{$"uniform", "distance"$\}$, $metric \in \{$"euclidean", "manhattan", "minkowski"$\}$, $p \in \{1, 2\}$ e $algorithm \in \{$"auto", "ball_tree", "kd_tree", "brute"$\}$.

Durante a validação cruzada, para os dados de treinamento, o modelo obteve uma precisão de 84,89%, com um desvio padrão de 2,32%. A revocação foi de 84,03%, com desvio padrão de 2,47%.

O F-measure foi de 84,16%, com um desvio padrão de 2,48%. A Figura V.10 ilustra a matriz de confusão para os dados de teste, com os parâmetros ajustados.

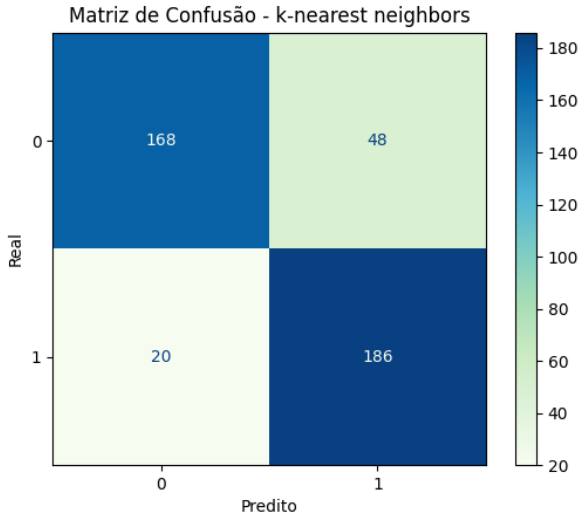


Figura V.10: Matriz de confusão - Classificador KNN

Para o conjunto de teste, o modelo apresentou uma precisão de 84,54%. A revocação foi de 83,89%, com um F1-measure de 83,84%. O desempenho do modelo também foi muito próximo ao modelo LR, com diferença mais significativa na revocação.

Decision Tree (DT)

A DT obteve os melhores parâmetros $max_leaf_nodes = 32$, $min_samples_split = 2$. Os parâmetros avaliados no GridSearchCV foram: $max_leaf_nodes \in \{2, 3, \ldots, 100\}$ e $min_samples_split \in \{2, 3, 4\}$.

A validação cruzada, para os dados de treinamento, indicou uma precisão de 85,86%, com um desvio padrão de 2,20%. A revocação foi de 85,83%, e o desvio padrão de 2,29%. O F-measure foi de 85,74%, com um desvio padrão de 2,26%. A Figura V.11 ilustra a matriz de confusão para os dados de teste, com os hiperparâmetros ajustados.

O resultado obtido das métricas de avaliação para o conjunto de teste foi uma precisão de 84,17% e uma revocação de 84,12%, com um F-measure de 84,11%.

Random Forest (RF)

O modelo Random Forest obteve os melhores parâmetros $criterion =$ "entropy", $max_depth = 8$, $max_features =$ "sqrt", $n_estimators = 500$. Os parâmetros avaliados no GridSearchCV foram $criterion =$ "entropy", $max_depth \in \{4,5,6,7,8\}$, $max_features \in$ {"sqrt", "log2"} e $n_estimators \in \{200, 500\}$.

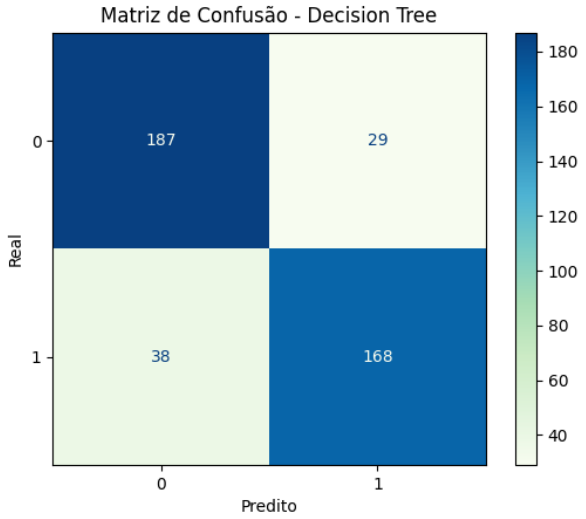


Figura V.11: Matriz de confusão - Classificador DT

Durante a validação cruzada, para os dados de treinamento, o modelo apresentou uma precisão de 85,28%, com um desvio padrão de 1,56%. A revocação foi de 82,04%, e o desvio padrão foi de 1,78%. O F-measure foi de 82,14%, com um desvio padrão de 1,88%. A Figura V.12 ilustra a matriz de confusão para os dados de teste, com os hiperparâmetros ajustados.

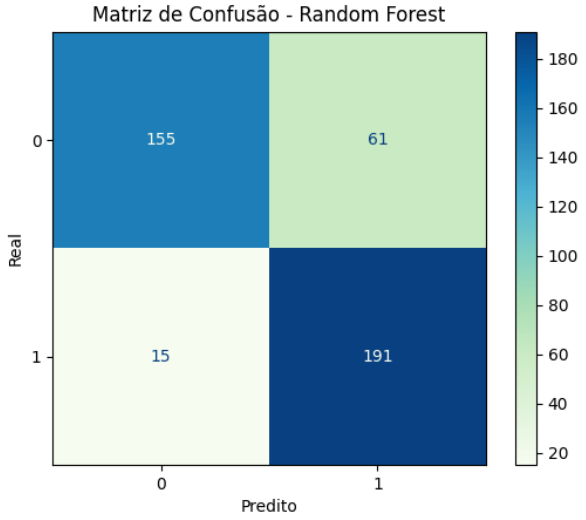


Figura V.12: Matriz de confusão - Classificador RF

O desempenho foi sólido para o conjunto de teste, com uma precisão de 83,67%, uma revocação de 81,99%, e um F-measure de 81,82%.

AdaBoost (AB)

O modelo AB obteve os melhores parâmetros $learning_rate = 1$ e $n_estimators = 200$, avaliados a partir dos parâmetros no GridSearchCV: $learning_rate \in \{0.01, 0.1, 1\}$ e $n_estimators \in$

$\{50, 100, 200\}$.

Na validação cruzada, para os dados de treinamento, a precisão foi de 83,48%, com um desvio padrão de 3,21%. A revocação foi de 80,73%, com desvio padrão de 3,46%. O F-measure foi de 80,50%, com um desvio padrão de 4,06%. A Figura V.13 ilustra a matriz de confusão para os dados de teste, com os parâmetros ajustados.

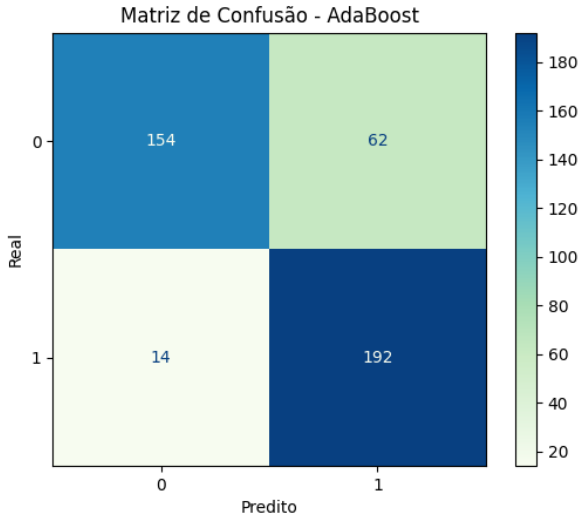


Figura V.13: Matriz de confusão - Classificador AB

Para o conjunto de teste, o modelo apresentou uma precisão de 83,82% e uma revocação de 81,99%, com um F-measure de 81,80%.

Gradient Boosting (GB)

O modelo GB obteve os melhores parâmetros $learning_rate = 0.5$, $max_depth = 5$, $n_estimators = 200$, avaliados a partir dos parâmetros no GridSearchCV: $learning_rate \in \{0.01, 0.1, 0.5\}$, $max_depth \in \{3, 5, 7\}$, e $n_estimators \in \{50, 100, 200\}$.

A validação cruzada, para os dados de treinamento, revelou uma precisão de 86,61%, com um desvio padrão de 2,20%. A revocação foi de 86,39%, e o desvio padrão foi de 2,18%. O F-measure foi de 86,42%, com um desvio padrão de 2,20%. A Figura V.14 ilustra a matriz de confusão para os dados de teste, com os parâmetros ajustados.

Para o conjunto de teste, o modelo apresentou uma precisão de 82,75%. A revocação foi de 82,70%, e o F-measure foi de 82,70%, indicando um bom equilíbrio entre precisão e revocação.

Redes Neurais Multi-Layer Perceptron (MLP TF-IDF)

O modelo MLP obteve os melhores hiperparâmetros configurados como: $units_layer1 = 192$, $dropout_layer1 = 0.2$, $units_layer2 = 32$, $dropout_layer2 = 0.2$, $learning_rate = 0.001$, $tuner/e$-

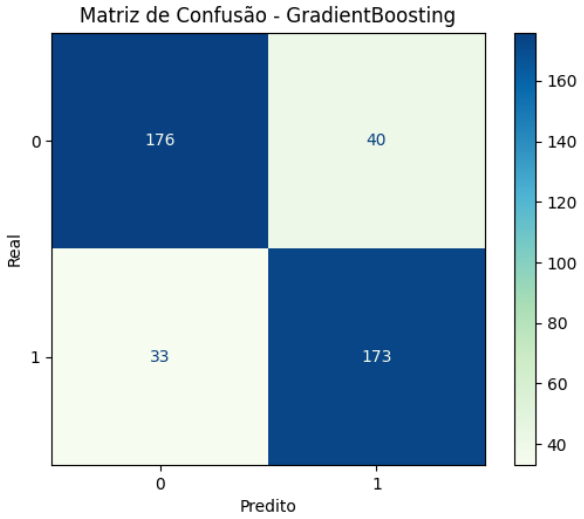


Figura V.14: Matriz de confusão - Classificador GB

pochs = 4, *tuner/initial_ epoch* = 2, *tuner/bracket* = 2, *tuner/round* = 1, *tuner/trial_ id* = '0005'. Para o ajuste desses hiperparâmetros, foi utilizada a biblioteca *Keras Tuner*, que possibilitou uma busca eficiente pelos melhores valores.

A arquitetura da rede neural foi configurada com duas camadas densas (*Dense*) e camadas de regularização (*Dropout*). A primeira camada densa teve o número de unidades ajustado como hiperparâmetro, variando entre 64 e 256 em incrementos de 64, com a função de ativação *ReLU*. A camada de *Dropout* subsequente teve sua taxa de abandono (*dropout_ rate*) ajustada entre 0,1 e 0,5 em incrementos de 0,1. A segunda camada densa variou entre 32 e 128 unidades, ajustadas em incrementos de 32, com ativação *ReLU*, seguida por outra camada de *Dropout* com os mesmos limites. A camada de saída foi composta por um único neurônio com ativação *sigmoid* para classificação binária. O otimizador escolhido foi o *Adam*, com a taxa de aprendizado (*learning rate*) ajustada entre os valores *1e-3*, *1e-4* e *1e-5*, e o modelo foi compilado com a função de perda *binary_ crossentropy* e a métrica de acurácia para avaliação de desempenho.

A validação cruzada, para os dados de treinamento, apresentou uma precisão média de 83,91% e um desvio padrão de 4,16%. A revocação média foi de 84,38%, com desvio padrão de 4,05%, enquanto o F-measure alcançou uma média de 84,05% com desvio padrão de 3,06%. A Figura V.15 ilustra a matriz de confusão para os dados de teste, com os parâmetros ajustados.

O modelo apresentou para o conjunto de teste uma precisão de 83,02%, uma revocação de 81,11% e um F-measure de 82,05%, demonstrando um bom equilíbrio entre precisão e revocação.

Redes Neurais Convolucionais (CNN)

Para o ajuste dos hiperparâmetros do modelo CNN, também foi utilizada a biblioteca *Keras Tuner*, permitindo uma busca eficiente pelos melhores valores. O modelo foi configurado com uma

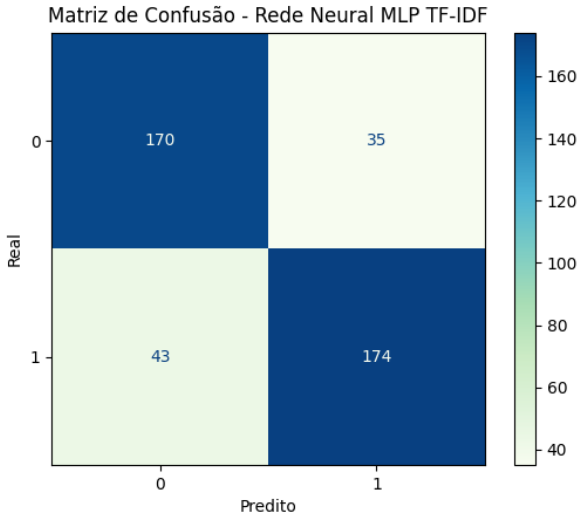


Figura V.15: Matriz de confusão - Classificador MLP TF-IDF

camada de *Embedding* inicial, onde a dimensão de entrada foi definida como 10.000 e a dimensão de saída ajustada como hiperparâmetro, variando entre 32 e 128 em incrementos de 32. Em seguida, foi adicionada uma camada *Conv1D*, na qual o número de filtros variou entre 32 e 128, em incrementos de 32, e o tamanho do *kernel* foi ajustado entre 3, 5 e 7. A ativação utilizada nesta camada foi *ReLU*. Após a convolução, uma camada de *GlobalMaxPooling1D* foi inserida para redução dimensional.

Na etapa de saída, uma camada densa foi configurada com o número de unidades variando entre 32 e 128, ajustadas em incrementos de 32, e ativação *ReLU*, seguida por uma camada de *Dropout* cuja taxa foi ajustada entre 0,1 e 0,5 em incrementos de 0,1. A camada de saída final consistiu em um único neurônio com ativação *sigmoid* para classificação binária. O modelo foi compilado com o otimizador *Adam*, a função de perda *binary_ crossentropy* e a métrica de acurácia para avaliação de desempenho.

Após o ajuste dos hiperparâmetros, os melhores valores obtidos foram: dimensão dos *embeddings* igual a 64, número de filtros igual a 96, tamanho do *kernel* de 3, número de unidades densas igual a 96 e taxa de *dropout* igual a 0,4.

Na validação cruzada, para os dados de treinamento, os resultados indicaram uma precisão média de 84,38% com um desvio padrão de 3,79%, uma revocação média de 84,39% com desvio padrão de 3,09%, e um F-measure médio de 84,32% com desvio padrão de 2,63%. A Figura V.16 ilustra a matriz de confusão dos resultados do modelo para os conjuntos de teste.

Para o conjunto de teste, o modelo apresentou uma precisão de 85,79%, revocação de 77,88% e F-measure de 81,64%, demonstrando uma pequena diferença entre precisão e revocação.

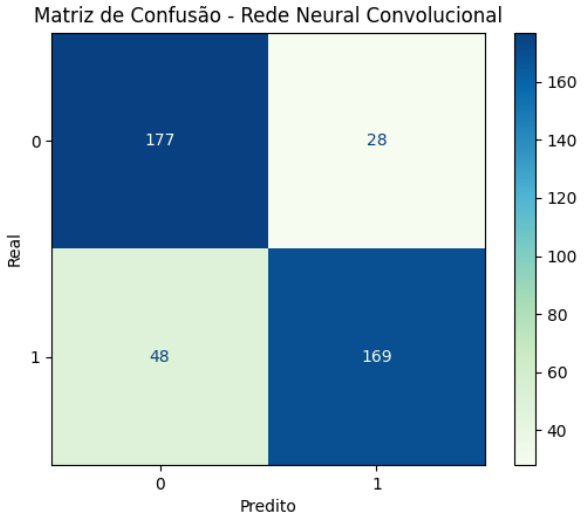


Figura V.16: Matriz de confusão - Classificador CNN

Bidirectional Encoder Representations from Transformers (BERT)

Para o ajuste e treinamento do modelo baseado em BERT, foi utilizada a biblioteca *Transformers* em conjunto com o *Keras Tuner* para buscar os melhores hiperparâmetros. A arquitetura do modelo foi configurada utilizando a entrada limitada a um comprimento máximo de 128 *tokens* (MAX_LENGTH). As entradas do modelo incluíram as IDs dos *tokens* ($input_ids$) e a máscara de atenção ($attention_mask$). A camada *Lambda* encapsulou o modelo BERT, extraindo o estado oculto da última camada (768 dimensões) referente ao primeiro *token* (*[CLS]*), que foi utilizada como a representação do texto.

A camada de saída foi composta por dois neurônios com ativação *softmax*, permitindo a classificação binária do conteúdo. O modelo foi compilado utilizando o otimizador *Adam* com uma taxa de aprendizado ajustável ($learning_rate$), a função de perda *SparseCategoricalCrossentropy* e a métrica de acurácia. Após o ajuste dos hiperparâmetros, os melhores valores encontrados foram: $learning_rate = 3 \times 10^{-5}$, $batch_size = 32$ e $epochs = 5$. Esses resultados refletem o desempenho otimizado do modelo, considerando o balanceamento entre tempo de treinamento e capacidade de generalização.

Na validação cruzada, para os dados de treinamento, os resultados foram inconsistentes, com uma precisão média de 49,24% e desvio padrão de 2,85%, uma revocação média de 50,12% com desvio padrão elevado de 24,80%, e um F-measure médio de 47,13% com desvio padrão de 12,37%. Esses resultados sugerem que sua capacidade de generalização ainda é limitada, possivelmente devido ao tamanho do conjunto de dados ou à complexidade do modelo em relação aos dados disponíveis.

A Figura V.17 ilustra a matriz de confusão para os dados de teste, com os hiperparâmetros ajustados.

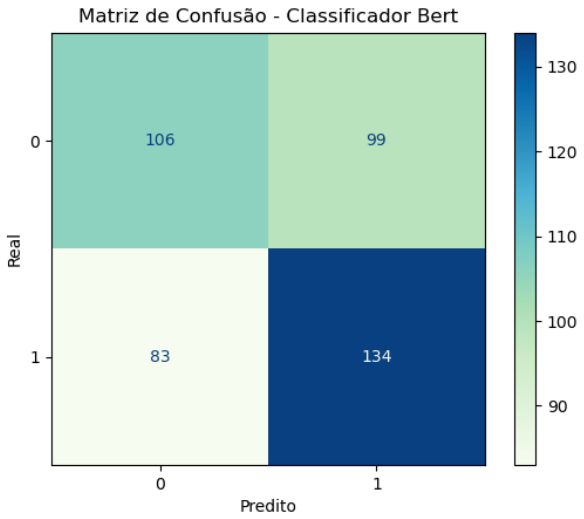


Figura V.17: Matriz de confusão - Classificador BERT

Para os dados de teste, o modelo BERT apresentou uma precisão de 57,51%, uma revocação de 61,75% e um F-measure de 59,56%, indicando um desempenho razoável, com a revocação ligeiramente superior à precisão.

Robustly Optimized BERT Pretraining Approach (RoBERTa)

Para o ajuste e treinamento do modelo baseado em RoBERTa, foi utilizada a biblioteca *Transformers* em conjunto com o *Keras Tuner* para a busca dos melhores hiperparâmetros. O comprimento máximo das entradas foi limitado a 128 *tokens* (MAX_LENGTH). As entradas do modelo incluíram IDs dos *tokens* ($input_ids$) e máscaras de atenção ($attention_mask$). A arquitetura contou com uma camada *Lambda* que encapsulou o modelo RoBERTa, extraindo o estado oculto da última camada referente ao primeiro *token* (*[CLS]*), utilizado como representação vetorial do texto.

A camada de saída foi composta por dois neurônios com ativação *softmax*, adequada para a classificação binária. O modelo foi compilado com o otimizador *Adam*, utilizando uma taxa de aprendizado ajustável ($learning_rate$) como hiperparâmetro, a função de perda *SparseCategoricalCrossentropy* e a métrica de acurácia. Após a otimização, os melhores hiperparâmetros encontrados foram: $learning_rate = 1 \times 10^{-5}$, $batch_size = 32$ e $epochs = 5$. Esses valores demonstraram um bom equilíbrio entre desempenho e capacidade de generalização do modelo, refletindo a robustez do RoBERTa em tarefas de classificação de texto, mesmo em conjuntos de dados limitados.

Os resultados da validação cruzada revelaram maior variabilidade, com uma precisão média de 68,55% e um desvio padrão de 3,08%, enquanto a revocação média foi de 42,75%, com um desvio padrão mais elevado de 6,77%. O F-measure médio foi de 52,21%, com desvio padrão de 4,52%, evidenciando limitações na capacidade de generalização do modelo em partições diferentes do conjunto de dados de treinamento.

A Figura V.18 ilustra a matriz de confusão para os dados de teste, com os hiperparâmetros ajustados.

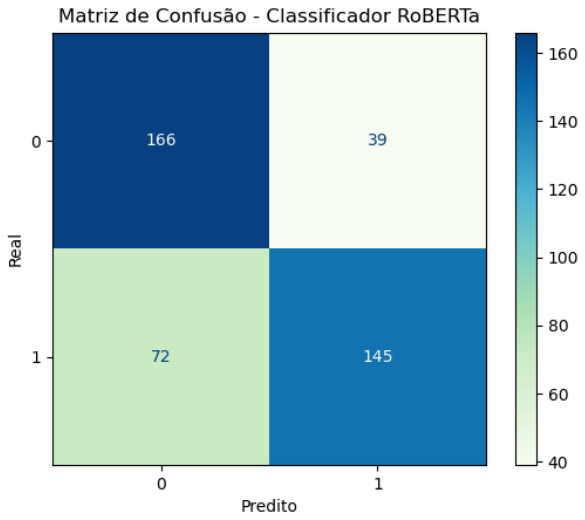


Figura V.18: Matriz de confusão - Classificador RoBERTa

O modelo RoBERTa apresentou para o conjunto de teste uma precisão de 78,80%, uma revocação de 66,82% e um F-measure de 72,31%, indicando um desempenho razoável com um bom equilíbrio entre precisão e revocação nos dados de teste.

Ao comparar os resultados dos modelos BERT e RoBERTa, observa-se que o RoBERTa apresentou um desempenho superior em termos de precisão, revocação e F-measure nos dados de teste. No entanto, na validação cruzada, apesar da superioridade do RoBERTa em precisão e F-measure, ambos os modelos apresentaram limitações em sua capacidade de generalização, o que pode ser atribuído ao tamanho do conjunto de dados e à complexidade inerente das arquiteturas LLM.

Generative Pre-trained Transformer (GPT)

A tarefa de classificação com o algoritmo GPT, na versão GPT-3.5-turbo, foi aplicada individualmente a cada texto do conjunto de dados TEXT-HOMOFENSA-PT por meio de um prompt estruturado, no qual o modelo foi instruído a determinar se a frase apresenta discurso homofóbico. O prompt utilizado foi:

```
A seguinte frase pode ser considerada discurso homofóbico?
Frase: "{text}"
Responda apenas com "sim" se for homofóbico e "não" se não for homofóbico.
```

Além disso, o modelo foi contextualizado previamente com a seguinte instrução: “Você é um especialista em detecção de discurso de ódio homofóbico.”

A saída gerada pelo GPT-3.5-turbo foi processada para padronizar as respostas, convertendo "sim" e "não" em valores numéricos, sendo 1 para discurso homofóbico e 0 para não homofóbico. Dessa forma, foi possível comparar as previsões do modelo com os rótulos reais e calcular métricas de desempenho. A Figura V.19 ilustra a matriz de confusão para o retorno dos valores preditos e valores reais.

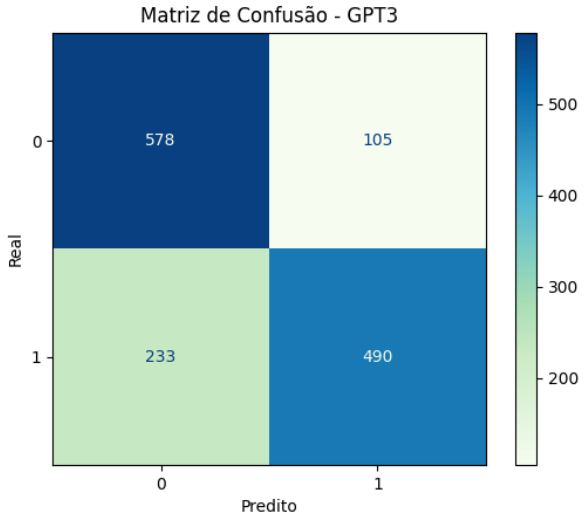


Figura V.19: Matriz de confusão - Classificador GPT

A avaliação do modelo foi realizada, obtendo as métricas de precisão em 77,00%, revocação em 76,00% e F-measure em 76,00%. A utilização do GPT-3.5-turbo como classificador permite a análise do discurso homofóbico sem a necessidade de treinamento supervisionado tradicional, o que reduz a dependência de bases de dados rotuladas extensas. No entanto, essa abordagem também apresenta desafios, uma vez que o desempenho do modelo depende diretamente da clareza do prompt e da consistência das respostas geradas.

V.4.2 Resumo dos classificadores

Ao término da aplicação dos modelos de classificação ao dataset TEXT-HOMOFENSA-PT de dados extraídos da rede social *X*, pode-se resumir as métricas de avaliação dos mesmos por meio da Tabela V.2. O modelo de LR obteve o melhor desempenho geral em termos de F-measure, alcançando um valor de 85,31%. Quando comparado com o trabalho de de Pelle and Moreira [2017], verifica-se que a avaliação do modelo LR para a métrica F1-measure foi superior ao modelo SVM abordado pelo autor, 82,00%.

Esse desempenho pode ser atribuído à simplicidade e eficiência do modelo, aliado à escolha otimizada dos hiperparâmetros, como o uso de C=1 e o solver liblinear, que são bem adequados para conjuntos de dados balanceados. Além disso, os classificadores tradicionais, como SVM com F1-score de 84,36%, KNN com F1-score de 83,84% e DT com F1-score de 84,11%, demonstraram bom

desempenho, próximos ao LR. Os hiperparâmetros bem ajustados e a ausência de desbalanceamento no conjunto de dados contribuíram para seus resultados.

Por outro lado, os modelos baseados em redes neurais e modelos de linguagem pré-treinados apresentaram F-measures inferiores, com BERT alcançando 59,56% e RoBERTa 72,32%. Esse desempenho pode ser atribuído a limitações relacionadas ao tamanho do dataset, uma vez que modelos de grande escala (LLM), como BERT e RoBERTa, geralmente requerem grandes volumes de dados para atingir seu pleno potencial. Outra limitação relevante é a escolha dos hiperparâmetros: embora tenham sido ajustados com o auxílio da biblioteca *Keras Tuner*, é possível que não tenham sido totalmente otimizados devido a restrições de tempo de treinamento e recursos computacionais (CPU). As redes neurais MLP, com F-measure de 82,05%, e CNN, com F-measure de 81,64%, apresentaram desempenho razoável; no entanto, ambas são mais suscetíveis a problemas de *overfitting*, especialmente em datasets menores.

Classificador	Precisão	Revocação	F-measure
LR	85,37%	85,31%	85,31%
NB	78,09%	77,49%	77,42%
SVM	84,40%	84,36%	84,36%
KNN	84,54%	83,89%	83,84%
DT	84,17%	84,12%	84,11%
RF	83,67%	81,99%	81,82%
AB	83,82%	81,99%	81,80%
GB	82,75%	82,70%	82,70%
MLP	83,02%	81,11%	82,05%
CNN	85,79%	77,88%	81,64%
BERT	57,51%	61,75%	59,56%
RoBERTa	78,80%	66,82%	72,32%
GPT	77,00%	76,00%	76,00%

Tabela V.2: Tabela resumo dos resultados modelos de classificação

Capítulo VI Conclusões

O discurso de ódio é a manifestação de incitação à discriminação racial, sexual, social ou religiosa [Meyer-Pflug, 2008]. Para Costa and Nardi [2015], o termo Homofobia é utilizado para conceituar o preconceito e discriminação contra pessoas cuja orientação sexual seja diferente da heterossexual. Os autores citam, no contexto histórico do século XX, a interpretação dominante da psicologia e da psiquiatria a respeito da homossexualidade corroborou para o preconceito contra homossexuais. Ainda segundo eles, o termo "homossexualismo" foi considerado por muito tempo no manual de diagnóstico de doenças psiquiátricas. Por fim, os autores abordam a homofobia como um preconceito contra a diversidade sexual.

O STF do Brasil, em 13 de junho de 2019, julgou como crime de racismo casos de agressões contra pessoas de orientação sexual LGBT [UOL, 2020]. Na ação denominada "*Homofobia e omissão legislativa*", lista-se o julgamento e as alegações da suprema corte. Dentre os trechos da determinação, destaca-se o enquadramento de atos homofóbicos e transfóbicos como formas contemporâneas de racismo [Brasil, 2019].

Os Objetivos de Desenvolvimento Sustentável (ODS) da Agenda 2030 da ONU destacam a importância de combater todas as formas de discriminação, incluindo o discurso de ódio e a homofobia, como parte do compromisso global com a igualdade e a inclusão [Organização das Nações Unidas, 2015]. A ODS 10, em sua meta 10.2, enfatiza a promoção da inclusão social, econômica e política de todos, sem distinção de raça, gênero, orientação sexual ou qualquer outra condição. Além disso, a meta 10.3 reforça a necessidade de eliminar práticas discriminatórias, o que inclui comportamentos que promovam discursos de ódio.

O objetivo geral da pesquisa foi desenvolver um método de identificação do discurso homofóbico em mensagens do *X*. O interesse do estudo relaciona-se com conteúdos textuais, apresentando conceitos e trabalhos relacionados a métodos de classificação automática do discurso de ódio, essencialmente o discurso homofóbico. Para alcançar o objetivo principal, teve-se como objetivos específicos:

- Criação de um léxico de termos homofóbicos
- Criação de uma base de dados de textos, produzidos por meio da extração da informação dos *tweets* da rede social em questão

- Rotulagem dos textos a partir do julgamento humano para as classes “sim”, “não” e “não tem certeza”.

- Aplicação de modelos de classificação ao conjunto de tweets rotulados, para: (i) classificadores tradicionais, como SVM, KNN, DT e Naive Bayes; (ii) classificadores ensemble, como Random Forest, Adaptive Boosting e Gradient Boosting; (iii) redes neurais artificiais, como CNN e MLP; (iv) LLM, como BERT, RoBERTa e GPT.

A Metodologia proposta MCCCH foi dividida em etapas do processo. O MCLH é o passo no processo de geração do Léxico, o MCDT é a etapa de geração do dataset de *tweets*. O MRTH é o estágio de geração do dataset de *tweets* rotulados. Por fim, o MCAM é o passo da classificação dos tweets quanto à homofobia.

O dataset de *tweets* foi construído a partir da etapa MCLH, utilizando 54 Termos Homofóbicos do Léxico LEX-HOMOFENSA-PT. A coleta dos *tweets* totalizou 1.620 textos, sendo 30 amostras para cada um dos Termos Homofóbicos do Léxico. A etapa MRTH foi desenvolvida a partir de 3 julgadores voluntários, cuja ação foi classificar cada texto com os rótulos “sim”, “não” e “talvez”. Após esta etapa, definiu-se o rótulo final de cada *tweet* coletado a partir do julgamento da maioria dos voluntários. Sempre que dois ou mais julgadores convergiam, a classe de convergência era atribuída ao texto, formando o dataset TEXT-HOMOFENSA-PT.

O próximo passo da Metodologia contou com a filtragem dos rótulos “sim”, “não” para a tarefa de classificação de textos MCAM. No final, 1.406 textos foram selecionados distribuídos entre os rótulos “sim” com 723 *tweets* (51%) e “não” com 683 *tweets* (49%). Para a Tarefa de Classificação de Textos, utilizou-se os modelos de Machine Learning e Deep Learning abordados na II. Os classificadores estão divididos em: (i) classificadores tradicionais, como SVM, KNN, DT e Naive Bayes; (ii) classificadores ensemble, como Random Forest, Adaptive Boosting e Gradient Boosting; (iii) redes neurais artificiais, como CNN e MLP; (iv) LLM, como BERT, RoBERTa e GPT.

Ao término da aplicação dos modelos de classificação ao dataset TEXT-HOMOFENSA-PT de dados extraídos da rede social *X*, pode-se concluir que o modelo de LR obteve o melhor desempenho geral em termos de F-measure, alcançando um valor de 85,31%. Em relação ao trabalho de de Pelle and Moreira [2017], embora para dados distintos, verifica-se que a avaliação do modelo LR para a métrica F1-measure foi superior ao modelo SVM abordado pelo autor, 82,0% para o conjunto de textos extraídos de comentários do portal de notícias G1.

Esse desempenho pode ser atribuído à simplicidade e eficiência do modelo, aliado à escolha otimizada dos hiperparâmetros, como o uso de C=1 e o solver liblinear, que são bem adequados para conjuntos de dados balanceados. Além disso, os classificadores tradicionais, como SVM com F1-score de 84,36%, KNN com F1-score de 83,84% e DT com F1-score de 84,11%, demonstraram bom

desempenho, próximos ao LR. Os hiperparâmetros bem ajustados e a ausência de desbalanceamento no conjunto de dados contribuíram para seus resultados.

Por outro lado, os modelos baseados em redes neurais e modelos de linguagem pré-treinados apresentaram F-measures inferiores, com BERT alcançando 59,56% e RoBERTa 72,32%. Esse desempenho pode ser atribuído a limitações relacionadas ao tamanho do dataset, uma vez que modelos de grande escala (LLM), como BERT e RoBERTa, geralmente requerem grandes volumes de dados para atingir seu pleno potencial. Outra limitação relevante é a escolha dos hiperparâmetros; embora tenham sido ajustados com o auxílio da biblioteca *Keras Tuner*, é possível que não tenham sido totalmente otimizados devido a restrições de tempo de treinamento e recursos computacionais (CPU). As redes neurais MLP, com F-measure de 82,05%, e CNN, com F-measure de 81,64%, apresentaram desempenho razoável; no entanto, ambas são mais suscetíveis a problemas de *overfitting*, especialmente em datasets menores.

Durante o desenvolvimento da pesquisa, algumas limitações foram observadas:

- A MCLH contou com a participação espontânea de pessoas elencando Termos Homofóbicos. A língua é um organismo vivo e por isso, constantemente, novos termos surgem na Internet para insultar pessoas LGBTQIAP+. A necessidade de atualização frequente do Léxico Homofóbico contribui para alcançar novos padrões de discursos de ódio.

- A MRTH contou com a participação de 3 julgadores voluntários. Essa etapa foi considerada uma das mais desafiadoras, pois exigiu um alto nível de dedicação, com os voluntários destinando várias horas à tarefa de rotulagem.

- Estima-se que a inclusão de um número maior de julgadores poderia aumentar o indicador *Kappa de Fleiss*, melhorando a consistência das avaliações.

- Uma amostra maior de textos associados a cada termo do Léxico Homofóbico, superior aos 30 textos utilizados nesta pesquisa, poderia contribuir para uma análise mais robusta e representativa.

- Os modelos baseados em Redes Neurais e LLM tiveram limitações em relação ao seu desempenho, atribuídos a quantidade de textos disponíveis, tempo de processamento e recursos computacionais.

- O classificador baseado no GPT possui limitação de desempenho associado diretamente da qualidade do *prompt* fornecido, onde variações na formulação do texto de entrada podem impactar significativamente a resposta gerada.

- O uso do GPT-3.5 Turbo requer acesso à API da OpenAI, o que implica custos financeiros para a requisição.

VI.1 Trabalhos Futuros

- Expansão do Léxico Homofóbico: ampliação do léxico LEX-HOMOFENSA-PT para incorporar novos termos e expressões que constantemente surgem na Internet como formas de insulto direcionado à população LGBTQIAP+.

- Generalização da Criação de Termos do Léxico: generalização do processo de criação do léxico LEX-HOMOFENSA-PT, incluindo a aplicação de técnicas automatizadas de análise de textos e mineração de dados para identificar novos termos ofensivos. Essa abordagem pode facilitar a expansão contínua do léxico e reduzir a dependência exclusiva de contribuições humanas.

- Aumento do Conjunto de Dados: criação de uma base de dados com um número significativamente maior de textos associados a cada termo do léxico, superando a limitação de 30 textos por termo utilizada nesta pesquisa. Um volume maior de dados contribuirá para o treinamento de modelos mais robustos, especialmente os baseados em Redes Neurais e Modelos de Linguagem de Grande Escala (LLM).

- Generalização para outras Redes Sociais: adaptar a metodologia para que possa ser aplicada a diferentes plataformas de redes sociais, ampliando o escopo de análise para além do *X* (antigo Twitter). Essa generalização possibilitaria identificar padrões de discurso de ódio em contextos variados, considerando as especificidades de cada rede social.

- Ampliação do número de Julgadores: no processo de rotulagem (MRTH), um maior número de julgadores voluntários poderia melhorar a consistência das classificações, refletida em indicadores como o *Kappa de Fleiss*. Métodos automatizados para rotulagem inicial, validados posteriormente por humanos, também poderim ser explorados para reduzir a carga de trabalho manual.

- Otimização de modelos de classificação: Para os modelos de classificação, seria interessante avaliar arquiteturas mais avançadas ou adaptar LLM pré-treinados para contextos específicos. Melhorias poderiam ser alcançadas por meio da otimização de hiperparâmetros e estratégias para mitigar problemas de *overfitting*.

- Comparação entre o GPT-3.5 Turbo e versões mais avançadas, como GPT-4 e futuros modelos da OpenAI, para verificar se há melhora nas métricas de desempenho do modelo, como precisão, revocação e F1-measure.

- Integração de Multimodalidade: investigar a integração de diferentes modalidades, como texto e imagem, para a detecção de discursos de ódio poderia aumentar o alcance e a precisão dos

métodos desenvolvidos. Essa abordagem poderia oferecer uma visão mais ampla e detalhada sobre os padrões de discriminação em redes sociais.

- Escrita de artigo científico sobre o método desenvolvido nessa dissertação, destacando os desafios enfrentados e os aprendizados obtidos ao longo da pesquisa.

Referências

Aggarwal, C. C. Machine learning for text, by charu c. aggarwal, new york. 28:541–543, 2018.

Akhtar, S., Basile, V., and Patti, V. A new measure of polarization in the annotation of hate speech. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11946 LNAI:588 – 603, 2019.

Allan, S. H., Sivakumar, S., Rohan, R., Jayaguptha, N., and Thenmozhi, D. Quartet@lt-edi 2024: Support vector machine based approach for homophobia/transphobia detection in social media comments. Association for Computational Linguistics (ACL), 2024.

Anand Kumar, M. Multilingual models for sentiment and abusive language detection for dravidian languages. Incoma Ltd, 2023.

Andrew, J. J. Judithjeyafreeda@lt-edi: Using gpt model for recognition of homophobia/transphobia detection from social media. page 78 – 82, 2023.

Arora, A., Mattoo, A., Chaudhary, D., Gorton, I., and Kumar, B. Mentr@lt-edi-2024: Multilingual ensemble of transformer models for homophobia/transphobia detection. Association for Computational Linguistics (ACL), 2024.

Ashraf, N., Taha, M., Taha, A., and Nayel, H. Nayel @lt-edi-acl2022: Homophobia/transphobia detection for equality, diversity, and inclusion using svm. page 287 – 290, 2022.

Balaji, M. J. and Chinmaya, H. A study on sentimental analysis, homophobia-transphobia detection for dravidian languages. volume 3395, page 138 – 146, 2022.

Bhandari, V. and Goyal, P. bitsa_nlp@LT-EDI-ACL2022: Leveraging pretrained language models for detecting homophobia and transphobia in social media comments. In Chakravarthi, B. R., Bharathi, B., McCrae, J. P., Zarrouk, M., Bali, K., and Buitelaar, P., editors, *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 149–154, Dublin, Ireland. Association for Computational Linguistics, 2022.

Bharathi, B. and Varsha, J. Ssncse nlp@ tamilnlp-acl2022: Transformer based approach for detection of abusive comment for tamil language. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 158–164, 2022.

Brasil. *Direito Constitucional - Ação Direta de Inconstitucionalidade por omissão*. Supremo Tribunal Federal, 2019.

Burnap, P. and Williams, M. L. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet*, 7(2):223–242, 2015.

Chakravarthi, B. R. Detection of homophobia and transphobia in youtube comments. *International Journal of Data Science and Analytics*, 2023.

Chakravarthi, B. R., Hande, A., Ponnusamy, R., Kumaresan, P. K., and Priyadharshini, R. How can we detect homophobia and transphobia? experiments in a multilingual code-mixed setting for social media governance. *International Journal of Information Management Data Insights*, 2(2), 2022.

Chakravarthi, B. R., Priyadharshini, R., Ponnusamy, R., Kumaresan, P. K., Sampath, K., Thenmozhi, D., Thangasamy, S., Nallathambi, R., and McCrae, J. P. Dataset for identification of homophobia and transophobia in multilingual youtube comments, 2021.

Chanda, S., Mishra, A., and Pal, S. Sentiment analysis and homophobia detection of code-mixed dravidian languages leveraging pre-trained model and word-level language tag. volume 3395, page 184 – 195, 2022.

Cohn, J., Kuntz, A., and Birnbaum, L. Attitudebuzz: Using social media data to localize complex attitudes. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ASONAM '15, page 1569–1570, New York, NY, USA. Association for Computing Machinery, 2015.

Costa, A. B. and Nardi, H. C. Homofobia e preconceito contra diversidade sexual: debate conceitual. *Temas em Psicologia*, 23:715 – 726, 2015.

Coutinho, V. M. and Malheiros, Y. Detecção de mensagens homofóbicas em português no twitter usando análise de sentimentos. pages 1–12, Porto Alegre, RS, Brasil. SBC, 2020.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. Automated hate speech detection and the problem of offensive language, 2017.

de Pelle, R. P. and Moreira, V. P. Offensive comments in the brazilian web: a dataset and baseline results, 2017.

Exame. *Redes sociais registram 4,62 bi de usuários - e vão continuar crescendo*, 2022.

Fadil, N. M. and Lavanya, S. Sentiment analysis of youtube comments in dravidian code-mixed language using deep neural network. volume 3395, page 100 – 105, 2022.

Fisher, R. A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

Fortuna, P., Rocha, J., Soler Company, J., Wanner, L., and Nunes, S. A hierarchically-labeled portuguese hate speech dataset. pages 94–104, 2019a.

Fortuna, P., Rocha da Silva, J., Soler-Company, J., Wanner, L., and Nunes, S. A hierarchically-labeled Portuguese hate speech dataset. In *Proceedings of the Third Workshop on Abusive Language Online*, Florence, Italy. Association for Computational Linguistics, 2019b.

García-Díaz, J., Caparros-Laiz, C., and Valencia-García, R. UMUTeam@LT-EDI-ACL2022: Detecting homophobic and transphobic comments in Tamil. In Chakravarthi, B. R., Bharathi, B., McCrae, J. P., Zarrouk, M., Bali, K., and Buitelaar, P., editors, *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 140–144, Dublin, Ireland. Association for Computational Linguistics, 2022.

Goffredo, P., Basile, V., Cepollaro, B., and Patti, V. Counter-twit: An italian corpus for online counterspeech in ecological contexts. page 57 – 66, 2022.

Goswami, D., Puspo, S. S. C., Raihan, M. N., and Emran, A. N. B. MasonTigers@LT-EDI-2024: An ensemble approach towards detecting homophobia and transphobia in social media comments. In Chakravarthi, B. R., B, B., Buitelaar, P., Durairaj, T., Kovács, G., and García Cumbreras, M. Á., editors, *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 164–172, St. Julian's, Malta. Association for Computational Linguistics, 2024.

Hatebase, I. How it works?, 2020.

Hegde, A., Kavya, G., Coelho, S., and Shashirekha, H. L. Mucs@lt-edi2023: Homophobic/transphobic content detection in social media text using mbert. page 287 – 294, 2023.

Hegde, A. and Shashirekha, H. L. Leveraging dynamic meta embedding for sentiment analysis and detection of homophobic/transphobic content in code-mixed dravidian languages. volume 3395, page 147 – 156, 2022.

Honey, C. and Herring, S. Beyond microblogging: Conversation and collaboration via twitter. pages 1 – 10, 2009.

Hossin, M. and M.N, S. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5:01–11, 2015.

Investe, V. *Brasil tem a quarta maior base de usuários do Twitter no mundo*, 2022.

Karayigit, H., Akdagli, A., and Aci, Homophobic and hate speech detection using multilingual-bert model on turkish social media, 2022.

Khurana, D., Koli, A., Khatter, K., and Singh, S. Natural language processing: State of the art, current trends and challenges, 2017.

Kumar, G., Singh, J. P., and Kumar, A. A deep multi-modal neural network for the identification of hate speech from social media. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12896 LNCS:670 – 680, 2021.

Kumaresan, P. K., Ponnusamy, R., Sharma, D., Buitelaar, P., and Chakravarthi, B. R. Dataset for identification of homophobia and transphobia for Telugu, Kannada, and Gujarati. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N., editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4404–4411, Torino, Italia. ELRA and ICCL, 2024.

Kusner, Matt e Sun, Y. and Kolkin, Nicholas e Weinberger, K. From word embeddings to document distances. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France. PMLR, 2015.

Lande, K., Ponnusamy, R., Kumaresan, P. K., and Chakravarthi, B. R. Kaustubhsharedtask@lt-edi 2023: Homophobia-transphobia detection in social media comments with nlpaug-driven data augmentation. page 71 – 77, 2023.

MacAvaney, S., Yao, H.-R., Yang, E., Russell, K., Goharian, N., and Frieder, O. Hate speech detection: Challenges and solutions. *PLOS ONE*, 14(8):1–16, 2019.

Maimaitituoheti, A., Yong, Y., and Xiaochao, F. Ablimet @lt-edi-acl2022: A roberta based approach for homophobia/transphobia detection in social media. page 155 – 160, 2022.

Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

Manukonda, D. P. and Kodali, R. G. bytellm@lt-edi-2024: Homophobia/transphobia detection in social media comments - custom subword tokenization with subword2vec and bilstm. Association for Computational Linguistics (ACL), 2024.

Meyer-Pflug, S. R. *Discurso do ódio*, 2008.

Monard, M. C. and Baranauskas, J. A. Conceitos sobre aprendizado de máquina. In *Sistemas Inteligentes Fundamentos e Aplicações*, pages 89–114. Manole Ltda, Barueri-SP, 1 edition, 2003.

Musto, C., Pio Sansonetti, A., Polignano, M., Semeraro, G., and Stranisci, M. Hatechecker: A tool to automatically detect hater users in online social networks. volume 2481, 2019.

Muti, A., Marchiori Manerba, M., Korre, K., and Barrón-Cedeño, A. LeaningTower@LT-EDI-ACL2022: When hope and hate collide. In Chakravarthi, B. R., Bharathi, B., McCrae, J. P., Zarrouk, M., Bali, K., and Buitelaar, P., editors, *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 306–311, Dublin, Ireland. Association for Computational Linguistics, 2022.

Nilsson, F., Al-Azzawi, S. S., and Kovács, G. Leveraging sentiment data for the detection of homophobic/transphobic content in a multi-task, multi-lingual setting using transformers. volume 3395, page 196 – 207, 2022.

Ninalga, D. Cordyceps@lt-edi: Patching language-specific homophobia/transphobia classifiers with a multilingual understanding. page 185 – 191, 2023.

Nozza, D. Nozza@lt-edi-acl2022: Ensemble modeling for homophobia and transphobia detection. page 258 – 264, 2022.

Organização das Nações Unidas. Transformando nosso mundo: A agenda 2030 para o desenvolvimento sustentável, 2015.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Ponnusamy, R., Kumaresan, P. K., Ponnusamy, K. K., Rajkumar, C., Priyadharshini, R., and Chakravarthi, B. R. Team_tamil at hodi: Few-shot learning for detecting homotransphobia in italian language. volume 3473, 2023.

Pranith, P., Samhita, V., Sarath, D., and Thenmozhi, D. Homophobia and transphobia detection of youtube comments in code-mixed dravidian languages using deep learning. volume 3395, page 117 – 123, 2022.

Quadrado, J. C. and Ferreira, E. d. S. Ódio e intolerância nas redes sociais digitais. *Rev. katálysis 23 (03)*, 2020.

Rajalakshmi, R., Duraphe, A., and Shibani, A. DLRG@DravidianLangTech-ACL2022: Abusive comment detection in Tamil using multilingual transformer models. In Chakravarthi, B. R., Priyadharshini, R., Madasamy, A. K., Krishnamurthy, P., Sherly, E., and Mahesan, S., editors, *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 207–213, Dublin, Ireland. Association for Computational Linguistics, 2022.

Recuero, R. *A conversação em rede: comunicação mediada pelo computador e redes sociais na Internet*. Porto Alegre: Sulina, 2012.

Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. *Recommender Systems Handbook*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010.

Saha, K., Kim, S. C., Reddy, M. D., Carter, A. J., Sharma, E., Haimson, O. L., and Choudhury, M. D. The language of lgbtq+ minority stress experiences on social media. *Proceedings of the ACM on Human-Computer Interaction*, 3:1 – 22, 2019.

Samuel, A. A., Kumar, L. S., Navaneethakrishnan, S. C., and Sakuntharaj, R. A sequential dnn for sentiment analysis of dravidian code-mixed language comments on youtube. volume 3395, page 177 – 183, 2022.

Saumya, S., Jha, V., and Biradar, S. Sentiment and homophobia detection on youtube using ensemble machine learning techniques. volume 3395, page 92 – 99, 2022.

Shanmugavadivel, K., Subramanian, M., Shri Durga, R., Srigha, S., Samyuktha, K., and Nithika, K. Kec-ai-nlp@lt-edi-2024:homophobia and transphobia detection in social media comments using machine learning. Association for Computational Linguistics (ACL), 2024.

Shanmugavadivel, K., Subramanian, M., Shri Durga, R., Srigha, S., Sree Harene, J., and Yasvanth Bala, P. Kec ai nlp@dravidianlangtech:abusive comment detection in tamil language using machine learning techniques. page 293 – 299, 2023.

Sharma, D., Gupta, V., and Singh, V. K. Detection of homophobia transphobia in malayalam and tamil: Exploring deep learning methods. *Communications in Computer and Information Science*, 1798 CCIS:217 – 226, 2023.

Sharma, D., Gupta, V., and Singh, V. K. *Abusive comment detection in Tamil using deep learning*. Elsevier, 2024.

Sheik, R., Balanathan, R., and S., J. N. Mitigating abusive comment detection in Tamil text: A data augmentation approach with transformer model. In D. Pawar, J. and Lalitha Devi, S., editors, *Proceedings of the 20th International Conference on Natural Language Processing (ICON)*, pages 460–465, Goa University, Goa, India. NLP Association of India (NLPAI), 2023.

Shwetha, S., Sivakumar, S., Priyadharshini, T., Thenmozhi, D., Bharathi, B., and Krithika, S. Tercet@lt-edi: Homophobia/transphobia detection in social media comment. page 266 – 271, 2023.

Silva, A. and Roman, N. Hate speech detection in portuguese with naïve bayes, svm, mlp and logistic regression. In *Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional*, pages 1–12, Porto Alegre, RS, Brasil. SBC, 2020.

Silva, L. R. L., Botelho-Francisco, R. E., Alisson Augusto de Oliveira, A. A. d., and Pontes, V. R. A gestão do discurso de ódio nas plataformas de redes sociais digitais: um comparativo entre facebook, twitter e youtube. *Revista Ibero-Americana de Ciência da Informação*, 12:470–492, 2019.

Silva, R. L., Nichel, A., Martins, A. C. L., and Borchardt, C. K. *Discursos de ódio em redes sociais: jurisprudência brasileira = Hate speech in social networks: brazilian case law*, 2011.

Singh, M. and Motlicek, P. Idiap submission@lt-edi-acl2022: Homophobia/transphobia detection in social media comments. page 356 – 361, 2022.

Sivaprasath, S., Kumar, L. S., and Thavareesan, S. Homophobia, transphobia detection in tamil, malayalam, english languages using logistic regression and code-mixed data using awd-lstm. volume 3395, page 169 – 176, 2022.

Sonali, Gidnakanala, N., Raksha, G., Kavya, G., Hegde, A., and Shashirekha, H. Mucs@lt-edi-2024: Learning approaches to empower homophobic/transphobic comment identification. Association for Computational Linguistics (ACL), 2024.

S.PAULO, O. E. D. Youtuber contratado pelo governo fez posts racistas e homofóbicos no Twitter, 2017.

Swaminathan, K., Sampath, H., Gayathri, G., and Bharathi, B. Ssncse_nlp@lt-edi-acl2022: Homophobia/transphobia detection in multiple languages using svm classifiers and bert-based transformers. page 239 – 244, 2022.

Twitter, I. Perguntas frequentes de novos usuários, 2020.

Ugursandi, S. and Kumar, A. M. Sentiment analysis and homophobia detection of youtube comments. volume 3395, page 157 – 168, 2022.

UOL. *Grupo gay condena perfil homofóbico que já tem 15 mil adeptos no Twitter*, 2010.

UOL. *Entenda o que significa cada uma das letras desta bandeira de luta*, 2020.

Upadhyay, I. S., Srivatsa, K. A., and Mamidi, R. Sammaan@LT-EDI-ACL2022: Ensembled transformers against homophobia and transphobia. In Chakravarthi, B. R., Bharathi, B., McCrae, J. P., Zarrouk, M., Bali, K., and Buitelaar, P., editors, *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 270–275, Dublin, Ireland. Association for Computational Linguistics, 2022.

Vaidhegi, D., Priya, M., Rajalakshmi, S., Angel Deborah, S., and Mirnalinee, T. Ssntech2@lt-edi-ranlp-2023: Homophobia/transphobia detection in social media comments using linear classification techniques. page 166 – 171, 2023.

Vandersmissen, B. Automated detection of offensive language behavior on social networking sites, 2012.

Varsha, J., Bharathi, B., and Meenakshi, A. Sentiment analysis and homophobia detection of youtube comments in code-mixed dravidian languages using machine learning and transformer models. volume 3395, page 124 – 137, 2022.

Vásquez, J., Andersen, S., Bel-enguix, G., Gómez-adorno, H., and Ojeda-trueba, S.-l. HOMO-MEX: A Mexican Spanish annotated corpus for LGBT+phobia detection on Twitter. In Chung, Y.-l., R\"ottger, P., Nozza, D., Talat, Z., and Mostafazadeh Davani, A., editors, *The 7th Workshop on Online Abuse and Harms (WOAH)*, pages 202–214, Toronto, Canada. Association for Computational Linguistics, 2023.

Vilela, M. O léxico do português: perspectiva geral. *Filologia e Linguística Portuguesa*, (1):31–50, 1997.

Wong, S. G.-J. and Durward, M. cantnlp@lt-edi-2024: Automatic detection of anti-lgbtq+ hate speech in under-resourced languages. Association for Computational Linguistics (ACL), 2024.

Wong, S. G.-J., Durward, M., Adams, B., and Dunn, J. cantnlp@lt-edi-2023: Homophobia/-transphobia detection in social media comments using spatio-temporally retrained language models. page 103 – 108, 2023.

Yadav, S., Kaushik, A., and McDaid, K. dkit@lt-edi-2024: Detecting homophobia and transphobia in english social media comments. Association for Computational Linguistics (ACL), 2024.

Apêndice A - API Twitter V2

O Programa de Pesquisa Acadêmica do Twitter concede acesso à API do Twitter para pesquisadores acadêmicos, de maneira gratuita. A submissão se deu através de acesso a `https://developer.twitter.com/en/use-cases/do-research/academic-research`, inserindo um breve relato da pesquisa que se quiser realizar, indicando o professor orientador, a instituição de ensino e uma comprovação do vínculo do pesquisador com a instuição.

A figura VI.1 ilustra o email de confirmação da submissão, em seguida, no mesmo dia, foi recebido a aprovação da submissão conforme VI.2.

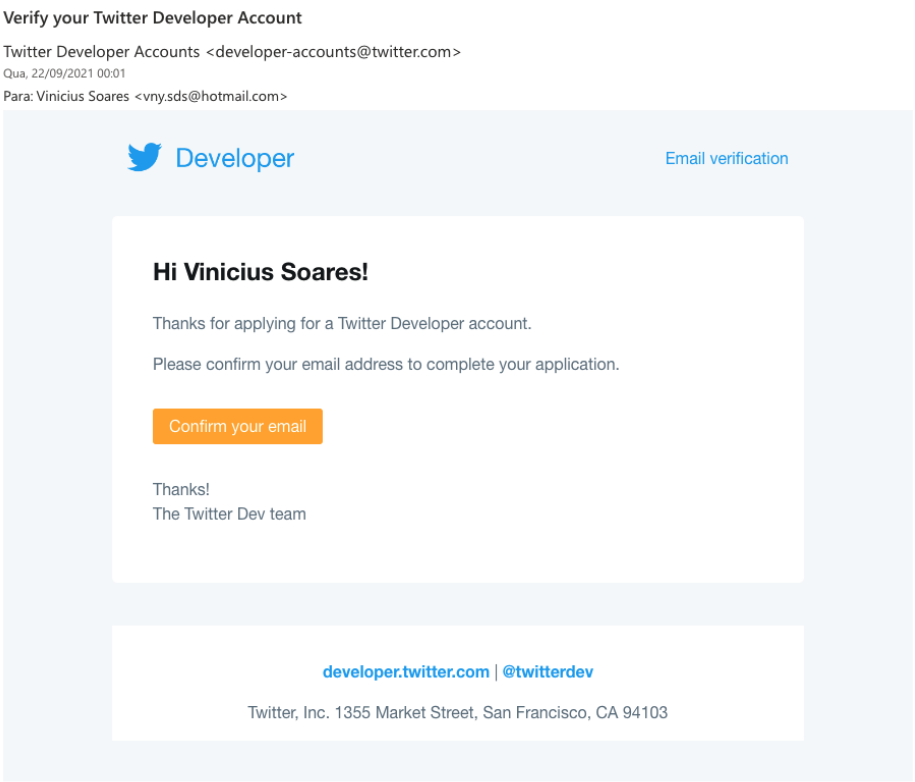


Figura VI.1: E-mail de confirmação da submissão API Twitter v2 - Pesquisador

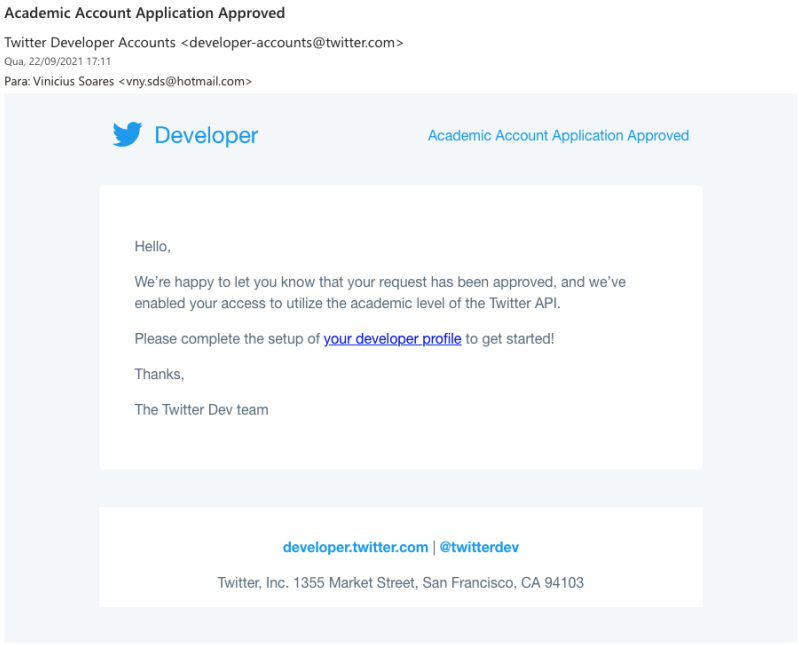


Figura VI.2: E-mail de aprovação da submissão API Twitter v2 - Pesquisador

A figura VI.3 ilustra o resumo do projeto na API, bem como as credenciais de acesso a mesma.

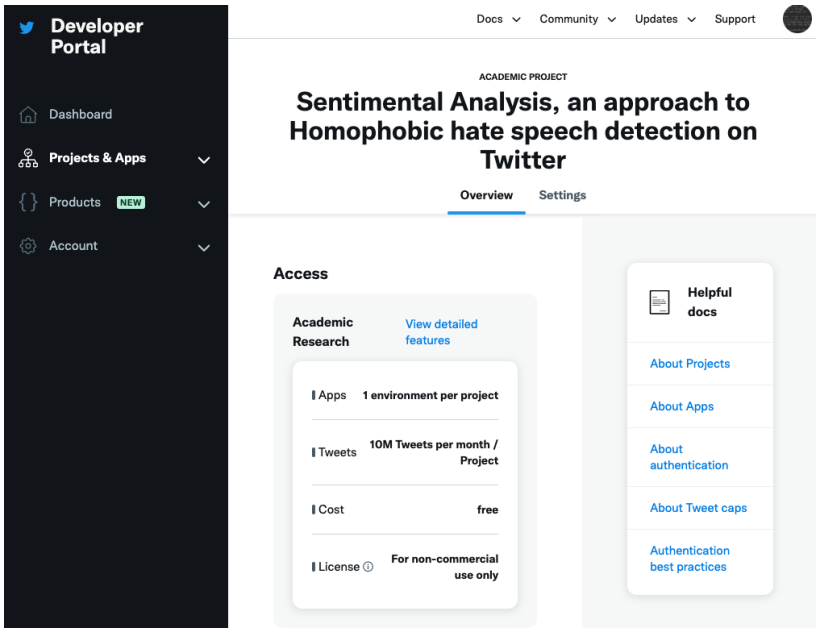
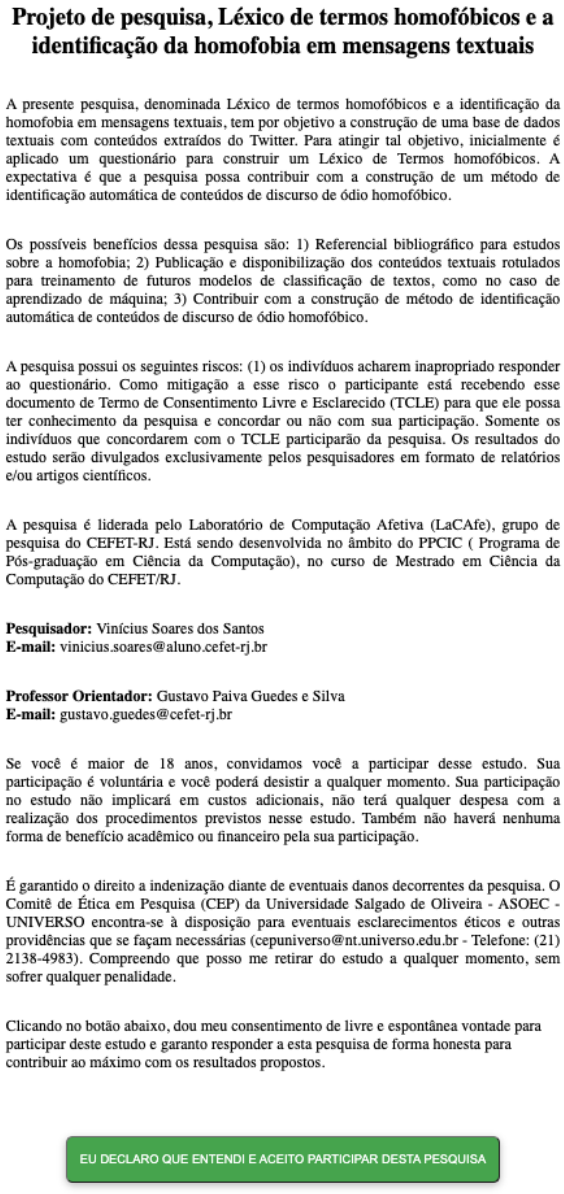


Figura VI.3: Área do desenvolvedor API Twitter v2 - Pesquisador

Apêndice B - Termo de consentimento livre e esclarecido



Projeto de pesquisa, Léxico de termos homofóbicos e a identificação da homofobia em mensagens textuais

A presente pesquisa, denominada Léxico de termos homofóbicos e a identificação da homofobia em mensagens textuais, tem por objetivo a construção de uma base de dados textuais com conteúdos extraídos do Twitter. Para atingir tal objetivo, inicialmente é aplicado um questionário para construir um Léxico de Termos homofóbicos. A expectativa é que a pesquisa possa contribuir com a construção de um método de identificação automática de conteúdos de discurso de ódio homofóbico.

Os possíveis benefícios dessa pesquisa são: 1) Referencial bibliográfico para estudos sobre a homofobia; 2) Publicação e disponibilização dos conteúdos textuais rotulados para treinamento de futuros modelos de classificação de textos, como no caso de aprendizado de máquina; 3) Contribuir com a construção de método de identificação automática de conteúdos de discurso de ódio homofóbico.

A pesquisa possui os seguintes riscos: (1) os indivíduos acharem inapropriado responder ao questionário. Como mitigação a esse risco o participante está recebendo esse documento de Termo de Consentimento Livre e Esclarecido (TCLE) para que ele possa ter conhecimento da pesquisa e concordar ou não com sua participação. Somente os indivíduos que concordarem com o TCLE participarão da pesquisa. Os resultados do estudo serão divulgados exclusivamente pelos pesquisadores em formato de relatórios e/ou artigos científicos.

A pesquisa é liderada pelo Laboratório de Computação Afetiva (LaCAfe), grupo de pesquisa do CEFET-RJ. Está sendo desenvolvida no âmbito do PPCIC (Programa de Pós-graduação em Ciência da Computação), no curso de Mestrado em Ciência da Computação do CEFET/RJ.

Pesquisador: Vinícius Soares dos Santos
E-mail: vinicius.soares@aluno.cefet-rj.br

Professor Orientador: Gustavo Paiva Guedes e Silva
E-mail: gustavo.guedes@cefet-rj.br

Se você é maior de 18 anos, convidamos você a participar desse estudo. Sua participação é voluntária e você poderá desistir a qualquer momento. Sua participação no estudo não implicará em custos adicionais, não terá qualquer despesa com a realização dos procedimentos previstos nesse estudo. Também não haverá nenhuma forma de benefício acadêmico ou financeiro pela sua participação.

É garantido o direito a indenização diante de eventuais danos decorrentes da pesquisa. O Comitê de Ética em Pesquisa (CEP) da Universidade Salgado de Oliveira - ASOEC - UNIVERSO encontra-se à disposição para eventuais esclarecimentos éticos e outras providências que se façam necessárias (cepuniverso@nt.universo.edu.br - Telefone: (21) 2138-4983). Compreendo que posso me retirar do estudo a qualquer momento, sem sofrer qualquer penalidade.

Clicando no botão abaixo, dou meu consentimento de livre e espontânea vontade para participar deste estudo e garanto responder a esta pesquisa de forma honesta para contribuir ao máximo com os resultados propostos.

EU DECLARO QUE ENTENDI E ACEITO PARTICIPAR DESTA PESQUISA

Figura VI.4: Termo de consentimento livre e esclarecido

Apêndice C - Léxico homofóbico

Termos Homofóbicos
afeminada
afetada
agasalha croquete
aidetica
baitola
bambi
biba
bicha
boiola
caminhoneira
coca cola é fanta
cola velcro
dá a rosca
dá cu
dá ré no kibe
desmunheca
desviada
enrustida
entendida
escorregar no quiabo
fancha
flozô
franga
fresco
fruta
fufa
gay
guena
indeciso
kitty
mamador
manja rola
marica
maricona
mocinha
morde a fronha
mulher macho
mulherzinha
paneleira
pão com ovo
passiva
princesa
qualira
queima rosca
rabeta
sabonete
sapa
sapatão
tchola
transviado
traveco
viado
xibungo
xinxa