UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

UMA ABORDAGEM PARA DETECÇÃO DE DISCURSO DE ÓDIO UTILIZANDO APRENDIZADO DE MÁQUINA BASEADO EM CRUZAMENTO DE IDIOMAS

ANDERSON ALMEIDA FIRMINO

Campina Grande - Paraíba, maio de 2022

ANDERSON ALMEIDA FIRMINO

UMA ABORDAGEM PARA DETECÇÃO DE DISCURSO DE ÓDIO UTILIZANDO APRENDIZADO DE MÁQUINA BASEADO EM CRUZAMENTO DE IDIOMAS

Tese submetida à **Coordenação do Curso de Pós-Graduação em Ciência da Computação** da **Universidade Federal de Campina Grande**, como requisito parcial à obtenção do título de **Doutor em Ciência da Computação**.

Orientador:

Prof. Cláudio de Souza Baptista, Ph.D.

Campina Grande - Paraíba, maio de 2022

F525a Firmino, Anderson Almeida.
Uma abordagem para detecção de discurso de ódio utilizando aprendizado de máquina baseado em cruzamento de idiomas / Anderson Almeida Firmino. – Campina Grande, 2022.
97 f. : il. color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2022.
"Orientação: Prof. Dr. Cláudio de Souza Baptista".
Referências.

1. Processamento de Linguagem Natural. 2. Detecção de Discurso de Ódio. 3. Redes Sociais. 4. Cross-Lingual Learning. I. Baptista, Cláudio de Souza. II. Título.

CDU 004.438:81'322.2(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA SEVERINA SUELI DA SILVA OLIVEIRA CRB-15/225

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
POS-GRADUACAO CIENCIAS DA COMPUTACAO
Rua Aprigio Veloso, 882, - Bairro Universitario, Campina Grande/PB, CEP 58429-900

FOLHA DE ASSINATURA PARA TESES E DISSERTAÇÕES

ANDERSON ALMEIDA FIRMINO

UMA ABORDAGEM PARA DETECÇÃO DE DISCURSO DE ÓDIO UTILIZANDO APRENDIZADO DE MÁQUINA BASEADO EM CRUZAMENTO DE IDIOMAS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação como pré-requisito para obtenção do título de Doutor em Ciência da Computação.

Aprovada em: 18/05/2022

Prof. Dr. CLÁUDIO DE SOUZA BAPTISTA, Orientador, UFCG

Prof. Dr. HERMAN MARTINS GOMES, Examinador Interno, UFCG

Prof. Dr. EANES TORRES PEREIRA, Examinador Interno, UFCG

Prof. Dr. GERALDO BRAZ JUNIOR, Examinador Externo, UFMA

Prof. Dr. WINDSON VIANA DE CARVALHO, Examinador Externo, UFC

Documento assinado eletronicamente por **CLAUDIO DE SOUZA BAPTISTA**, **PROFESSOR 3 GRAU**, em 18/05/2022, às 11:18, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **HERMAN MARTINS GOMES**, **PROFESSOR 3 GRAU**, em

18/05/2022, às 11:44, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Geraldo Braz Junior**, **Usuário Externo**, em 18/05/2022, às 19:21, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **EANES TORRES PEREIRA**, **PROFESSOR 3 GRAU**, em 26/05/2022, às 13:14, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

A autenticidade deste documento pode ser conferida no site https://sei.ufcg.edu.br/autenticidade, informando o código verificador **2399488** e o código CRC **86B8DF26**.

Referência: Processo nº 23096.033891/2022-88 SEI nº 2399488

AGRADECIMENTOS

Agradeço primeiramente a Deus pela oportunidade de ter concluído mais esta etapa em minha vida. Sou grato pela força dada nos momentos difíceis e por Ele ter me sustentado até aqui.

Agradeço a todos os meus familiares por todo o suporte dado. Em especial, agradeço à minha mãe por sempre ter me incentivado ao longo da caminhada.

Agradeço a todos os meus amigos do grupo Dunamis por terem estado junto comigo nos momentos bons e ruins. Por terem orado sempre que eu pedi, por terem paciência e por toda a ajuda que vocês me deram. Vocês são especiais!

Agradeço ao meu orientador, professor Cláudio Baptista. Sou grato por ter acreditado em mim desde a graduação. Sou grato pelos conselhos, pelas conversas, e até mesmo pelos puxões de orelha quando necessário. Obrigado pela confiança e parceria durante todos estes anos.

Agradeço ao professor Anselmo Paiva que, apesar de não figurar oficialmente como coorientador deste trabalho, sua participação foi imprescindível. Sou grato por todas as contribuições e por ter atendido aos meus chamados, mesmo em finais de semana.

Agradeço também aos meus colegas do Laboratório de Sistemas de Informação (LSI) pelo clima agradável de trabalho, pelos cafés, salgadinhos, conversas e boas risadas.

Enfim, agradeço a todos os que colaboraram de forma direta ou indireta para a conclusão deste trabalho.

RESUMO

O crescimento das mídias sociais em todo o mundo trouxe benefícios e desafios para a sociedade. Dentre os desafios, destaca-se a proliferação do discurso de ódio nas redes sociais. Hodiernamente, a detecção de discurso do ódio tornou-se uma tarefa árdua. Cerca de 22,5 milhões de postagens com discurso de ódio foram removidas nas redes sociais entre abril e junho de 2020. Destarte, faz-se necessário o desenvolvimento de pesquisas que busquem soluções automatizadas para identificar e remover discurso de ódio nas redes sociais. Nesta tese, propõe-se uma nova metodologia para detecção de discurso de ódio em textos em português. Esta metodologia faz uso de *Cross-Lingual Learning*, que consiste em usar transferência de aprendizagem em Modelos de Linguagem Pré-Treinados (MLPTs) com um idioma com grandes corpora disponíveis (idioma fonte) para resolver problemas em idiomas com menos dados anotados (idioma alvo). A metodologia proposta compreende quatro etapas: aquisição de corpora, definição de MLPT, estratégias de treinamento e avaliação. Foram realizados experimentos utilizando Modelos de Linguagem Pré-Treinados em diferentes idiomas: Inglês, Italiano e Português (BERT e XLM-R) para verificar qual deles se adequava melhor ao método proposto. Corpora em inglês (WH) e italiano (Evalita 2018) foram utilizados como idioma fonte e dois corpora em português (idioma alvo) foram utilizados: OffComBr-2 e *Hate Speech Dataset* (HSD). Os resultados dos experimentos demonstraram que a metodologia proposta é competitiva com o estado da arte: para o corpus OffComBr-2 obteve-se o melhor resultado dentre os trabalhos que utilizaram o mesmo corpus, com Medida F1 = 92%; e para o corpus HSD, obteve-se o segundo melhor resultado, com Medida F1 = 90%.

Palavras-chave: *Cross-Lingual Learning*, Detecção de Discurso de Ódio, Redes Sociais, Processamento de Linguagem Natural.

ABSTRACT

The growth of social media around the world has brought both benefits and challenges to society. Among the challenges, we highlight the proliferation of hate speech in social networks. Detecting hate speech has become an arduous task in today's world. About 22.5 million posts with hate speech were removed from social networks between April and June 2020. Thus, it is necessary to develop research that seek automated solutions to identify and remove hate speech in social networks. In this thesis, we propose a new methodology for detecting hate speech in Portuguese texts. This methodology uses Cross-Lingual Learning, which consists of using transfer learning in Pre-Trained Language Models with a language with large corpora available (source language) to solve problems in languages with less annotated data (target language). The proposed methodology comprises four stages: corpora acquisition, definition of PTLM, training strategies and evaluation. We carried out experiments using Pre-Trained Language Models in different languages: English, Italian and Portuguese (BERT and XLM-R) to verify which one best suited the proposed method. Corpora in English (WH) and Italian (Evalita 2018) were used as source language and two corpora in Portuguese (target language) were used: OffComBr-2 and Hate Speech Dataset (HSD). The results of the experiments showed that the proposed methodology is promising: for the OffComBr-2 corpus, the best state-of-the-art result was obtained (F1 Score = 92%); and for the HSD corpus, the second best result was obtained (F1 Score = 90%).

Keywords: Cross-Lingual Learning, Hate Speech Detection, Social Media, Natural Language Processing.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
AUC	*Area Under the Curve*
BERT	*Bidirectional Encoder Representations from Transformers*
CL	*Cascade Learning*
CLL	*Cross-Lingual Learning*
CNN	*Convolutional Neural Network*
Evalita	*Evaluation of NLP and Speech Tools for Italian*
GPU	*Graphics Processing Unit*
GRU	*Gated Recurrent Unit*
HSD	*Hate Speech Dataset*
IA	Inteligência Artificial
JL	*Joint Learning*
LSTM	*Long-Short Term Memory*
MLP	*Multi-Layer Perceptron*
MLPT	Modelo de Linguagem Pré-Treinado
OffComBr	*Offensive Comments in the Brazilian Web*
PLN	Processamento de Linguagem Natural
POS	*Part Of Speech*
RI	Recuperação da Informação
RN	Rede Neural
SVM	*Support-Vector Machine*
XLM	*Cross-Lingual Language Model*
ZST	*Zero-shot Transfer*

SUMÁRIO

1 INTRODUÇÃO

O crescimento da tecnologia móvel trouxe diversos benefícios para a sociedade. De acordo com uma pesquisa recente (SHEARER; MITCHELL, 2021), as pessoas preferem usar seus *smartphones* e mídias sociais para consumo de notícias em vez de jornais impressos e televisão. Deste modo, elas não precisam esperar horários fixos para assistirem a telejornais e se atualizarem. Com os dispositivos móveis, é possível ver notícias em tempo real. Plataformas como Facebook e Twitter estão entre as mais utilizadas no tocante ao acesso a informações e notícias.

Relatórios disponibilizados pelo *Data Reportal* mostram que o uso mais comum de dispositivos móveis tem sido para atividades sociais em todo o mundo (KEMP, 2021). Cada vez mais pessoas querem manifestar suas opiniões pessoais sobre diversos assuntos em veículos de imprensa online (MATHEW et al., 2019). Segundo os relatórios citados, a maioria das pessoas entrevistadas costumava compartilhar vídeos e fotos, muitas vezes em suas redes sociais. Além disso, mais da metade dos usuários de telefones celulares e mídias sociais nos países pesquisados já postaram algo sobre suas reflexões sobre assuntos que consideram importantes.

Apesar dos benefícios e da comodidade que as mídias sociais aprovisionam, o anonimato proporcionado por esses meios pode ser prejudicial à sociedade, tendo em vista que as pessoas podem assumir um comportamento mais agressivo no uso de suas redes (FORTUNA; NUNES, 2018). Um exemplo disso é a crescente proliferação de discurso de ódio na Internet. Tendo isso em mente, a Comissão da União Europeia conduziu e financiou várias iniciativas para enfrentar este comportamento. Em 2016, a mesma comissão pressionou várias plataformas de mídia social e organizações como Facebook, Microsoft, Twitter e YouTube a assinarem um Código de Discurso de ódio, que inclui diretrizes como a revisão de notificações válidas para remoção de discurso de ódio em 24 horas.

Fortuna e Nunes (2018) definem o discurso de ódio como uma linguagem que ataca e incita a violência contra certos grupos de pessoas com base em suas características específicas, como aparência física, religião, descendência, nacionalidade ou origem étnica e gênero.

A detecção de discurso de ódio tornou-se uma tarefa árdua no mundo atual. Enquanto algumas pessoas tentam combater a prática, parte da internet não parece se incomodar com a proliferação de robôs que espalham preconceito e notícias falsas. De acordo com a Bloomberg, o Facebook removeu 22,5 milhões de postagens com discurso de ódio

entre abril e junho de 2020 (o número é o dobro do que foi removido no primeiro trimestre do mesmo ano). Além disso, o Facebook anunciou que seus termos foram atualizados para banir o "discurso de ódio implícito" (WAGNER, 2020).

A detecção e remoção de comentários de discurso de ódio na Internet são questões extremamente relevantes para a sociedade, uma vez que muitos estudos correlacionam discurso de ódio com crimes (BURNAP; WILLIAMS, 2016; SCHMIDT; WIEGAND, 2017; MONDAL et al., 2018). A identificação e o monitoramento de usuários que espalham comentários odiosos podem prevenir esse tipo de ataque à sociedade.

É importante ressaltar que, apesar da liberdade de expressão ser reconhecida como uma das condições fundamentais de uma sociedade democrática, existem limitações no exercício desta liberdade. O discurso de ódio pode se enquadrar nestas limitações, a depender do ponto de vista jurídico. Assim, a tentativa de reduzir o discurso de ódio é motivada não apenas por questões práticas de retenção de usuários, mas debruça-se também sobre questões jurídicas (SILVA, 2020). No Capítulo 2, é contemplada uma análise preliminar de aspectos jurídicos concernentes ao discurso de ódio.

A detecção de discurso de ódio envolve vários desafios, dentre os quais destaca-se a dificuldade em identificar todos os insultos contra grupos, em função dos fenômenos sociais e da evolução da linguagem, que cresce em alta velocidade principalmente entre os jovens que acessam as redes sociais com frequência. Além disso, requer-se conhecimento da cultura local, bem como da organização social da comunidade pesquisada. Por fim, tem-se o predomínio da língua inglesa sendo explorada nas pesquisas da área. Assim, outros idiomas têm poucos ou nenhum trabalho em relação à detecção de discurso de ódio.

A presente tese de doutorado deverá permitir mitigar este último problema explicado. Conforme mencionado acima, a grande maioria das pesquisas revisadas nesta tese sobre detecção de discurso de ódio usa bases de dados em inglês. Utilizamos bases de dados em outras línguas, principalmente o português, por ser a nossa língua materna e por haver poucas pesquisas que utilizam esse idioma.

Conforme discutido por Pikuliak et al. (2021), a maioria dos idiomas não tem dados suficientes disponíveis para criar modelos de última geração. Portanto, a capacidade de criar sistemas inteligentes para essas linguagens é restrita. A importância das tarefas de Processamento de Linguagem Natural (PLN) para idiomas com menos recursos (ou dados anotados) surgiu recentemente durante várias crises em regiões do mundo onde as pessoas falam idiomas que não são comumente tratados na comunidade PLN, como os surtos de Ebola na África Ocidental (por exemplo, línguas do Níger-Congo).

De acordo com pesquisas relacionadas a conferências recentes de PLN, o inglês é o idioma mais pesquisado e é o único idioma considerado em mais de 60% dos artigos publicados (PIKULIAK et al., 2021). Uma potencial solução para resolver esse problema é a utilização de cruzamento de idiomas (do inglês, *Cross-Lingual Learning* - CLL (PAMUNGKAS; PATTI, 2019; HU et al., 2020; BIGOULAEVA et al., 2021)), que consiste na criação de soluções para idiomas com poucos dados anotados usando idiomas com muitos dados disponíveis. Assim, modelos de aprendizagem podem ser empregados e utilizados nestes idiomas com poucos dados.

Nesta tese, investiga-se a detecção de discurso de ódio em textos utilizando cruzamento de idiomas. Os idiomas utilizados neste trabalho são o italiano, o inglês e o português. Português e italiano são originários da mesma língua materna - o latim, enquanto que inglês possui origem anglo-saxônica. Os corpora em português utilizados foram OffComBr-2 (PELLE; MOREIRA, 2017) e HSD (FORTUNA et al., 2019), e os corpora em italiano utilizados foram disponibilizados no Evalita 2018, na tarefa *Hate Speech Detection* (HaSpeeDe) (BOSCO et al., 2018). Já o corpus em inglês utilizado foi disponibilizado por Waseem e Hovy (2016).

1.1 PROBLEMAS DE PESQUISA

A pesquisa corrente busca avaliar a viabilidade do uso de diferentes idiomas como idioma fonte, como italiano e inglês, na construção da solução proposta. Para atender às demandas desta pesquisa, busca-se responder às seguintes questões de pesquisa:

- Q1: O uso de CLL melhora o desempenho do modelo proposto?
- Q2: O uso de CLL traz resultados relevantes em relação ao estado da arte?
- Q3: O uso de CLL com um idioma de base latina como idioma fonte melhora o desempenho do modelo proposto?
- Q4: O uso de CLL com um idioma de base anglo-saxônica como idioma fonte melhora o desempenho do modelo proposto?

1.2 OBJETIVOS

O objetivo principal desta pesquisa é desenvolver uma abordagem para realizar a detecção de discurso de ódio em português usando cruzamento de idiomas (CLL), tendo em vista que há poucas bases de dados disponíveis em português sobre discurso de ódio. Os seguintes objetivos específicos foram definidos para atingir o objetivo principal:

- Desenvolver uma abordagem para detecção de discurso de ódio em português utilizando CLL;
- Avaliar a adequabilidade de usar uma língua latina - como o Italiano - como idioma fonte em CLL para detecção de discurso de ódio em português;
- Avaliar a adequabilidade de usar uma língua de origem anglo-saxônica - como o Inglês - como idioma fonte em CLL para detecção de discurso de ódio em português;

1.3 CONTRIBUIÇÕES

As contribuições desta tese residem na inovação do uso de CLL com dados de discurso de ódio em português, além de ter obtido o melhor resultado da literatura com a base de dados OffComBr2 - fornecida por Pelle e Moreira (2017). Além disso, foram obtidos bons resultados com o corpus de Fortuna et al. (2019).

Deste modo, tem-se uma estratégia que permite atingir bons resultados em classificação de textos utilizando CLL, mesmo que no idioma alvo não haja corpora suficientes. Outra contribuição desta tese está na investigação sobre os idiomas que podem ser úteis como idiomas fonte na metodologia proposta, tendo o português como idioma alvo.

1.4 PUBLICAÇÕES

A primeira publicação (FIRMINO et al., 2021) consistiu de um artigo enviado para a conferência DEXA 2021, cujo *qualis* é A4. Neste artigo, apenas uma versão inicial dos resultados foi enviada. Foi utilizado apenas um Modelo de Linguagem Pré-Treinado (XLM-R), e foi utilizado apenas o corpus de Pelle e Moreira (2017).

Uma segunda publicação com resultados mais completos foi enviada para o periódico *Expert Systems With Applications* (*qualis* A1). Esta publicação ainda encontra-se sob revisão, e contém os resultados presentes nesta tese.

1.5 ESTRUTURA

O restante desta tese está estruturado da seguinte forma. No Capítulo 2, é fornecida uma base conceitual a respeito de cruzamento de idiomas e detecção de discurso de ódio. No Capítulo 3, concentra-se a revisão do estado da arte na detecção de discurso de ódio em textos. A metodologia e os *corpora* usados nesta tese são abordados no Capítulo 4. No Capítulo 5, tratam-se dos experimentos realizados e resultados obtidos. Finalmente, no Capítulo 6 são destacadas conclusões e trabalhos futuros identificados.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo contém as principais definições e conceitos que serão explorados ao longo desta tese, sendo estruturado da seguinte forma: na Seção 2.1 é apresentada uma discussão sobre Processamento de Linguagem Natural (PLN); na Seção 2.2 discute-se sobre aprendizado de máquina supervisionado, e de forma mais específica, redes neurais profundas, *word embeddings*, transferência de aprendizagem e métricas de avaliação. Na Seção 2.3, é trazida uma discussão sobre detecção de discurso de ódio e na Seção 2.4 Cruzamento de Idiomas é explorado.

2.1 PROCESSAMENTO DE LINGUAGEM NATURAL

De acordo com Chowdhury (2003), Processamento de Linguagem Natural (PLN) é uma coleção de técnicas para tornar a fala humana compreensível por computadores. De uma perspectiva científica, o PLN tenta modelar os princípios cognitivos subjacentes à compreensão e geração de linguagens humanas. Do ponto de vista da engenharia, o PLN concentra-se no desenvolvimento de novas aplicações práticas para facilitar as interações entre computadores e linguagens humanas.

O Processamento de Linguagem Natural é um campo interdisciplinar que combina várias áreas. Uma delas é a linguística computacional. A linguagem é o objeto de estudo em linguística. As abordagens computacionais podem ser usadas, como o são em disciplinas científicas como biologia computacional e astronomia computacional, embora sirvam apenas como um suplemento. Por outro lado, o foco do processamento da linguagem natural está no desenvolvimento e análise de técnicas de computação e representações para o processamento da linguagem humana natural. O PLN visa a fornecer novas habilidades de computação no contexto das linguagens humanas, como extrair informações de textos, traduzir entre línguas, responder a perguntas e manter uma conversa (EISENSTEIN, 2018).

Outra área que vale a pena ser destacada é o aprendizado de máquina. Aprendizado de máquina (AM), que permite aquisição automática de conhecimento, é usado em abordagens modernas para o processamento de linguagem natural. Tradicionalmente, um dos principais focos de aprendizado de máquina é resolver problemas de classificação: dado um *corpus* de documentos, classifique cada documento de acordo com seu tema rótulo. O AM oferece várias estratégias para tarefas como a conversão de uma sequência de *tokens* discretos em um vocabulário em uma sequência de *tokens* discretos em outro vocabulário - uma generalização do que é comumente referido como tradução. Grande parte da pesquisa

atual de processamento de linguagem natural pode ser classificada como aprendizado de máquina aplicado (CHOWDHURY, 2003; SAMMUT; WEBB, 2011). O processamento de linguagem natural, por outro lado, tem propriedades que o diferenciam de muitos outros campos de aplicações de aprendizado de máquina. Um dos desafios de se trabalhar com dados textuais é o fato de a linguagem ser composicional: unidades como palavras podem ser combinadas para formar frases, que podem ser combinadas usando os mesmos princípios para formar parágrafos.

A terceira área que pode ser citada é a ciência da computação. O PLN está ligado ao estudo dos sistemas computacionais. Técnicas de paralelização como *MapReduce* (DEAN; GHEMAWAT, 2008) podem acelerar o processamento de grandes conjuntos de dados de texto não rotulados, e abordagens de *streaming* podem resumir fontes de dados de alto volume, como dados de mídias sociais. Muitas técnicas tradicionais de processamento de linguagem natural não são adequadas para a paralelização em GPU, mostrando novas direções para a pesquisa na interface de processamento de linguagem natural e paralelização de unidades de processamento gráfico. Os avanços de aprendizado profundo são uma das principais forças motrizes por trás do PLN atual e do ponto de inflexão da inteligência artificial mais geral. Neste contexto, nota-se o renascimento das redes neurais com uma ampla gama de aplicações práticas, incluindo aquelas para a indústria (DENG; LIU, 2018).

Por fim, também pode-se citar o reconhecimento de fala. O reconhecimento de fala é a tarefa de transformar um fluxo de texto de áudio, que é como a linguagem natural é comumente comunicada (YU; DENG, 2016). De um ponto de vista, este é um problema de processamento de sinal que pode ser considerado uma etapa de pré-processamento antes do processamento de linguagem natural. Os ouvintes humanos, por outro lado, dependem muito do contexto para o reconhecimento de fala: o conhecimento das palavras ao redor altera a percepção e auxilia na correção do ruído. Como resultado, o reconhecimento de fala é frequentemente combinado com a análise de texto, particularmente modelos estatísticos de linguagem, que estimam a probabilidade de uma sequência de texto (CHOWDHURY, 2003).

Tradução automática (LEE et al., 2009; LUQMAN; MAHMOUD, 2019), resumo (LIU; LAPATA, 2019; EL-KASSAS et al., 2021), recuperação de informações multilíngue e entre idiomas (SHARMA; MITTAL, 2018; LITSCHKO et al., 2018), reconhecimento de voz (DONG et al., 2018; NASSIF et al., 2019), resposta a perguntas (SOARES; PARREIRAS, 2020; VAKULENKO et al., 2021) e geração de textos (RADFORD et al., 2018; ZHU et al., 2018) são exemplos de aplicações de Processamento de Linguagem Natural.

2.2 APRENDIZADO DE MÁQUINA SUPERVISIONADO

O aprendizado supervisionado, geralmente conhecido como aprendizado de máquina supervisionado, é uma subcategoria de inteligência artificial e aprendizado de máquina. O aprendizado supervisionado pode ser definido como o uso de conjuntos de dados rotulados para treinar algoritmos que classificam os dados com precisão ou preveem resultados. O aprendizado supervisionado é usado para descrever tarefas de previsão porque o objetivo é prever/classificar um resultado específico de interesse (por exemplo, presença ou ausência de um transtorno mental) (JIANG et al., 2020).

Muitos sistemas de aprendizado supervisionado contam com características cuidadosamente construídas para traduzir os dados em um formato que pode ser usado para o aprendizado. Em uma tarefa como a busca de palavras, identificar o radical de cada palavra pode ser útil para que um sistema possa generalizar mais rapidamente em frases relacionadas, como 'amor', 'amar', 'amante' e 'eu te amo'. Uma característica feita à mão, como um dicionário que mapeia cada palavra em uma única forma de radical, pode ser útil na resolução do problema citado (EISENSTEIN, 2018). Os problemas de aprendizado supervisionado podem ser organizados em problemas de regressão e classificação (ENGELEN; HOOS, 2020) (Figura 2.1):

- Classificação: quando a variável de saída é uma categoria, como "vermelho" ou "azul", ou "doença" e "sem doença", um problema é tido como sendo de classificação. O classificador reconhece certas entidades no conjunto de dados e faz suposições sobre como essas entidades devem ser rotuladas ou definidas.
- Regressão: quando a variável de saída é um valor real, como "dólares" ou "peso", então tem-se um problema de regressão. Para explorar a relação entre variáveis dependentes e independentes, a regressão é usada.

Na Figura 2.1, tem-se um problema de classificação à esquerda. Pode-se ver que há bolas de duas cores representando duas classes existentes. A solução consistiu de encontrar uma forma de separar as duas classes, de modo que houvesse o menor erro possível. Já do lado direito da Figura 2.1, tem-se um problema de regressão. A solução consiste de encontrar uma reta que melhor represente os dados.

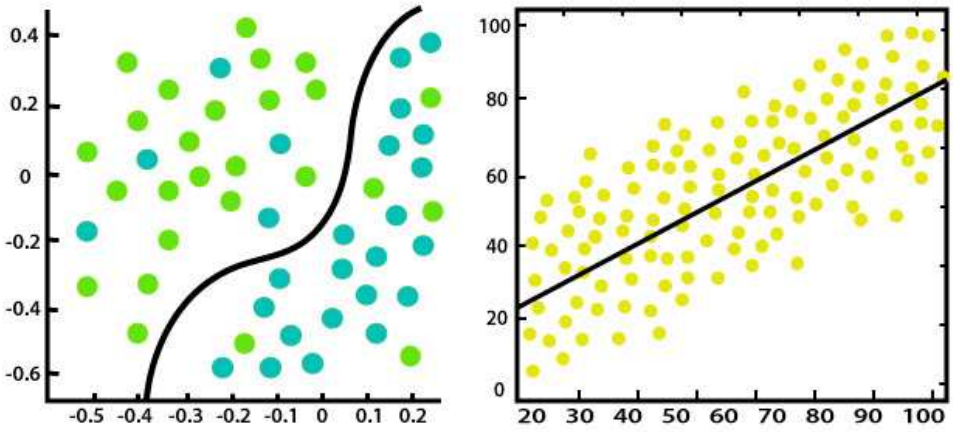

Figura 2.1 – Exemplos de problemas de classificação e regressão

Fonte: Elaborado pelo autor, 2022.

2.2.1 Redes Neurais Profundas

A história das Redes Neurais (RNs) teve início na década de 1940, quando McCulloch e Pitts criaram o primeiro modelo matemático de uma rede neural inspirado no funcionamento do cérebro humano (MCCULLOCH; PITTS, 1943). A partir desse modelo elementar, a capacidade e as aplicações de RNs tiveram um enorme progresso até o final do século passado. Hoje, as RNs são o maior e mais desenvolvido ramo da IA, obtendo maior sucesso na busca por criar inteligência semelhante à humana em máquinas. (IBA; NOMAN, 2020)

Uma rede neural *feedforward* também chamada de multicamadas *perceptrons* (do inglês, *Multi Layer Perceptrons* - MLPs) consiste em neurônios organizados em camadas. Como pode-se ver na Figura 2.2, a camadas à esquerda é chamada camada de entrada, a camada à direita é chamada de camada de saída, e as camadas intermediárias são chamadas de camadas escondidas. Um neurônio em uma determinada camada está conectado a todos ou a um subconjunto de neurônios na camada subsequente. (IBA; NOMAN, 2020)

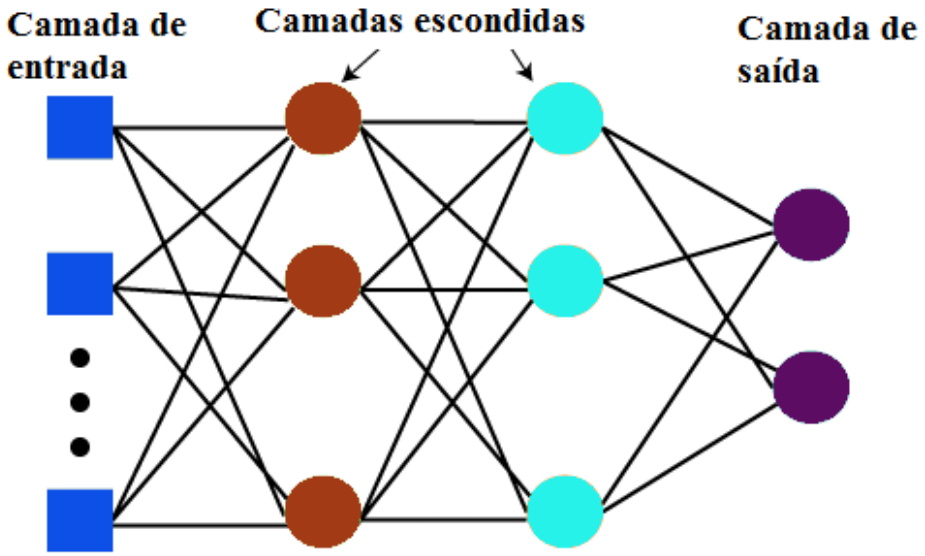

Figura 2.2 – Estrutura de uma MLP com duas camadas escondidas.

Fonte: Elaborado pelo autor, 2022

A ideia central do aprendizado profundo está na exploração das muitas camadas de processamento de informações para extração e transformação de características úteis para aprendizado supervisionado ou não supervisionado. Portanto, é uma extensão lógica das MLPs clássicas. Assim, redes neurais profundas podem utilizar centenas e até milhares de camadas que aprendem uma hierarquia de representações (CICHY; KAISER, 2019).

Os *transformers*, uma estrutura de redes neurais profundas que tem ganhado relevância no contexto de PLN, introduziram um mecanismo de atenção que processa toda a entrada de texto simultaneamente para aprender relações contextuais entre palavras (ou sub-palavras). Um *transformer* inclui duas partes - um codificador que lê a entrada de texto e gera uma representação dela (por exemplo, um vetor para cada palavra) e um decodificador que produz o texto traduzido dessa representação (VASWANI et al., 2017). A Figura 2.3 mostra a arquitetura de um *transformer*.

Do lado esquerdo da Figura 2.3, tem-se o codificador, que é composto por seis camadas iguais. Cada camada tem duas subcamadas. A primeira delas é um mecanismo *multi-head attention*, e a segunda é uma rede *feed-forward*. A saída de cada subcamada é normalizada e os resíduos são tratados. A primeira camada ajuda o codificador a ver outras palavras na frase de entrada à medida que codifica uma palavra específica. Já do lado direto, tem-se o decodificador, também composto por seis camadas iguais.

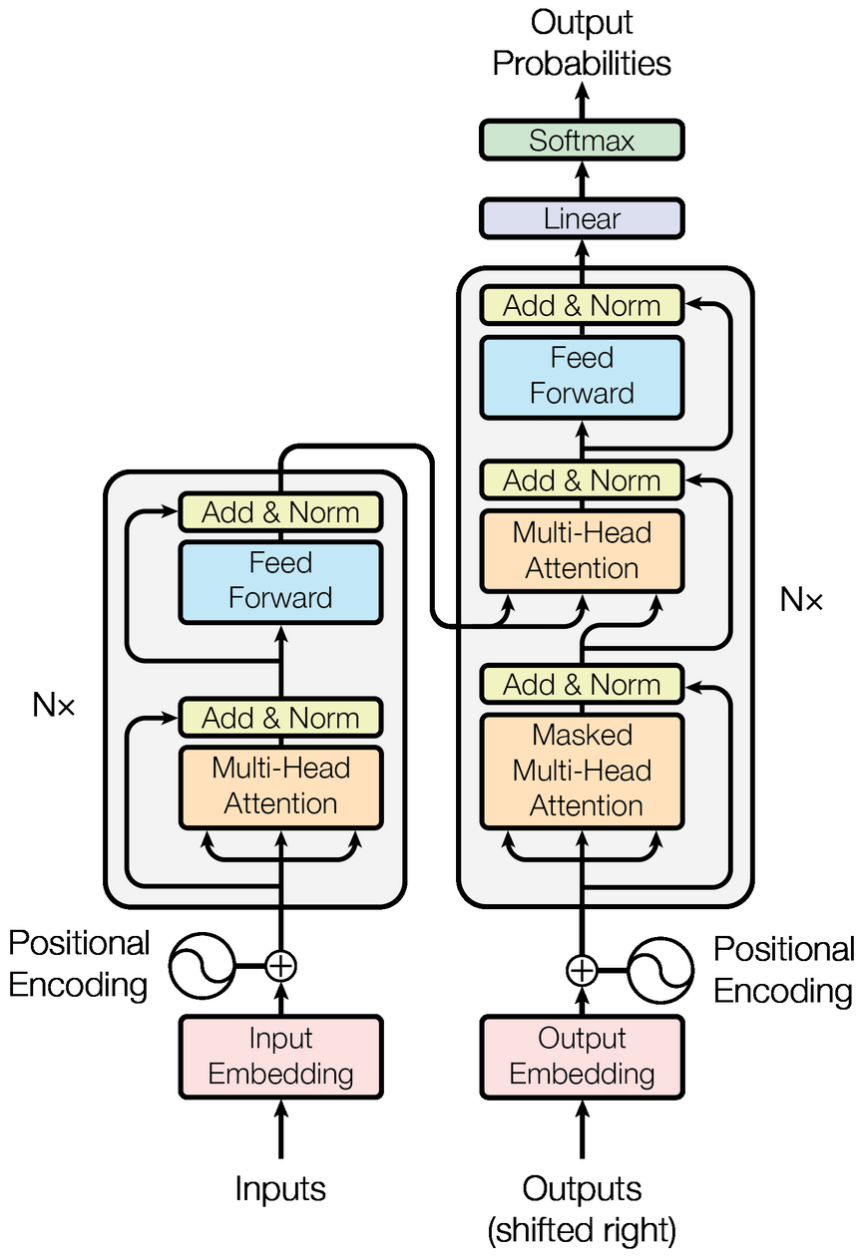

Figura 2.3 – Arquitetura de um transformer.

Fonte: Extraído de Vaswani et al. (2017)

Além das duas subcamadas presentes no codificador, o decodificador tem uma terceira subcamada, com um mecanismo *multi-head attention* sobre a saída do codificador. Esta última camada ajuda o decodificador a se concentrar em partes relevantes da sentença de entrada. Da mesma forma como no codificador, são empregadas conexões residuais e normalizações.

A subcamada de auto-atenção é modificada para prevenir repetições de posições nos *embeddings* - a serem melhor discutidos na próxima subseção - na etapa de decodificação. Isto combinado com o fato de que os *embeddings* de saída são deslocados em uma posição,

garante que as previsões para a posição i possam depender apenas das saídas conhecidas em posições menores que i (VASWANI et al., 2017).

Uma função de atenção pode ser descrita como o mapeamento de uma consulta e um conjunto de pares chave-valor para uma saída, onde a consulta, as chaves, os valores e a saída são todos vetores. Para cada palavra em uma sentença, são criados vetores de chaves, valores e consultas. A saída é calculada como uma soma ponderada dos valores, onde o peso atribuído a cada valor é calculado por uma função de compatibilidade da consulta com a chave correspondente.

O próximo passo no cálculo da auto-atenção é calcular uma pontuação. É necessário pontuar cada palavra da sentença de entrada contra a palavra que está sendo processada. A pontuação determina quanto foco colocar em outras partes da frase de entrada, à medida em que codifica-se uma palavra em uma determinada posição.

A pontuação é calculada tomando o produto escalar do vetor de consulta com o vetor chave da respectiva palavra que estamos pontuando. Então, se a auto-atenção para a palavra na posição 1 estiver sendo processada, a primeira pontuação seria o produto escalar de q1 e k1. A segunda pontuação seria o produto escalar de q1 e k2, e assim por diante (VASWANI et al., 2017; ALAMMAR, 2018).

A próxima etapa consiste em dividir as pontuações por 8 (a raiz quadrada da dimensão dos principais vetores usados em (VASWANI et al., 2017) – 64, o que leva a gradientes mais estáveis), então o resultado é passado por meio de uma operação *softmax*. O *softmax* normaliza as pontuações para que sejam todas positivas e somam 1. Esta pontuação *softmax* determina o quanto cada palavra está expressa nesta posição. A palavra nesta posição terá a pontuação *softmax* mais alta, mas às vezes é interessante atender a outra palavra relevante para a palavra atual.

O quinto passo consiste da multiplicação de cada vetor de valor pela pontuação *softmax*. O principal alvo nesta etapa é manter intactos os valores da(s) palavra(s) que se deseja ter um foco maior e desprezar palavras irrelevantes (multiplicando-as por valores como 0,001, por exemplo). Após isto, os vetores de valor ponderado são somados. Isso produz a saída da camada de auto-atenção nesta posição (para a primeira palavra). O vetor resultante é aquele que será enviado para a rede neural *feed-forward* (ALAMMAR, 2018).

Em 2018, o BERT (DEVLIN et al., 2018) - *Bidirectional Encoder Representations from Transformers* - trouxe algumas modificações na arquitetura original de *transformers*. Ele usa o codificador para aprender um modelo de linguagem mascarado, descartando

algumas das palavras e, em seguida, tentando prevê-las, permitindo que use todo o contexto, ou seja, palavras à esquerda e à direita de uma palavra mascarada. Essa tarefa é chamada de Modelagem de Linguagem Mascarada - *Masked Language Modeling* (MLM).

A segunda tarefa na qual o BERT foi pré-treinado é chamada de Predição da Próxima Sentença - *Next Sentence Prediction* (NSP). Para esta tarefa, o BERT foi alimentado com pares de frases. Em metade dos casos, a segunda frase era uma continuação da primeira, e na outra metade a segunda frase era um trecho de texto selecionado aleatoriamente no texto de treinamento. O BERT foi treinado utilizando uma base de dados da Wikipédia - contendo cerca de 800 milhões de palavras - e uma base de dados chamada *BookCorpus* - composta de cerca de 11.000 livros gratuitos disponibilizados na Internet, totalizando quase 1 bilhão de palavras. (DEVLIN et al., 2019)

Assim, o BERT foi pré-treinado pelo Google em corpora de texto muito grandes, e é possível utilizar sua compreensão de linguagem. Com um modelo pré-treinado, basta fazer um ajuste para alguma aplicação (classificação, reconhecimento de entidade, resposta a perguntas, etc.). Isso permite que sejam alcançados resultados altamente precisos em uma tarefa com um pequeno esforço computacional.

Embora o BERT tenha sido treinado em mais de 100 idiomas, não foi otimizado para modelos multilíngues - a maior parte do vocabulário não é compartilhada entre os idiomas e, portanto, o conhecimento compartilhado é limitado. Para lidar com isso, Lample e Conneau (2019) propuseram o XLM, fazendo duas grandes modificações na arquitetura do BERT. Primeiro, em vez de usar palavras ou caracteres como entrada do modelo, ele usa Codificação de Pares de *Bytes* (BPE) que divide a entrada nas sub-palavras mais comuns em todos os idiomas, aumentando assim o vocabulário compartilhado entre os idiomas.

A segunda alteração do XLM em relação ao BERT consistiu em passar como entrada para o treinamento os mesmos textos em dois idiomas diferentes (traduzidos manualmente), enquanto no BERT cada amostra de entrada é construída a partir de um único idioma. Como no BERT, o objetivo do modelo é realizar a predição dos *tokens* mascarados, no entanto, com a nova arquitetura, o modelo pode usar o contexto de um idioma para prever *tokens* no outro, já que palavras diferentes são palavras mascaradas em cada idioma.

O XLM também recebe um identificador do idioma e a ordem dos *tokens* em cada idioma, ou seja, a codificação posicional, separadamente. Os novos metadados ajudam o modelo a aprender a relação entre *tokens* relacionados em diferentes idiomas. A Figura 2.4 exibe a comparação entre o treinamento realizado pelo XLM com o do BERT. A parte

superior da figura refere-se à tarefa de treinamento utilizada no BERT (MLM), enquanto que a parte inferior refere-se à tarefa utilizada no XLM (TLM - *Translation Language Model*). Vê-se que em TLM, há idiomas diferentes sendo inseridos como entrada em um mesmo momento.

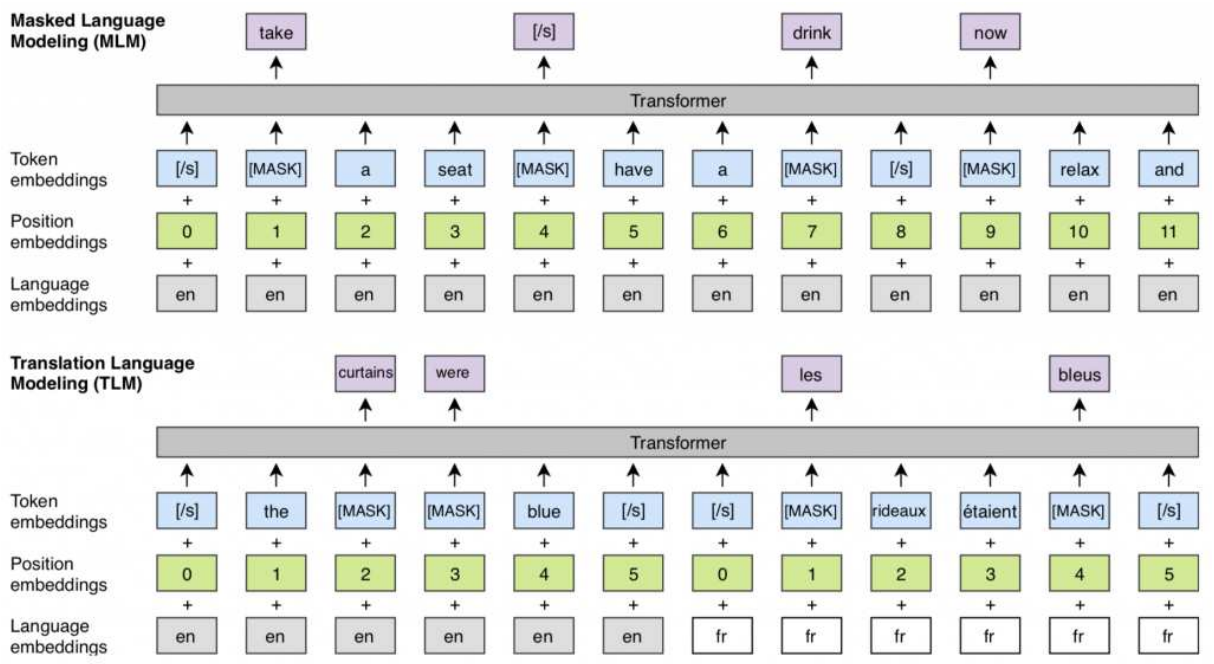

Figura 2.4 – Comparação entre treinamento do XLM e do BERT.

Fonte: Extraído de Lample e Conneau (2019)

O XLM-R (CONNEAU et al., 2020) é uma variação do XLM e possui versões *base* e *grande*. A versão base contém aproximadamente 270 milhões de parâmetros, com 12 camadas, 768 estados ocultos, 3072 estados ocultos *feed-forward* e 8 cabeças; enquanto que a versão grande contém 550 milhões de parâmetros, com 24 camadas, 1024 estados ocultos, 4096 estados ocultos *feed-forward* e 16 cabeças.

O XLM-R usa o estado oculto final h do primeiro *token* [CLS] (um *token* especial, que armazena o resultado da execução dos mecanismos de atenção sobre o texto passado como entrada) como a representação de toda a sequência para tarefas de classificação de texto. Para prever a probabilidade do rótulo c, um classificador *softmax* simples é adicionado ao topo do XLM-R, conforme exibido na Equação 1, onde W é a matriz de parâmetros específicos da tarefa (RANASINGHE; ZAMPIERI, 2020).

$$p(c|h) = softmax(Wh) \tag{1}$$

onde c = label a ser predito; h = estado oculto final; W = matriz de parâmetros específicos da tarefa.

Ao maximizar o logaritmo da probabilidade do rótulo correto, todos os parâmetros do XLM-R são ajustados, bem como o vetor de pesos W. Pode-se ver um diagrama da arquitetura do XLM-R para classificação na Figura 2.5.

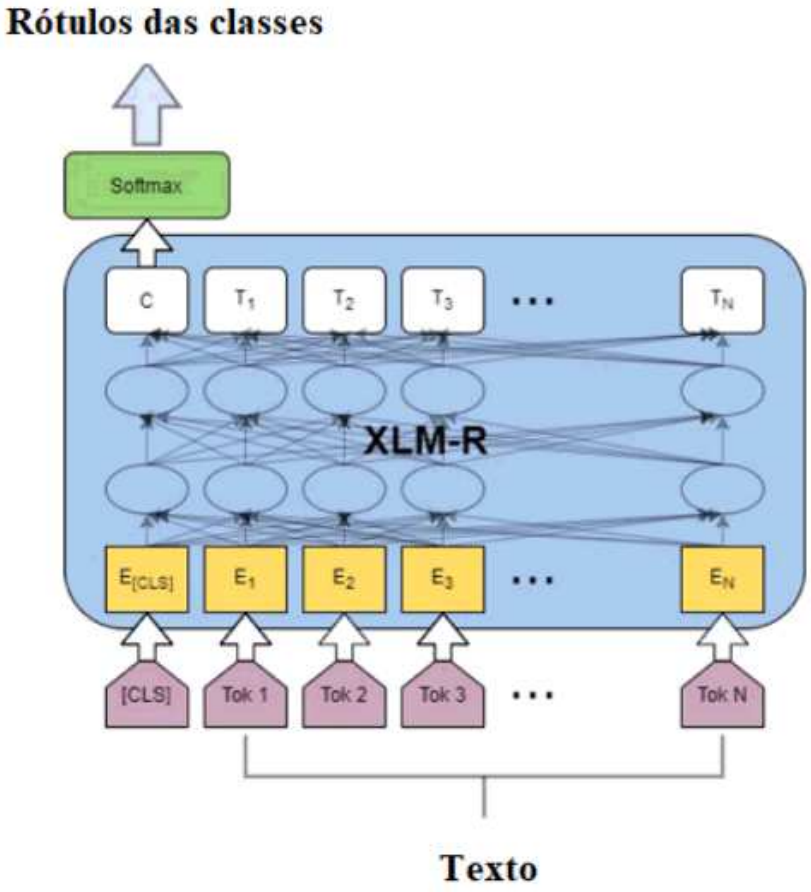

Figura 2.5 – Arquitetura do XLM-R.

Fonte: Adaptado de Ranasinghe e Zampieri (2020)

Além do XLM, foram desenvolvidos vários outros *transformers* baseados no BERT. ALBERT (LAN et al., 2019), que também foi desenvolvido pela empresa Google, alcançou pontuações de *benchmark* mais altas do que BERT, XLNet (YANG et al., 2019) e RoBERTa (LIU et al., 2019), mas apenas em seu maior tamanho - "ALBERT-xxlarge". O ALBERT-base, por exemplo, geralmente tem o mesmo desempenho ou pior do que o BERT-base.

Os autores de RoBERTa e XLNet não consideraram a tarefa NSP como confiável e optaram por removê-la completamente de seu pré-treinamento. ELECTRA (CLARK et al., 2019) é particularmente notável por afirmar que supera o BERT em suas configurações menores (ou seja, BERT-base), e não apenas nos tamanhos maiores.

2.2.1.1 *Word Embeddings*

Em pesquisas de PLN, o desenvolvimento de abordagens para criar representação de palavras é imprescindível. O Modelo de Espaço Vetorial, geralmente atribuído a Salton (1975) e originário da comunidade de Recuperação da Informação (do inglês, *Information Retrieval* - IR), é o modelo mais bem sucedido e influente para codificar palavras e documentos como vetores (WANG et al., 2020).

As primeiras abordagens para criar *word embeddings* consistiam de representações *one-hot*, em que cada palavra é representada como um vetor de tamanho de vocabulário com apenas uma entrada sendo igual a um, e as outras iguais a zero. Por sua simplicidade, representações *one-hot* foram amplamente adotadas como base da PLN e RI (SCHÜTZE et al., 2008).

Contudo, representações *one-hot* não levam em conta as relações semânticas entre as palavras, além destas representações gerarem vetores com muitos zeros. Uma evolução dessas representações é a criação de *word embeddings*. *Embeddings* modelam o contexto e a relação entre a palavra alvo e suas palavras de contexto por meio de redes neurais e contém informações semânticas frutíferas, que podem ser obtidas ao treinar modelos de linguagem, ou ao construir especificamente redes neurais para gerá-los (WANG et al., 2020).

Na Figura 2.6, tem-se uma visualização de *embeddings* gerados em um espaço vetorial. No primeiro exemplo da esquerda, é possível observar relações de gênero entre os *embeddings*, de forma que as palavras 'homem' e 'mulher' possuem a mesma distância das palavras 'rei' e 'rainha'. Já no segundo exemplo, vê-se algo semelhante, porém a relação encontrada é concernente a tempos e formas verbais. No último exemplo, é possível observar a similaridade das distâncias das representações geradas dos países para as respectivas capitais.

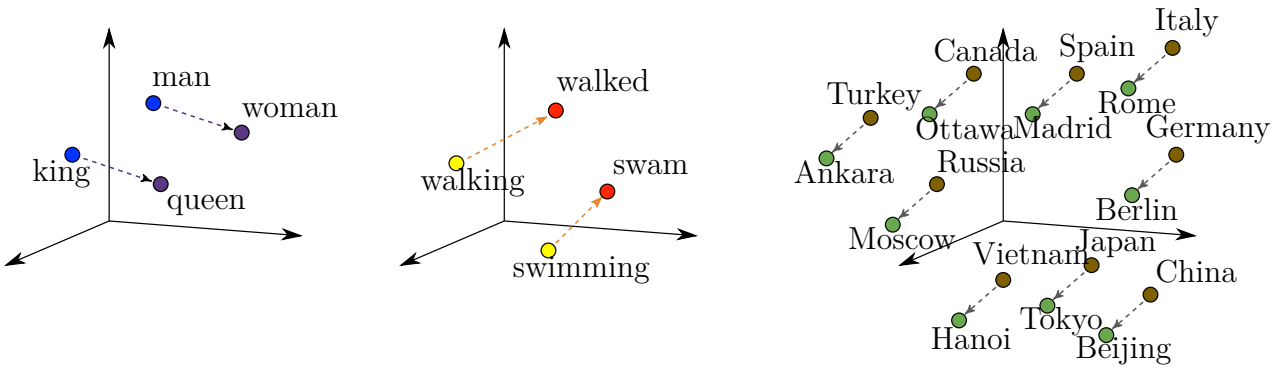

Figura 2.6 – Visualização de *embeddings* em um espaço vetorial.

Fonte: Extraído de Google (2022)

Uma das abordagens pioneiras na criação de *embeddings* foi o *Word2Vec* (MIKOLOV et al., 2013). *Word2Vec* é eficiente e eficaz para aprender representações de palavras a partir de corpus, que implementa dois modelos: *CBOW* e *Skip-gram*. Esses dois modelos podem capturar efetivamente a semântica das palavras e facilmente transferí-las para outras tarefas posteriores.

As abordagens de criação de *embeddings* existentes até então atribuíam um vetor distinto a cada palavra, considerando assim as palavras como sendo atômicas. Para resolver esta limitação, o *fastText* (JOULIN et al., 2017), um categorizador baseado em *CBOW* e de código aberto criado pela *Facebook AI Research*, usa informações de n-gram de sub-palavras, que podem obter a relação de ordem entre os caracteres e capturar melhor a semântica interna entre as palavras.

Peters et al. (2018) propuseram um método de representação de palavras profundamente contextualizado que leva em conta a natureza complexa do uso de palavras em termos de semântica e gramática, e as possíveis mudanças contextuais destas palavras. A ideia é treinar um modelo LSTM bidirecional com um grande corpus, com o modelo de linguagem como alvo, e então usar o LSTM para gerar a representação das palavras. O nome desta abordagem é ELMo (do inglês, *Embeddings from Language Models*).

Cada MLPT possui um módulo para a criação de *embeddings*. O GPT (do inglês, *Generative Pre-Training*) (RADFORD et al., 2018), por exemplo, usa um modelo de linguagem unidirecional. Já o BERT (DEVLIN et al., 2019) usa a técnica de *transformer* bidirecional, que pode efetivamente explorar a informação semântica profunda de uma frase. A vantagem disso é que as características aprendidas podem unir o contexto em ambas as direções das sentenças de entrada.

2.2.2 Transferência de Aprendizagem

Antes de criar uma previsão realista, modelos de aprendizado profundo (*deep learning*) grandes e bem-sucedidos exigem treinamento com dezenas ou até milhões de dados. O treinamento é bastante caro em termos de tempo e recursos. A transferência de aprendizagem (*transfer learning*) é uma abordagem de aprendizado profundo (e aprendizado de máquina) em que o conhecimento é passado de um modelo para o outro. Podemos usar a transferência de aprendizagem para resolver uma tarefa específica aplicando todo ou parte de um modelo que já foi treinado em uma tarefa diversa (WEISS et al., 2016).

A transferência de aprendizagem é uma técnica para melhorar um modelo de um domínio, transferindo informações de outro. A ideia de transferência de aprendizagem

pode ter se originado na psicologia educacional. De acordo com a teoria de transferência de generalização do psicólogo C. H. Judd, aprender a transferir é o resultado da generalização da experiência. Desde que uma pessoa generalize sua experiência, é viável passá-la de um cenário para outro. De acordo com essa hipótese, deve haver uma ligação entre dois processos de aprendizagem para que a transferência ocorra. Alguém que aprendeu a tocar violino pode aprender a tocar piano mais rápido do que outros, porque violino e piano são instrumentos musicais com algum conhecimento comum (ZHUANG et al., 2020).

Usando modelos de aprendizado profundo, existem três maneiras básicas de fazer a transferência de aprendizagem (WEISS et al., 2016; ZHUANG et al., 2020). A primeira delas é aplicando modelos pré-treinados diretamente a uma tarefa de destino. Esta é a técnica mais simples de resolver uma tarefa de destino aplicando um modelo de uma tarefa de origem. BERT (DEVLIN et al., 2018), GloVe (PENNINGTON et al., 2014) e outros modelos pré-treinados são usados diretamente.

A segunda forma é usando modelos pré-treinados para extração de características. Em vez de usar a abordagem de ponta a ponta do modelo como no exemplo anterior, pode-se considerar a rede neural profunda pré-treinada como um extrator de características removendo-se a última camada de saída totalmente conectada. Quando o conjunto de dados da tarefa de destino é pequeno, a técnica de extração de características é a melhor opção.

A última forma é ajustando as últimas camadas do modelo pré-treinado. Pode-se dar um passo adiante, não apenas treinando o classificador de saída, mas também ajustando-se os pesos em algumas das camadas do modelo pré-treinado. Quando o conjunto de dados da tarefa de destino é grande e tem um domínio semelhante ao conjunto de dados de origem, a técnica de ajuste fino (*fine-tuning*) funciona melhor.

Há uma variedade de exemplos disponíveis de aplicações de processamento de linguagem natural que usam transferência de aprendizagem, incluindo classificação de sentimento, classificação de texto, detecção de e-mail de spam, dentre outros. A classificação de imagem e a classificação de vídeo são duas outras aplicações de transferência de aprendizagem bem representadas. (WEISS et al., 2016; ZHUANG et al., 2020).

2.2.3 Métricas de Avaliação

Considerando-se que a detecção de discurso de ódio é um problema de classificação de texto, são utilizadas métricas de avaliação para mensurar a eficácia do classificador criado. Na análise desta eficácia, existem quatro métricas principais a serem consideradas (GORDON; KOCHEN, 1989). Para cada entrada, um classificador binário retorna um

resultado positivo ou negativo. Como resultado, pode-se computar para um conjunto de instâncias fornecidas ao classificador:

- TP (verdadeiros positivos), que é o número de instâncias marcadas corretamente como positivas pelo classificador;
- FP (falsos positivos), que é o número de instâncias erroneamente marcadas como positivas pelo classificador;
- TN (verdadeiros negativos), que é o número de instâncias marcadas corretamente como negativas pelo classificador;
- FN (falsos negativos), que é o número de instâncias marcadas incorretamente como negativas pelo classificador.

A acurácia, precisão, revocação e medida F1 são as quatro métricas comumente usadas para avaliar o desempenho de um classificador binário (GOUTTE; GAUSSIER, 2005). Também define-se

$$P = TP + FN \tag{2}$$

como o número de casos positivos e

$$N = TN + FP \tag{3}$$

como o número de exemplos negativos.

A acurácia é definida como a proporção de casos devidamente rotulados pelo classificador, ou seja,

$$Acurácia = \frac{(TN + TP)}{(N + P)} \tag{4}$$

A precisão é definida como

$$Precisão = \frac{(TP)}{(FP + TP)} \tag{5}$$

, que é a proporção de instâncias marcadas corretamente como positivas de todas as instâncias designadas como positivas. No contexto da recuperação de informação, essa é a porcentagem de resultados retornados que são genuinamente relevantes.

A fração de casos marcados como positivos de todas as instâncias positivas é chamada de revocação (*recall* em inglês) e é calculada conforme a Equação 6:

$$Revocação = \frac{(TP)}{(FN + TP)} \tag{6}$$

No contexto de recuperação de informação, esta é a percentagem de resultados recuperados de todos os resultados no conjunto de dados. Medida F1 é a média harmônica de precisão e revocação, calculada da seguinte forma:

$$Medida\text{-}F1 = \frac{(2 \times Precisão \times Revocação)}{(Precisão + Revocação)} \tag{7}$$

Quando se deseja encontrar um classificador com precisão e revocação balanceados, em vez de alta precisão e revocação baixa ou vice-versa, utiliza-se essa métrica.

Embora a acurácia pareça ser a métrica mais clara e para resumir o desempenho de um classificador de forma adequada à primeira vista, não é a escolha ideal ao se trabalhar com um conjunto de dados não balanceado. Por exemplo, se há um conjunto de dados em que 20% dos casos são positivos e 80% negativos e, se um classificador foi desenvolvido com todos os casos marcados como negativos, ele teria uma acurácia de 80% neste conjunto de dados e pareceria ter um bom desempenho. Porém, precisão, revocação e medida F1 iriam refletir de forma mais acertada o desempenho do classificador.

2.3 DETECÇÃO DE DISCURSO DE ÓDIO

O discurso de ódio é um crime que tem aumentado nos últimos anos, não apenas em contatos pessoais, mas também de forma online. Por um lado, devido ao anonimato proporcionado pela Internet e pelas redes sociais em particular, as pessoas têm maior probabilidade de se envolver em comportamentos hostis. Por outro lado, o desejo das pessoas de expressar seus pensamentos online aumentou. O discurso de ódio está em constante expansão devido ao desenvolvimento de material digital gerado por usuários, especialmente nas redes de mídia social. O interesse pela identificação online do discurso de ódio e, principalmente, pela automação dessa atividade, tem aumentado constantemente nos últimos anos (SCHMIDT; WIEGAND, 2017).

Segundo Fortuna e Nunes (2018), “O discurso do ódio é a linguagem que ataca ou diminui, que incita à violência ou ao ódio contra grupos, com base em características específicas como aparência física, religião, descendência, nacionalidade ou origem étnica, orientação sexual, identidade de gênero ou outro, e pode ocorrer com diferentes estilos

linguísticos, mesmo em formas sutis ou quando se usa o humor". Se a agressão pode se manifestar física e explicitamente, também pode se manifestar sutilmente. É a situação em que os estereótipos são reforçados, permitindo justificar discriminações e preconceitos desfavoráveis a determinados grupos. Como resultado, todas as formas sutis de preconceito, incluindo piadas, devem ser classificadas como discurso de ódio. Isso ocorre porque essa forma de piada revela relações entre os grupos de curingas e os grupos-alvo das piadas, bem como relações raciais e estereótipos (KUIPERS; ENT, 2016).

Entre todas as redes sociais, o Twitter espalha a maioria das mensagens contendo comentários de ódio (HEWITT et al., 2016; FORTUNA; NUNES, 2018). Uma vez que o Twitter confia em sua comunidade para relatar *tweets* abusivos e porque existe uma equipe dedicada a revisar e remover manualmente esses *tweets*, a remoção de mensagens contendo discurso de ódio é uma tarefa complexa (HEWITT et al., 2016). Além disso, uma comissão da União Europeia acusou o Twitter de não lidar com a remoção do discurso de ódio em sua plataforma (KOTTASOVá, 2017).

A detecção de discurso de ódio também é relevante no contexto de eventos. Um número maior de mensagens de ódio em um curto período pode mostrar algum comportamento suspeito em uma comunidade. Essas informações podem contornar incidentes como violência racial, ataques terroristas ou outros crimes antes que eles aconteçam, fornecendo assim passos para uma governança antecipada (SCHMIDT; WIEGAND, 2017).

É importante ressaltar que o conceito de discurso de ódio é diferente de outros conceitos do senso comum. No Quadro 2.1, são mostrados alguns conceitos destacando as diferenças desses conceitos para o discurso de ódio (FORTUNA; NUNES, 2018).

Detecção de discurso de ódio é uma tarefa repleta de desafios. A divergência de entendimento quanto à rotulagem de discurso de ódio por humanos, mostra que esta classificação é ainda mais difícil para modelos computacionais. Anotar discurso de ódio é uma tarefa demorada: há muito mais comentários bons ou neutros do que comentários de ódio em qualquer amostra de dados aleatória e, portanto, muitos comentários devem ser anotados para que seja possível encontrar um número considerável de ocorrências de discurso de ódio (SCHMIDT; WIEGAND, 2017; FORTUNA; NUNES, 2018).

Apesar da natureza ofensiva do discurso de ódio, a linguagem abusiva pode ser muito fluente e correta, e figuras de linguagem - como o sarcasmo - podem ser usadas. Outro ponto importante no tocante a detecção de discurso de ódio é a presença de vieses nas bases de dados existentes. De acordo com Badjatyia et al. (2019), é comum a existência de dados rotulados como odiosos pelo simples fato de conter uma palavra de cunho negativo no texto (ex: Aquele árabe matou as plantas). Sap et al. (2019) tentaram identificar

vieses presentes em conjuntos de dados existentes. A princípio, foram analisados os *tweets* classificados como odiosos, e foi visto que a maior parte deles continha algum termo do dialeto *African American English* e foram falsamente identificados.

Quadro 2.1 – Conceitos relacionados à temática de discurso de ódio.

Conceito	**Definição**	**Diferença para discurso de ódio**
Ofensa	Expressão de hostilidade para com uma pessoa.	O discurso de ódio inclui ofensas, mas é direcionado a um grupo específico.
Cyberbullying	Ato agressivo e intencional vindo de uma pessoa ou grupo contra uma vítima que não consegue se defender facilmente.	Uma das premissas do discurso de ódio é que ele precisa ser dirigido a um grupo de pessoas, não a um indivíduo específico.
Linguagem obscena	Envolve o uso de palavrões abusivos, intimidantes e palavrões em discussões.	O discurso de ódio pode ou não conter termos obscenos, e no discurso de ódio o alvo é sempre um grupo.
Extremismo / Radicalização	Envolve grupos que promovem a violência, considerando alguns outros grupos como inferiores e incitando a segmentação da população.	O discurso extremista e radical geralmente envolve questões como guerra, religião, enquanto o discurso de ódio pode ocorrer de maneiras mais sutis.

Fonte: Elaborado pelo autor, 2022.

Pode-se afirmar que o processo de detecção de discurso de ódio tem cinco etapas principais: extração de dados, pré-processamento de dados, seleção de características, classificação e avaliação (POLETTO et al., 2021). Na extração de dados, os textos a serem utilizados na pesquisa são coletados. Segundo Fortuna e Nunes (2018), a fonte de dados mais utilizada em trabalhos de pesquisa é o Twitter, que possui uma API aberta e de fácil obtenção de conjuntos de dados.

Na etapa de pré-processamento, os textos são tratados e procedimentos como lematização, que reduz uma palavra à sua forma base e agrupa diferentes formas da mesma palavra; remoção de *stopwords*, que consiste na remoção de palavras com muitas ocorrências no texto, e que não auxiliam na tarefa de classificação (ex: do, uma, dentre outras); *tokenização*, que consiste na segmentação de palavras de uma sentença, podem ser realizados.

A etapa de seleção de características é uma das mais importantes, considerando que o desempenho do modelo depende de quão boas foram as características escolhidas. Exemplos de características são n-gramas, que consiste de localizar agrupamentos de

palavras; *embeddings*, que são representações vetoriais de textos; e *pos-tagging*, que consiste de rotular cada palavra em uma sentença com a sua respectiva classificação gramatical.

Posteriormente, tem-se a etapa de classificação. Aqui, soluções de aprendizado de máquina (como regressão logística, árvores de decisão), aprendizado profundo (CNN, *transformers*) ou combinações de técnicas podem ser usadas. A etapa final consiste na avaliação da solução elaborada. Para tanto, utilizam-se métricas (como medida F1, AUC) que indicam se as predições realizadas pela solução foram boas ou não.

2.3.1 Aspectos Jurídicos no Tocante a Discurso de Ódio

Muitos países exigem que os proprietários de sites ajam rapidamente e removam as ocorrências de discurso de ódio assim que descobertas. Para ser mais específico, a União Europeia abordou o discurso de ódio em várias ocasiões, incluindo na Convenção Internacional para a Eliminação da Discriminação Racial (1965), no Pacto Internacional sobre Direitos Civis e Políticos (1996) e, mais recentemente, nas Recomendações da Comissão Europeia contra o Racismo e a Intolerância (ECRI) (SRBA et al., 2021).

O Plano de Ação Rabat das Nações Unidas (COUNCIL, 2013), que estabelece instruções para diferenciar entre liberdade de expressão e discurso de ódio, recomenda discriminar entre três tipos de expressão: “expressão que constitui um crime; expressão que não é punível, mas pode justificar ação civil ou sanções administrativas; expressão que não dá origem a sanções penais, civis ou administrativas, mas que causa apreensão em termos de tolerância, civilidade e apreço pelos direitos dos outros ”.

Na Constituição Federal, em seu art. 5, nos incisos IV, V, VI e X está dito: “Todos são iguais perante a lei, sem distinção de qualquer natureza, garantindo-se aos brasileiros e aos estrangeiros residentes no País a inviolabilidade do direito à vida, à liberdade, à igualdade, à segurança e à propriedade, nos termos seguintes:

- IV - é livre a manifestação do pensamento, sendo vedado o anonimato;
- V - é assegurado o direito de resposta, proporcional ao agravo, além da indenização por dano material, moral ou à imagem;
- VI - é inviolável a liberdade de consciência e de crença, sendo assegurado o livre exercício dos cultos religiosos e garantida, na forma da lei, a proteção aos locais de culto e a suas liturgias;
- X - são invioláveis a intimidade, a vida privada, a honra e a imagem das pessoas, assegurado o direito a indenização pelo dano material ou moral decorrente de sua violação;”

Além disso, é importante citar o Marco Civil da Internet, em seu art. 2, incisos II e III: “A disciplina do uso da internet no Brasil tem como fundamento o respeito à liberdade de expressão, bem como:

- II - os direitos humanos, o desenvolvimento da personalidade e o exercício da cidadania em meios digitais;
- III - a pluralidade e a diversidade;”

Do ponto de vista do direito legislado, no Brasil ainda não se tem uma legislação específica sobre discurso de ódio. Todavia, há vários dispositivos legais esparsos que podem ser invocados para, de forma indireta e em conjunto, construir um conceito de discurso de ódio. Na Lei Federal Contra o Preconceito 7.716 / 89, tem-se o artigo 20 que proíbe “Praticar, induzir ou incitar discriminação ou preconceito de raça, cor, etnia, religião ou nacionalidade” com pena de reclusão de dois a cinco anos e uma multa. Ainda no referido artigo, o item 2 prescreve: “Se algum dos crimes previstos no caput for praticado por meio de mídia ou publicação de qualquer espécie” (NETO; RODRIGUES, 2021).

Já no Código Penal, no art. 140, caput, tem-se a tipificação de “injuriar alguém, ofendendo-lhe a dignidade ou o decoro”, como crime. No §3º, prevê-se como injúria qualificada quando “na utilização de elementos referentes a raça, cor, etnia, religião, origem ou a condição de pessoa idosa ou portadora de deficiência”. A pena prescrita é de um a três anos de reclusão, além de multa.

Outras normas no tocante ao tema que estão em vigor no direito positivo brasileiro e merecem ser citadas são : art. 13, inc. 5, do Pacto de *San José da Costa Rica*, do art. 20 do Pacto Internacional de Direitos Civis e Políticos (ambos em vigor no Brasil desde 1992), do art. 4º da Convenção Internacional sobre a Eliminação de Todas as Formas de Discriminação Racial (em vigor no Brasil desde 1969), do art. 3º da Lei 2.889/56 (Lei do Genocídio), do art. 26, I, da Lei 12.288/2010 (Estatuto da Igualdade Racial), do art. 1º, VII da Lei 10.466/2002 (sobre infrações penais de repercussão interestadual ou internacional que exigem repressão uniforme), e arts. 57-A a 57-J da Lei 9.504/97 (Lei das Eleições).

É importante levar em consideração que atualmente tramita-se na Câmara dos Deputados o Projeto de Lei 2630/2020, popularmente conhecido como “Lei das *Fake News*”. Este projeto propõe, em seu artigo 3º, que “constitui crime de ódio a ofensa à vida, à integridade corporal, ou à saúde de outrem motivada por preconceito ou discriminação em razão de classe e origem social, condição de migrante, refugiado ou deslocado interno, orientação sexual, identidade e expressão de gênero, idade religião, situação de rua e deficiência”.

Existem diversas discussões no tocante a este projeto: enquanto alguns juristas o defendem como forma de proteção, dando ao poder público a regulação das condutas, outros são totalmente contrários, preocupando-se com uma supressão da pluralidade de pensamento, transformando assim as mídias sociais em entes que se autorregulam, podendo conceder ou retirar direitos dos cidadãos (NETO; RODRIGUES, 2021).

2.4 CRUZAMENTO DE IDIOMAS

O desenvolvimento de soluções baseadas em PLN para idiomas com poucos dados anotados pode proporcionar novas opções de pesquisa. A importância do PLN para idiomas com poucos dados anotados foi demonstrada durante várias crises em partes do mundo, onde as pessoas falam línguas que não são frequentemente abordadas pela comunidade do PLN, como no caso do terremoto haitiano de 2010 (LEWIS, 2010) (com o crioulo haitiano) ou do Surto de Ebola na África Ocidental (com línguas Níger-Congo).

A barreira do idioma permaneceu um sério impedimento para ajudar as pessoas afetadas por desastres, tanto para funcionários de campo, quanto para sistemas de monitoramento de canais de emergência ou mídia social. Mesmo que essas situações graves não ocorram, a comunidade de pesquisa deve se esforçar para democratizar os sistemas baseados em PLN, especialmente porque as tecnologias de informação modernas estão começando a influenciar categorias de pessoas mais marginalizadas social e ambientalmente (BLOMMAERT, 2008).

De acordo com pesquisas publicadas em conferências recentes de PLN, o inglês é o idioma mais investigado e é o único idioma considerado em mais de 60% dos artigos publicados (FORTUNA; NUNES, 2018; POLETTO et al., 2021). Em contraste, outros idiomas - como o crioulo haitiano, citado acima - carecem não apenas da atenção da academia, mas também de uma variedade de recursos, como dados, modelos e ferramentas (PIKULIAK et al., 2021).

Cruzamento de idiomas, ou CLL (*Cross-Lingual Learning*) visa a resolver a escassez em idiomas com poucos dados anotados. Resumindo, a ideia do CLL é usar dados anotados de outros idiomas para criar novos modelos de PLN. Como resultado, o CLL pode auxiliar o desenvolvimento de novos sistemas inteligentes em idiomas com poucos dados anotados. CLL já provou ser eficaz em uma variedade de aplicações de PLN, incluindo tradução automática, análise de sentimento, análise de dependência, reconhecimento de entidade nomeada, análise semântica e análise morfológica, entre outros (PIKULIAK et al., 2021).

Sejam D_T e D_S domínios distintos pertencentes a P(L), sendo L o conjunto de todas os idiomas e P(L) o conjunto das partes de L, tem-se a Proposição 8, (PIKULIAK

et al., 2021):

$$\exists l_a \in D_T \quad \exists l_b \in D_S \quad l_a \neq l_b \tag{8}$$

em que l_a e l_b são idiomas distintos; D_S é o domínio fonte, e D_T é o domínio alvo

De acordo com a Proposição 8, sempre há pelo menos um par de idiomas diferentes entre os quais o conhecimento pode ser transferido durante o treinamento. Qualquer idioma fonte b pode estar presente no D_S. Assim, com a utilização de CLL vê-se que há a utilização de dois idiomas distintos, em que o idioma alvo carece de dados anotados, enquanto que o idioma fonte têm muitos dados e podem ser aproveitados para melhorar os resultados daquele (PIKULIAK et al., 2021).

Pikuliak et al. (2021) investigaram quais aspectos do conhecimento são compartilhados ou transferidos entre os idiomas. Eles criaram quatro categorias principais de CLL, as quais foram chamadas de paradigmas de transferência (Figura 2.7):

- Transferência de rótulo: anotações são transferidas de uma amostra para outra.
- Transferência de característica: é como a transferência de rótulos, mas em vez de rótulos, as características da amostra são transferidas.
- Transferência de parâmetro: Os valores dos parâmetros são transferidos de modelos paramétricos para modelos paramétricos. Isso se traduz essencialmente na transferência de comportamento do modelo.
- Transferência de representação: os valores previstos da representação oculta são transmitidos entre os modelos. O modelo de destino é ensinado a gerar as representações pretendidas.

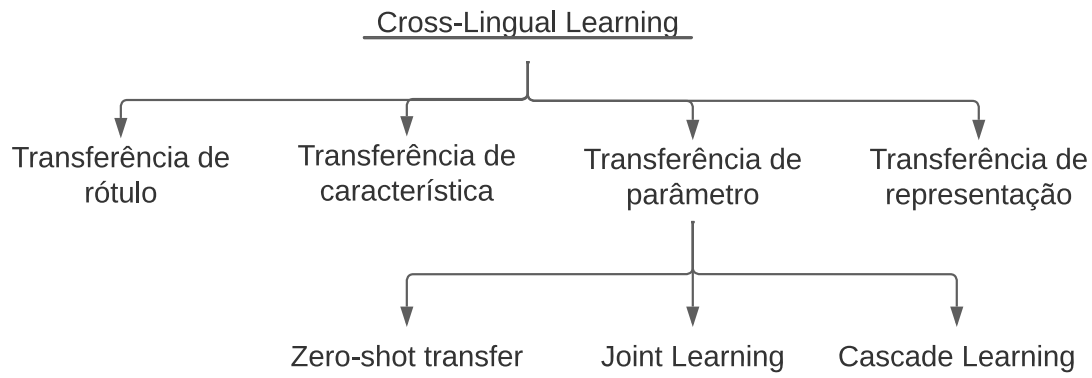

Figura 2.7 – Paradigmas de transferência usando CLL.

Fonte: Adaptado de Pikuliak et al. (2021)

Na Figura 2.7, são mostrados os paradigmas de transferência elencados por Pikuliak et al. (2021). Nesta tese, utilizou-se o paradigma transferência de parâmetros. Os três sub paradigmas vistos na figura (*zero-shot*, *joint learning* e *cascade learning*) foram utilizados na metodologia proposta como estratégias de treinamento. Mais detalhes são encontrados no Capítulo 4.1.

O conhecimento sobre amostras individuais é transferido no caso das transferências de rótulo e de características. O comportamento de um modelo para outro idioma é transferido segundo o paradigma transferência de parâmetros. No sentido de que se transmite conhecimento sobre as características da amostra, a transferência de representação é semelhante à transferência de características. No entanto, em vez de simplesmente transferir as características, ele instrui o modelo de destino a desenvolvê-las.

2.5 TESTES DE SIGNIFICÂNCIA ESTATÍSTICA

De acordo com Dror et al. (2020), existe uma grande dificuldade na comparação de performance de algoritmos baseados em *transformers*. Os motivos apontados pelos autores são a complexidade dos modelos, uma vez que o processo de treinamento não é determinístico pela inserção de *dropouts*, por exemplo, fazendo com que esses modelos não sejam totalmente compreendidos.

Além disso, as funções de perda desses modelos são não convexas, fazendo com que a solução encontrada (como um mínimo local, por exemplo) seja sensível à inicialização aleatória dos pesos e à ordem dos elementos nos vetores passados como entrada. Por fim, tem-se um número grande de hiper-parâmetros - como o número de neurônios, por exemplo

- gerando um enorme espaço de configurações.

A maioria dos testes de significância estatística opera usando valores p, que definem a probabilidade de que sob a hipótese nula, o valor esperado pelo teste seja maior ou igual à diferença observada (ou seja, para um teste unilateral, ou seja, assumimos A ser melhor que B). Em linhas gerais, utilizando valores p podemos descartar uma hipótese nula, concluindo que pode-se prosseguir com a hipótese alternativa. Porém, o valor p não expressa se a hipótese nula é verdadeira (DROR et al., 2019).

Diante das limitações em testes que se baseiam em valores p, e na complexidade de Modelos de Linguagem Pré-Treinados, Dror et al. (2019) introduziram a Ordem Quase Estocástica (do inglês, *Almost Stochastic Order* - ASO), um teste de significância para comparar duas distribuições de performance.

O teste de significância ASO se baseia no conceito de ordem estocástica: podemos comparar duas distribuições e declarar uma como estocasticamente dominante comparando suas funções de distribuição cumulativas.

Suponha-se então que A e B sejam dois algoritmos distintos, e que se deseja comparar os desempenhos de ambos. A função de distribuição acumulada de A é apresentada em vermelho e em verde para B na Figura 2.8. Se a função de A for menor que B para cada x, então sabe-se que o algoritmo A tem um melhor desempenho do que B.

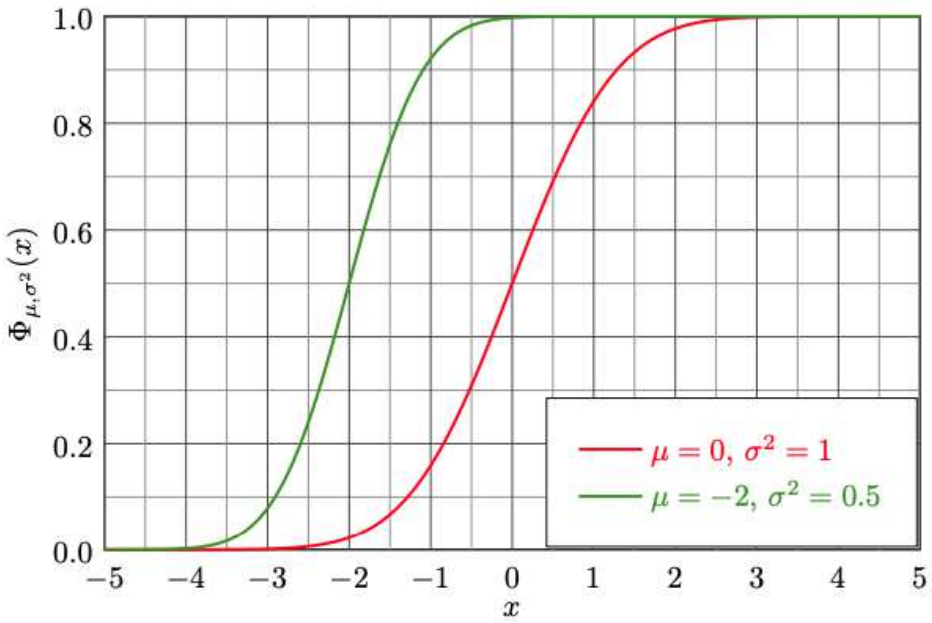

Figura 2.8 – Ordem Estocástica com A > B

Fonte: Adaptado de Ulmer et al. (2022)

Porém, esses casos raramente são tão claros na prática. Por isto, del Barrio et al. (2018) e Dror et al. (2019) consideram a noção de Dominância Quase Estocástica, quantificando até que ponto a ordem estocástica está sendo violada. Na Figura 2.9, tem-se as funções de distribuição acumulada para os algoritmos B e C (representado pela linha azul). A área vermelha representa a violação da ordem estocástica.

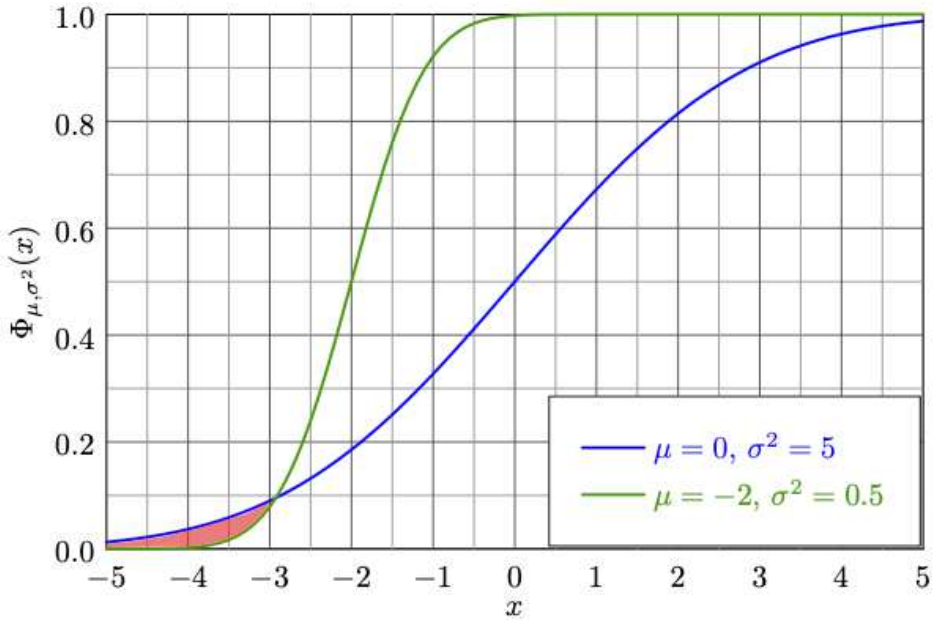

Figura 2.9 – Ordem Quase Estocástica com C > B

Fonte: Adaptado de Ulmer et al. (2022)

O teste ASO retorna um valor ϵ_{min}, que é um limite superior e expressa a quantidade de violação da ordem estocástica. Se $\epsilon_{min} < \tau$ (onde τ é 0,5 ou menos), A é considerado estocasticamente dominante sobre B em mais casos do que o inverso, então o algoritmo A pode ser declarado como superior.

O valor ϵ_{min} também pode ser interpretado como um *score* de confiança. Quanto mais baixo este valor, mais certeza tem-se de que A é melhor que B.

Como mencionado anteriormente, o teste ASO não computa valores p. Assim, a hipótese nula é formulada conforme a Equação 9.

$$H_0 : \epsilon_{min} \geq \tau \tag{9}$$

2.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo, foram discutidos todos os temas que permearam esta tese e que são necessários para a compreensão da metodologia desenvolvida para detecção de discurso de ódio utilizando CLL. PLN, redes neurais profundas e transferência de aprendizagem foram alguns dos temas abordados. Foram discutidos alguns aspectos jurídicos no tocante a detecção de discurso de ódio. Além disso, o conceito de CLL foi definido, bem como sua importância e utilidade em tarefas que envolvem Processamento de Linguagem Natural. No próximo capítulo, serão expostos os trabalhos relacionados a esta tese.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta os trabalhos relacionados sobre detecção de discurso de ódio utilizando cruzamento de idiomas (CLL). Inicialmente, são apresentados trabalhos que realizam detecção de ódio utilizando Aprendizado de Máquina Tradicional; em seguida, são apresentados os trabalhos que utilizaram solução de Aprendizado de Máquina Profundo. Por fim, são discutidos os trabalhos que utilizaram CLL em suas abordagens.

3.1 DETECÇÃO DE DISCURSO DE ÓDIO COM APRENDIZADO DE MÁQUINA TRADICIONAL

Schmidt e Wiegand (2017) escreveram o primeiro *survey* na área de discurso de ódio. É um trabalho breve, e que apresenta um panorama resumido acerca de 22 estudos na área de discurso de ódio. É importante ressaltar que os autores consideraram outros tipos de linguagens ofensivas, como *cyberbullying*, como sendo discurso de ódio - diferentemente da abordagem de Fortuna e Nunes (2018), que fez a diferenciação de discurso de ódio para os demais tipos de ofensa.

Fortuna e Nunes (2018) publicaram um importante *survey* na área de discurso de ódio. Eles elaboraram uma definição do conceito, baseado no código de conduta da Comissão da União Europeia, nos termos e condições de redes sociais, como Facebook e Twitter e em artigos científicos da área. Os autores classificaram os artigos pesquisados quanto ao tipo de discurso de ódio abordado (racismo, sexismo etc.); quanto às características utilizadas no processo de detecção de discurso de ódio (n-gramas, tf-idf etc.) e quanto ao classificador utilizado (SVM, redes neurais, dentre outros). Além disso, os autores apontam alguns desafios e oportunidades de pesquisa na área.

Waseem e Hovy (2016) apresentam uma base de dados com cerca de 16.000 *tweets*, dos quais 3.383 foram anotados como tendo conteúdo sexista e 1.972, como tendo conteúdo racista. Experimentos foram realizados utilizando regressão logística para classificar se o conteúdo do *tweet* continha discurso de ódio ou não, e a técnica PLN utilizada foi n-gramas de caracteres (n=4). O melhor resultado (medida F1 de 73.93%) foi obtido ao combinar o uso de n-gramas de caracteres com o gênero do usuário que postou o *tweet*.

Burnap e Williams (2016) lidaram com detecção de discurso de ódio em micro-textos, usando uma abordagem multi-classe. A ideia dos autores era identificar textos preconceituosos em relação à raça, orientação sexual e deficiências físicas. Foi utilizada uma base de *tweets* criada pelos autores relativa a dois eventos específicos: a reeleição de Barack Obama e o anúncio público de Jason Collins (primeiro atleta ativo em uma equipe

esportiva profissional americana a se declarar gay), ambos ocorridos nos Estados Unidos entre os anos 2012 e 2013. As características utilizadas foram *bag of words*, um dicionário montado com uma lista de termos odiosos extraídos da Wikipédia, dependências tipadas em conjunto com n-gramas (de tamanhos entre 1 e 5). Os autores trabalharam com a ideia de que existem classes-alvo de discurso de ódio e que pode haver interseção entre classes. Cerca de 2.000 *tweets* de cada evento (somando 4.000 *tweets*) foram anotados por pessoas usando o CrowdFlower (https://appen.com/). A proposta de utilizar dependência tipada na tarefa de detecção de discurso de ódio vem do trabalho anterior dos autores. No presente trabalho, a ideia foi testar a eficácia da proposta no caso de haver várias classes-alvo. Os resultados mostraram que a utilização de n-gramas de palavras em conjunto com n-gramas de dependências tipadas e termos odiosos apresentaram os melhores resultados para as três classes em estudo: deficiência, raça e orientação sexual.

Nobata et al. (2016) desenvolveram uma abordagem supervisionada para detecção de linguagem abusiva - que engloba discurso de ódio, profanidade, e linguagem depreciativa - utilizando técnicas de Processamento de Linguagem Natural. Três bases de dados foram utilizadas no trabalho. Na primeira, foram coletados 10% dos comentários diários dos sites *Yahoo News* e *Yahoo Finances* entre Outubro de 2012 e Janeiro de 2014. A segunda base de dados consistiu em comentários também extraídos do *Yahoo Finances* e *Yahoo News*, no período de Abril de 2014 a Abril de 2015. A terceira base de dados utilizada foi a disponibilizada publicamente em outro trabalho (DJURIC et al., 2015), com cerca de 950.000 comentários. Os autores utilizaram n-gramas, características linguísticas - como número de *tokens* nos comentários, número de sinais de pontuação, dentre outros-; características sintáticas, como *POS tagging* e características semânticas, como *embeddings* - word2vec. A combinação da utilização de todas as características resultou num medida F1 de 79,5% (comentários de finanças) e 81,7% (comentários de notícias) para o primeiro conjunto de dados; 78,3% para o segundo conjunto de dados e 82,6% para o terceiro conjunto de dados.

Bourgonje et al. (2017) foram os primeiros a utilizarem dados de discurso de ódio em mais de um idioma (inglês e alemão). Foram utilizadas três bases de dados públicas de textos: duas do Twitter (uma em inglês e outra em alemão), e uma da Wikipédia (em inglês). O trabalho realizou uma comparação utilizando várias técnicas para detecção de discurso de ódio, como Regressão Logística e Entropia Máxima, por exemplo. Para todas as bases de dados, foi utilizado apenas *bag of words*, que os autores consideraram como sendo o mesmo que uni-gramas. Foram testados cinco classificadores distintos, dentre os quais se destacaram *Naive-Bayes* e Regressão Logística.

Anagnostou et al. (2018) tratam de detecção automática de discurso de ódio em comentários do YouTube. A abordagem dos autores baseia-se em um motor de classificação

que utiliza a técnica SVM para classificar os comentários em *odiosos* ou não. Foram utilizadas as características tf-idf e *bag of words*.

Frenda et al. (2019) tratam de discurso de ódio contra mulheres. Os autores se propuseram a investigar analogias e diferenças entre sexismo e misoginia do ponto de vista computacional, e alegaram ser os primeiros a detectar ambos os conceitos usando uma mesma abordagem. Utilizando dados do IberEval 2018 e Evalita 2018 (ambos sobre misoginia) e de Waseem e Hovy (2016), sobre sexismo, os autores constataram que os *tweets* que foram rotulados como sexistas ou misóginos apresentavam mais pronomes femininos que os *tweets* rotulados como neutros. Para classificação dos textos, os autores realizaram um pré-processamento (remoção de *emoticons*, de *urls* e símbolos, e lematização), utilizaram n-gramas de caracteres e palavras e tf-idf, e usaram SVM como classificador.

No Quadro 3.1, tem-se um resumo dos trabalhos que utilizaram abordagens de aprendizado de máquina tradicionais para realizar detecção de discurso de ódio. Dentre as técnicas de classificação, destaca-se SVM, apresentando ótimos resultados para as bases de dados utilizadas. SVM mapeia os dados de entrada em um espaço de alta dimensão para categorizá-los, mesmo para cenários em que os dados não são linearmente separáveis. Já em relação às *features* utilizadas, vê-se a predominância do uso de *bag of words*, que apesar de apresentar baixa complexidade, conseguem realizar uma boa representação dos dados

Quadro 3.1 – Resumo dos trabalhos relacionados: aprendizado de máquina tradicional.

Trabalho	**Tema**	***Features***	**Técnica de Classificação**	**Fonte de Dados**
Waseem e Hovy (2016)	Sexismo e racismo	N-gramas de caracteres e gênero do usuário	Regressão Logística	Twitter
Burnap e Williams (2016)	Racismo, capacitismo e homofobia	Dependências tipadas, *bag of words* e termos odiosos	SVM	Twitter
Nobata et al. (2016)	Discurso de ódio em geral	N-gramas, *embeddings* e classificação gramatical	Regressão de Vowpal Wabbit	Twitter
Bourgonje et al. (2017)	Discurso de ódio em geral	*Bag of words*	Naive-Bayes	Twitter e Wikipédia
Anagnostou et al. (2018)	Discurso de ódio em geral	*Bag of words* e tf-idf	SVM	Youtube
Frenda et al. (2019)	Sexismo e misoginia	N-gramas, *bag of words*, tf-idf e lexicons	SVM	Twitter

Fonte: Elaborado pelo autor, 2022.

3.2 DETECÇÃO DE DISCURSO DE ÓDIO COM APRENDIZADO DE MÁQUINA PROFUNDO

Badjatiya et al. (2017) trataram da detecção de discurso de ódio - mais especificamente, racismo e sexismo - em microtextos. Os autores compararam diversas técnicas e o melhor desempenho (medida F1 de 93%) foi obtido ao se utilizar LSTM em conjunto

com *Gradient Boosting* e *bag of words*. Foi utilizada uma base de dados com 16.000 *tweets*, dos quais cerca de 3.000 estavam anotados como sendo sexistas e cerca de 1.900 como racistas. Segundo os autores, este foi o primeiro trabalho a utilizar Aprendizado de Máquina Profundo para detectar discurso de ódio.

Del Vigna et al. (2017) tratam da detecção de discurso de ódio em textos de redes sociais. Foram coletados cerca de 17.000 comentários do Facebook, os quais estavam em 99 postagens. Cinco estudantes de graduação anotaram os comentários como sendo discurso de ódio ou não. Os autores utilizaram dois classificadores na experimentação: um SVM e uma rede neural LSTM. Foram usados *lexicons* para identificar a polaridade dos sentimentos nos textos. No classificador SVM, foram utilizadas características como o número de *tokens*, n-gramas em nível de caracteres, palavras e lemas. No classificador LSTM, foram utilizados a polaridade de cada palavra e *lexicons*. Os autores realizaram experimentos com duas abordagens: uma abordagem multi-classe com níveis de discurso de ódio nos textos e a outra, binária. Os melhores resultados foram obtidos com a classificação binária, com 80% de medida F1 para SVM e 79% para LSTM.

Zhang et al. (2018) trataram de detecção de discurso de ódio em microtextos no tocante à temática de refugiados na Europa. Os autores utilizaram duas redes neurais para realizar tal tarefa - uma CNN e uma GRU -, além de *embeddings* do corpus obtido do *Google News*. Foram utilizados diversos corpora disponibilizados por outros autores e foram feitas comparações com o uso de técnicas, como SVM e a técnica criada, demonstrando que a técnica desenvolvida pelos autores obteve uma medida F1 superior às demais.

A abordagem de Pitsilis et al. (2018) para detecção de discurso de ódio baseia-se no uso de múltiplas redes LSTM para fazer a classificação. No pré-processamento, os autores vetorizaram os *tweets* utilizando tokenização, *padding* e indexação. Como características, foram usados o histórico de postagens dos usuários, chamado de tendência acerca de cada classe, e os vetores dos *tweets*. Os autores utilizaram a base de dados disponibilizada por Waseem e Hovy (2016), apresentando a medida F1 de 93,2%.

Silva et al. (2019) desenvolveram uma nova abordagem para detectar discurso de ódio em português, que compreende um modelo que utiliza CNN e um dicionário psicolinguístico, o *Linguistic Inquiry and Word Count* (LIWC), com Regressão Logística (LR + LIWC). Eles usaram três conjuntos de dados brasileiros: OffComBr2 e OffComBr3 (PELLE; MOREIRA, 2017) e HSD (FORTUNA, 2017), e compararam os resultados dos autores que disponibilizaram as bases de dados (*baselines*) com os deles. O melhor resultado obtido consistiu no uso de uma CNN junto com um *embedding* de palavras de tamanho igual a 300.

Soto et al. (2019) propuseram comparar o impacto do uso de diferentes estratégias de geração de vetores de palavras na tarefa de detecção de discurso de ódio com dados em português. Eles usaram Word2Vec e Wang2Vec (LING et al., 2015) com uma rede neural convolucional (CNN) para realizar a classificação. Os autores utilizaram os dados fornecidos por Pelle e Moreira (2017) e Fortuna (2017). Apesar de não terem obtido os melhores resultados com relação ao estado da arte, os autores argumentaram que os resultados ficaram muito próximos dos melhores resultados utilizando *embeddings* com dimensões menores e, portanto, utilizando menos recursos computacionais.

Chowdhury et al. (2019) trataram de discurso de ódio em Árabe. Apesar de não especificar o alvo específico, os autores relataram que a abordagem foi voltada para discurso de ódio de cunho religioso. Foram coletados cerca de 5.000 *tweets*, dos quais 42% foram rotulados como tendo discurso de ódio e os 58% restantes como neutros. Os autores alegam ser os primeiros a utilizarem vetores de palavras em conjunto com vetores de grafos (mais especificamente, de *retweets* e seguidores) dos usuários que postaram as mensagens odiosas. Foi utilizada uma combinação das redes LSTM + CNN. Assim, os resultados superaram soluções existentes na língua árabe, que usavam GRU e SVM, em 2%.

Paetzold et al. (2019) participaram do HatEval 2019, conseguindo o 7º lugar para a sub-tarefa A em inglês, que consistia na detecção de discurso de ódio em textos. A proposta dos autores baseia-se na utilização de Redes Neurais Recorrentes (RNN) e *embeddings* de caracteres. O modelo recebe uma sentença, que é decomposta em palavras, que por sua vez é decomposta em caracteres. Na primeira camada do modelo, os *embeddings* dos caracteres são postos como pesos; em seguida uma GRU bidirecional mapeia os caracteres para palavras. Em seguida, uma outra GRU bidirecional mapeia as palavras para sentenças. O modelo foi treinado utilizando apenas os dados de treino do desafio.

Venturott e Ciarelli (2020) também abordam o problema de se ter poucos dados para discurso de ódio na língua portuguesa. A solução adotada pelos autores consiste na utilização de uma técnica de aumento de dados (do inglês, *data augmentation*), podendo, desta forma, incrementar a base de dados utilizada e conseguindo melhores resultados. Para os experimentos, os autores utilizaram a estratégia Glove para a geração de *embeddings* e redes neurais LSTM e CNN. O melhor resultado obtido foi utilizando uma CNN com *data augmentation* sem a utilização de *embeddings* gerados.

Fortuna et al. (2021) criaram uma abordagem para detecção de discurso de ódio utilizando diferentes corpora em inglês. Eles padronizaram as classes presentes nos corpora para ser possível a utilização dos diferente corpora de forma intercambiável. Para realizar a classificação, foram utilizados os MLPTs ALBERT e BERT, e as abordagens fastText e SVM. Os resultados mostraram como uma abordagem *cross-dataset* pode ser usada

para detectar semelhança entre corpora e categorias, e ajudar a criar uma categorização uniforme de corpora.

Plaza-del-Arco et al. (2021) utilizaram bases de dados em espanhol e compararam diversos modelos, tanto de Aprendizado de Máquina tradicional quanto de Aprendizado de Máquina Profundo, quanto ao desempenho na tarefa de detecção de discurso de ódio em textos. As bases de dados utilizadas consistem de textos extraídos do Twitter; uma delas contendo 6.000 *tweets* obtidos pelo HaterNet - um sistema inteligente utilizado pelo Escritório Nacional Espanhol contra Crimes de Ódio da Secretaria de Estado de Segurança da Espanha -, e a outra contendo 1.600 *tweets* obtidos do desafio HatEval 2019. Os autores concluíram que os modelos que utilizam aprendizado profundo alcançaram melhores resultados que os de aprendizado de máquina tradicional. Além disso, os que utilizam transferência de aprendizagem (os quais os autores colocaram como uma categoria à parte) se sobressaíram dentre os modelos que usam aprendizado profundo.

Karim et al. (2021) desenvolveram uma abordagem para detectar discurso de ódio em Bengali utilizando MLPTs. Foram experimentadas diversas técnicas, desde *Naive Bayes* e Regressão Logística, até CNN e MLPTs. Os melhores resultados foram obtidos com uma combinação de MLPTs (BERT em Bengali, XLM-R e BERT multilíngue). Além de realizarem classificação binária, a abordagem desenvolvida pelos autores também se mostrou apta para realizar classificação multi-classe, conseguindo os melhores resultados no estado da arte para o idioma Bengali.

Soto et al. (2022) experimentaram o uso de diferentes *embeddings*, com dimensões distintas e utilizaram uma rede CNN para realizar a classificação dos textos. Os autores usaram *embeddings* específicos gerados para o HSD e *embeddings* obtidos do NILC (HARTMANN et al., 2017). Foram testadas as representações de *embeddings* *wang2vec*, *word2vec*, *fastText* e *Glove*. O melhor resultado obtido para o corpus HSD foi utilizando a representação *Glove* com 300 dimensões, com os *embeddings* NILC. A abordagem dos autores obteve o melhor resultado para o corpus HSD até então.

Um resumo dos trabalhos relacionados das abordagens que utilizam aprendizado de máquina profundo para detectar discurso de ódio pode ser visto no Quadro 3.2. Pode-se ver que o número de trabalhos relacionados com aprendizado de máquina profundo é bem superior ao de trabalhos com aprendizado de máquina tradicional. Dentre as *features* utilizadas nos trabalhos, destaca-se o uso de *embeddings*. A ideia é representar os textos de entrada como vetores de características, podendo assim estar dispostos em um espaço vetorial. Esta ideia se assemelha à proposta da técnica de classificação SVM.

Quadro 3.2 – Resumo dos trabalhos relacionados: aprendizado de máquina profundo.

Trabalho	Tema	*Features*	Técnica de Classificação	Fonte de Dados
Badjatiya et al. (2019)	Sexismo e racismo	*Embeddings*	LSTM + Gradient Boosting	Twitter
Del Vigna et al. (2017)	Política	N-gramas (caracteres, palavras e lemas) e número de *tokens*	LSTM	Facebook
Zhang et al. (2018)	Xenofobia	*Embeddings*	CNN + GRU	Twitter
Pitsilis et al. (2018)	Sexismo e racismo	*Embeddings* e histórico de postagens do usuário	Múltiplas LSTM	Twitter
Silva et al. (2019)	Discurso de ódio em geral	*Embeddings* e léxicos	CNN	Twitter e site de notícias
Soto et al. (2019)	Discurso de ódio em geral	*Embeddings*	CNN	Twitter e site de notícias
Chowdhury et al. (2003)	Religião	*Embeddings* de palavras e grafos	LSTM + CNN	Twitter
Paetzold et al. (2019)	Sexismo, xenofobia e política	*Embeddings* de caracteres	Múltiplas RNNs	Twitter e Facebook
Venturott e Ciarelli (2020)	Discurso de ódio em geral	*Embeddings*	CNN	Twitter
Fortuna et al. (2021)	Discurso de ódio em geral	*Embeddings*	MLPTs	Variada
Plaza-del-Arco et al. (2021)	Discurso de ódio em geral	*Embeddings*	BETO	Twitter
Karim et al. (2021)	Discurso de ódio em geral	*Embeddings*	MLPTs	Variada
Soto et al. (2022)	Discurso de ódio em geral	*Embeddings*	CNN	Twitter

Fonte: Elaborado pelo autor, 2022.

3.3 DETECÇÃO DE DISCURSO DE ÓDIO COM CRUZAMENTO DE IDIOMAS

Pagmunkas e Patti (2019) desenvolveram uma abordagem de detecção de discurso de ódio utilizando CLL, na qual utiliza-se o conceito de transferência de aprendizagem de um idioma com mais dados para outro idioma com menos dados anotados. Os idiomas estudados foram inglês, espanhol, italiano e alemão. Os melhores resultados foram obtidos utilizando Hurtlex (BASSIGNANA et al., 2018) e *embeddings* multilíngues como características e uma arquitetura LSTM. A melhor abordagem utilizou *Joint Learning* como estratégia de treinamento e *embeddings* multilíngues.

Hu et al. (2020) desenvolveram um *benchmark* para avaliar o desempenho de *transformers* em abordagens com cruzamento de idiomas. XTREME - nome do *benchmark* - abrange 40 linguagens tipologicamente diversas, incluindo 12 famílias linguísticas e com 9 tarefas incluindo diferentes níveis de sintaxe ou semântica. O XTREME concentra-se no cenário de cruzamento de idiomas com a utilização da estratégia *zero-shot*, em que os dados de treinamento anotados são fornecidos em inglês, e os dados de teste são em outros idiomas.

Corazza et al. (2020) criaram uma abordagem para detectar discurso de ódio

independente de idioma. Eles discutem que existem diversos trabalhos na literatura que utilizam *embeddings* específicos de um domínio, e outras características que só se aplicam a uma base de dados em particular. Foi construída uma arquitetura neural modular que usa uma camada escondida com 100 neurônios. A melhor configuração encontrada pelos autores para os dados em inglês foi o uso de LSTM com *embeddings* de caracteres; já para o italiano, os melhores resultados foram obtidos ao usar LSTM combinado com *embeddings* de caracteres, uni-gramas e transcrição de *emojis*. Para o idioma alemão, o melhor resultado foi obtido ao se utilizar *embeddings* de caracteres como features e uma rede GRU. As características utilizadas foram *embeddings* de palavras, *embeddings* de *emojis*, n-gramas, *emotion lexica* e características de redes sociais (número de *hashtags*, menções, etc).

Ranasinghe e Zampieri (2020) utilizaram *embeddings* de palavras multilíngue para detectar discurso de ódio. Além de realizar experimentos com idiomas diferentes, domínios diferentes também foram testados. Foram obtidos dados em inglês, espanhol, hindi e bengali. Foi utilizado o *framework* XLM-R para realizar a classificação. A ideia do uso de uma abordagem que utilize CLL é treinar o modelo em um idioma com mais dados e testar em outro idioma com menos dados. Os autores treinaram o modelo em inglês e testaram nos outros três idiomas pesquisados. Os resultados demonstrados ultrapassaram o estado da arte de cada conjunto de dados e idioma.

Stappen et al. (2020) desenvolveram uma abordagem para detectar discurso de ódio utilizando CLL, incluindo algumas amostras do idioma destino ao treinamento. A arquitetura proposta utiliza *embeddings* gerados por FastText, um extrator de características (BERT ou XLM) e uma abordagem chamada *Attention-Maximum-Average Pooling* (AXEL) para realizar a classificação. Essa abordagem melhora as características geradas pelo XLM, usando o *pooling* máximo e médio da saída do XLM.

Pagmunkas et al. (2021) desenvolveram uma abordagem para detecção de discurso de ódio utilizando cruzamento de idiomas. Eles utilizaram o inglês como idioma-fonte e seis idiomas como alvo: português, francês, espanhol, alemão, indonésio e italiano. Os autores experimentaram diversas combinações de modelos, desde modelos que utilizam Aprendizado de Máquina tradicional - como regressão logística -, até modelos que utilizam Aprendizado de Máquina Profundo - como BERT, por exemplo. Os autores utilizaram *zero-shot transfer* e *joint learning* nos experimentos. O modelo que obteve a melhor performance foi uma rede neural LSTM utilizando *embeddings* multilíngues fornecidos pelo Facebook (MUSE - (LAMPLE et al., 2018)).

Pelicon et al. (2021) realizaram uma análise sobre a viabilidade de modelos de linguagem que permitem representações multilíngues com dados de idiomas diferentes.

Foram usados cinco conjuntos de dados de discurso de ódio em diferentes idiomas: árabe, croata, alemão, inglês e esloveno. A metodologia dos autores consistiu das etapas: seleção um modelo de linguagem pré-treinado; treinamento do modelo com dados em uma ou mais linguagens intermediárias; ajuste fino no modelo, adicionando progressivamente dados no idioma alvo; e avaliação de desempenho.

Bigoulaeva et al. (2021) também realizaram detecção de discurso de ódio utilizando CLL. O idioma alvo utilizado na pesquisa foi o alemão, e o inglês foi usado como idioma fonte. Foram testados vários modelos de redes neurais profundas, utilizando as estratégias *zero-shot* e *joint learning*. Os autores desenvolveram assim, uma abordagem de transferência de aprendizagem utilizando CLL baseada em *embeddings* de palavras bilíngues (BWEs).

No Quadro 3.3, tem-se um resumo dos trabalhos relacionados que utilizaram cruzamento de idiomas na detecção de discurso de ódio. É possível ver que todos utilizaram *embeddings* como *features* para seus modelos. Vê-se então que esta forma de representar os textos é promissora quando mais de um idioma é utilizado, podendo fazer abstrações em textos. Pode-se ver também que modelos de linguagem tem sido utilizados e conseguem obter resultados promissores em abordagens com cruzamento de idiomas.

Quadro 3.3 – Resumo dos trabalhos relacionados: *cross-lingual learning*.

Trabalho	**Tema**	***Features***	**Técnica de Classificação**	**Fonte de Dados**
Pagmunkas e Patti (2019)	Discurso de ódio em geral	Uni-gramas, *embeddings* e léxicos	Múltiplas LSTM	Variada
Corazza et al. (2020)	Discurso de ódio em geral	*Embeddings* de caracteres, uni-gramas e transcrição de *emojis*	LSTM + GRU	Twitter
Ranasinghe e Zampieri (2020)	Discurso de ódio em geral	*Embeddings* multilíngues	XLM-R	Twitter e Facebook
Stappen et al. (2020)	Sexismo e xenofobia	*Embeddings*	BERT e XLM	Twitter
Pagmunkas et al. (2021)	Discurso de ódio em geral	*Embeddings* multilíngues	LSTM	Twitter
Pelicon et al. (2021)	Discurso de ódio em geral	*Embeddings*	BERT	Variada
Bigoulaeva et al. (2021)	Discurso de ódio em geral	*Embeddings*	BiLSTM	Variada
Nossa abordagem	Discurso de ódio em geral	*Embeddings*	BERT e XLM	Variada

Fonte: Elaborado pelo autor, 2022.

A abordagem proposta nesta tese se diferencia de Pagmunkas et al. (2021), de Bigoulaeva et al. (2021) e de Pelicon et al. (2021), pois nestes trabalhos foram usadas apenas duas estratégias de treinamento: *zero-shot* e *joint learning*. Além disso, estes trabalhos não tiveram o objetivo de identificar qual idioma teria mais impacto ao ser utilizado como idioma fonte na abordagem. No caso de Pagmunkas et al. (2021) e Bigoulaeva et al. (2021) não foram utilizados modelos de linguagem.

3.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Nesta seção, foram apresentados e discutidos os trabalhos relacionados, dividindo-os em três categorias: os que realizaram detecção de discurso de ódio utilizando abordagens de aprendizado de máquina tradicionais; os que utilizaram abordagens de aprendizado de máquina profundo, e os que utilizaram abordagens envolvendo cruzamento de idiomas.

Dentre os trabalhos apresentados, não foi encontrado nenhum que buscou investigar o impacto de se usar cruzamento de idiomas, ou seja, não foram encontrados trabalhos que realizassem estudos de ablação, demonstrando que o uso de CLL melhora o desempenho em alguma abordagem. Além disso, não foram encontrados trabalhos que investigam o impacto de se usar línguas de diferentes famílias como idiomas-alvo (ex: ao usar alemão como idioma alvo, utilizar inglês como idioma fonte teria mais impacto do que utilizar o holandês? Ou o resultado seria semelhante?). Neste sentido, esta pesquisa objetiva trazer estas investigações e preencher estas lacunas no estado da arte.

No próximo capítulo, será apresentada a abordagem criada para detecção de discurso de ódio utilizando CLL. Além disso, os corpora utilizados nesta pesquisa serão detalhados.

4 METODOLOGIA E DADOS

Neste capítulo, é apresentada a metodologia proposta para detecção de discurso de ódio em português utilizando CLL. Também, os conjuntos de dados utilizados nesta pesquisa são apresentados.

4.1 METODOLOGIA

Esta tese propõe a ideia de aplicação de CLL para realizar a detecção de discurso de ódio. Na Figura 4.1 é apresentada uma visão geral da metodologia, a qual contempla quatro etapas. A primeira etapa consiste na aquisição de corpora, uma vez que neste trabalho utiliza-se uma abordagem que necessita de mais de um idioma. Portanto, requer-se a utilização de pelo menos dois corpora em idiomas distintos.

Na segunda etapa, é feita a definição do(s) Modelo(s) de Linguagem Pré-Treinado(s) (MLPT) a ser(em) utilizado(s). A próxima etapa consiste na definição da estratégia de treinamento a ser empregada. Fundamentando-se no trabalho de Pikuliak et al. (2021), foram utilizadas cinco estratégias de treinamento para detecção de discurso de ódio em português - o idioma alvo (I_a). Assim, o modelo é induzido com as informações obtidas por meio dos dados anotados em um idioma fonte I_f. Por fim, na última etapa tem-se a avaliação do modelo.

Para todas as abordagens, o modelo final foi avaliado usando dados de I_a. Foram utilizadas precisão, revocação e medida F1 como métricas de avaliação. Cada passo da metodologia será detalhado nas próximas subseções.

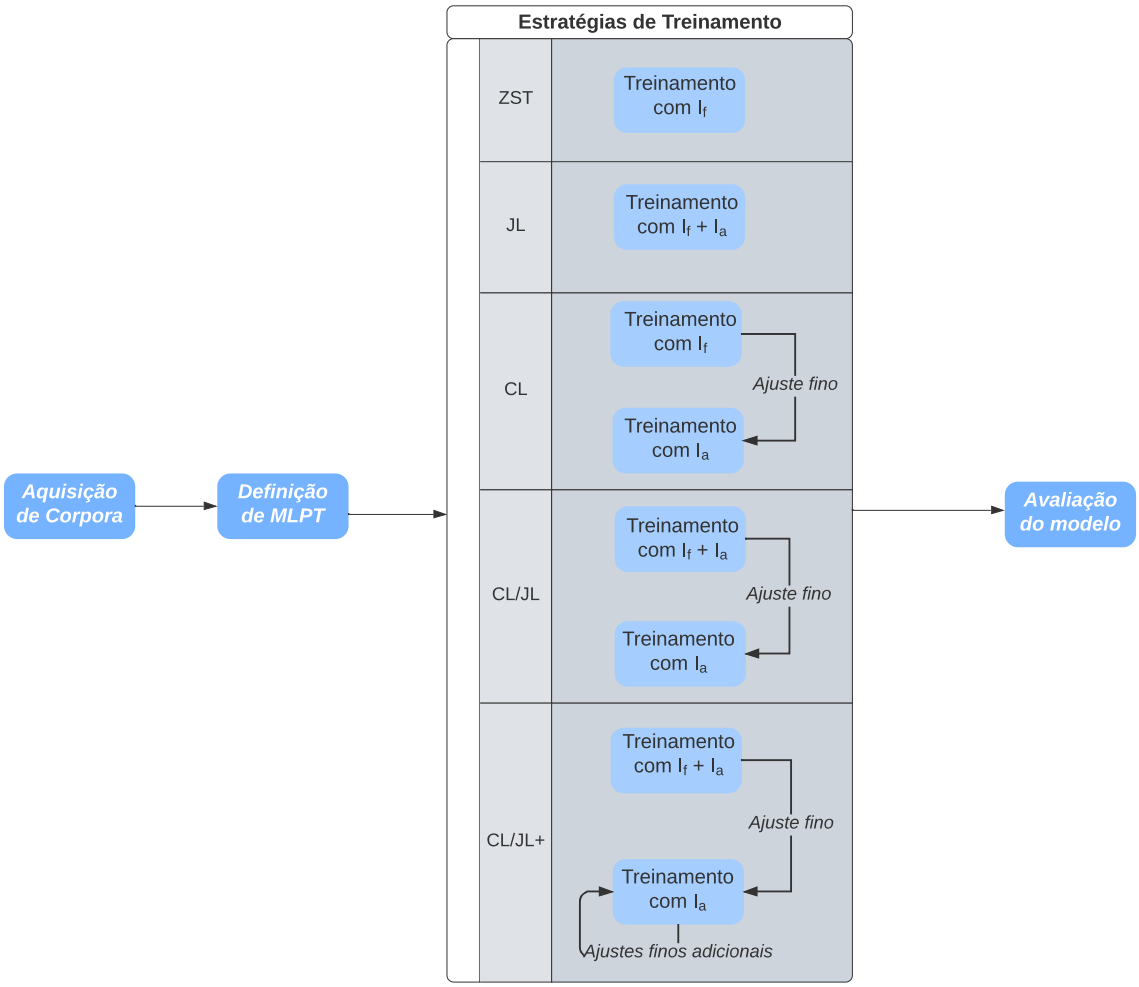

Figura 4.1 – Visão geral da metodologia proposta.

Fonte: Elaborado pelo autor, 2022.

4.1.1 Aquisição de Corpora

Inicialmente, é necessário construir ou obter corpora. É importante enfatizar que ao usar CLL, pelo menos dois corpora são necessários: um com textos no idioma fonte, e outro com textos no idioma alvo. Duas maneiras populares na construção de corpora são utilizando *web crawlers* (robôs que indexam conteúdos de *sites*) para coletar textos em páginas Web (PELLE; MOREIRA, 2017) ou utilizando API's de redes (WASEEM; HOVY, 2016; ARCO et al., 2021). Neste caso, deve-se definir o escopo dos textos a serem coletados, escolhendo quais palavras chave serão utilizadas na busca ou quais páginas da Web serão acessadas. Pelle e Moreira (2017), por exemplo, escolheram construir um corpus obtendo comentários de páginas de notícias sobre política e esportes. Já Waseem e Hovy (2016) construíram um corpus com *tweets* sobre o programa de televisão australiano *My Kitchen Rules*. Para tanto, foram obtidos *tweets* que contivessem a *hashtag* #MKR

(indicando as iniciais do programa).

Pode-se também utilizar corpora já construídos e disponibilizados publicamente (WASEEM; HOVY, 2016; DAVIDSON et al., 2017; CHUNG et al., 2019; FRENDA et al., 2019). Há ainda a possibilidade de utilizar-se corpora disponibilizados em conferências, eventos ou *workshops*, tais como Evalita 2018 (BOSCO et al., 2018), HatEval 2019 (BASILE et al., 2019), dentre outros.

Nesta tese, optou-se por utilizar corpora disponibilizados publicamente. Foram utilizados os corpora descritos nos trabalhos de Waseem e Hovy (2016), Pelle e Moreira (2017), Bosco et al. (2018) e Fortuna et al. (2019).

Após a obtenção dos corpora, o próximo passo é fazer pré-processamento nos textos. Em abordagens que usam redes neurais profundas, o pré-processamento mais comum consiste de computar a representação vetorial dos textos. É importante ressaltar que quando algum MLPT é utilizado, em geral, o próprio modelo contém um módulo responsável pela vetorização dos textos (BHASKARAN; BHALLAMUDI, 2019; HEIJDEN et al., 2021). Tendo em vista que nesta tese utilizam-se MLPT's; foram utilizados os módulos de vetorização dos próprios modelos.

4.1.2 Definição de MLPT

Nesta tese, foram utilizados dois MLPTs distintos: XLM e BERT. Com relação ao XLM, foi utilizada a distribuição XLM-RoBERTa - chamada XLM-R (CONNEAU et al., 2020). Já no que diz respeito ao BERT, utilizou-se três distribuições: o BERT para português - BERTimbau (SOUZA et al., 2020), o BERT para italiano (SCHWETER, 2020) e o BERT original (treinado em inglês) (DEVLIN et al., 2019). Utilizou-se o XLM-R por ele ter sido treinado em vários idiomas e ser multilíngue. Já a escolha do BERT italiano deu-se pois alguns dos corpora utilizados nesta pesquisa são em italiano. O mesmo se aplica para o BERTimbau e o BERT, já que também utiliza-se corpora em inglês e português.

O XLM-R tem apresentado bons resultados em tarefas envolvendo cruzamento de idiomas, alcançando uma acurácia 23% maior que a do MLPT BERT no uso de idiomas com poucos dados disponíveis (DEVLIN et al., 2019). O XLM-R foi treinado em 104 idiomas com 2,5 terabytes de dados, além de ser compatível com *benchmarks* monolíngues, ao mesmo tempo em que alcança os melhores resultados em *benchmarks* de cruzamento de idiomas (CONNEAU et al., 2020).

4.1.3 Estratégias de Treinamento

A primeira estratégia de treinamento é a *Zero-shot transfer* (ZST). Neste caso, nenhum dado do idioma I_a é utilizado no primeiro ajuste fino do MLPT. O treinamento é feito usando apenas os dados de I_f. A segunda estratégia denomina-se Joint Learning (JL) em que os dados de I_a e I_f são usados no primeiro ajuste fino do MLPT ao mesmo tempo.

A terceira estratégia, *Cascade Learning* (CL), consiste na utilização dos dados de I_f no primeiro ajuste fino, seguido por um ajuste fino adicional em que apenas dados de I_a são usados. A quarta estratégia é a CL/JL, em que no primeiro ajuste fino já existe a utilização de uma parte dos dados do corpus de I_a no treinamento. No demais, o fluxo segue como descrito na estratégia CL: após o primeiro ajuste fino, os pesos são salvos e depois carregados para um segundo ajuste fino; por fim, a parte do corpus do idioma alvo que não foi utilizada no treinamento no primeiro ajuste fino é utilizada neste treinamento e também na etapa de teste. A última estratégia de treinamento é a CL/JL+, em que são feitos ajustes finos adicionais utilizando os dados de I_a. Neste caso, o fluxo é semelhante a CL/JL, em que dados de I_f e I_a são utilizados no primeiro ajuste fino ao mesmo tempo, e são feitos vários ajustes finos com os dados de I_a.

4.1.4 Avaliação

O último passo da metodologia proposta consiste na avaliação do modelo induzido nas etapas anteriores. Métricas de avaliação são usadas para monitorar e mensurar o desempenho do modelo, tanto durante o treinamento, quanto durante o teste. Na metodologia proposta, as métricas utilizadas foram Precisão (Equação 5), Revocação (Equação 6) e Medida F1 (Equação 7) (ZHANG; ZHANG, 2009). É importante ressaltar que utilizou-se a medida F1 ponderada em todos os experimentos realizados nesta tese.

4.2 CORPORA UTILIZADOS

Nesta pesquisa, foram utilizados cinco conjuntos de dados distintos. Alguns corpora foram derivados destes cinco, resultando assim em seis corpora utilizados no total. O primeiro é composto por dados em língua portuguesa (OffComBr-2 - (PELLE; MOREIRA, 2017)), com comentários contendo discurso de ódio coletados no site de notícias brasileiro *g1.globo.com*. Os dados foram anotados de forma binária, indicando se havia ou não a presença de discurso de ódio. A partir do corpus OffComBr-2, foi gerado um novo corpus com as traduções dos textos para o inglês. A tradução foi realizada usando a Máquina de Tradução Neural da Google (WU et al., 2016).

O segundo conjunto de dados é composto por postagens do Facebook em italiano, disponibilizadas publicamente na conferência Evalita 2018 (BOSCO et al., 2018), na tarefa

Hate Speech Detection. A anotação dos dados foi realizada com duas classes: discurso de ódio ou neutro. O terceiro conjunto de dados consistiu da junção do segundo conjunto com postagens do Twitter em italiano, também disponibilizados na conferência Evalita 2018.

O quarto conjunto de dados utilizado nesta pesquisa consiste de *tweets* em português (FORTUNA et al., 2019), obtidos entre janeiro e março de 2017. Os autores realizaram uma classificação binária (indicando se havia a presença de discurso de ódio ou não) e uma hierárquica contendo nove subclasses de discurso de ódio (sexismo, racismo, religião, dentre outros). O quinto conjunto de dados utilizado também é composto por *tweets*, porém em língua inglesa. Waseem e Hovy (2016) coletaram os *tweets* durante um período de dois meses e anotaram os dados em três classes: sexismo, racismo e neutro.

4.2.1 Corpus OffComBr-2

Pelle e Moreira (2017) dividiram o corpus em duas partes: OffComBr-2, que contém os 1.250 comentários apontados como ofensivos ou neutros por pelo menos 2 juízes; OffComBr-3, que contém 1.033 comentários anotados (dos 1.250) por pelo menos 3 juízes. No OffComBr-2, 419 (de 1.250) comentários foram rotulados como ofensivos por pelo menos dois juízes, representando 32,5% do total. No OffComBr-3, 202 comentários foram rotulados como ofensivos (em 1.033), correspondendo a 19,5% dos casos. Dos 1.250 comentários, 419 foram ofensivos por pelo menos dois juízes, o que representa 32,5% do total. Os autores não forneceram informações se houve algum pré-processamento no conjunto de dados, porém observou-se que sinais de pontuação e acentos foram removidos dos textos.

O Quadro 4.1 exibe uma amostra de comentários desse conjunto de dados. Na Figura 4.2, tem-se uma nuvem de palavras com as palavras mais frequentes do corpus. É possível ver que a palavra mais frequente é Brasil, seguida por termos como ‘melhor’, ‘povo’, ‘dinheiro’, dentre outros. A explicação para isto é o fato do corpus ter sido gerado a partir de comentários sobre política e esportes. Apesar do corpus ter sido obtido em um site de notícias, no qual não há limitação de caracteres, vê-se no histograma gerado que a maioria das sentenças contém entre 0 e 20 palavras (Figura 4.3).

4.2.2 Corpora Evalita 2018

O primeiro corpus italiano utilizado nesta pesquisa foi o disponibilizado publicamente na conferência Evalita 2018. Ele foi desenvolvido por um grupo de pesquisa do *Istituto di Informatica e Telematica* em CNR, Pisa; foi criado em 2016 (VIGNA et al., 2017) e contém cerca de 17.000 comentários do Facebook, extraídos de noventa e nove postagens de páginas selecionadas. Cinco pessoas anotaram cerca de quatro mil comentários e os classificaram como sendo odiosos ou não. Os autores não disponibilizaram informações

sobre a realização de algum tipo de pré-processamento nos textos fornecidos. Observou-se que as letras maiúsculas e minúsculas foram mantidas, bem como sinais de pontuação e acentuação.

Quadro 4.1 – Amostra do corpus OffComBr-2.

Texto original	**Discurso de ódio**
cuidado com a poupanca pessoal Lembram o que aconteceu na epoca do Collor ne	Não
os cariocas tem o que merecem um pessoal que so sabem toma banho de sol e pratica a violencia e nao deu outra de onde se tira e nao coloca um dia acaba	Sim
Voces sao idiotas ou se fazem Voces sabem que ele e do executivo e quer que interfira nos outros poderes Esses salarios sao vinculados a texto constitucional se informem mais nao que ele seja santo mas pra nao passarem vergonha de falar coisa que nao sabem	Sim
Porque nao corta os gastos dos politicos	Não
PEC DA VIDAAAA VIDA LIVRE DE MAMATA ESQUERDALHAAAAAA kkkkkkkkkk	Sim

Fonte: Elaborado pelo autor, 2022.

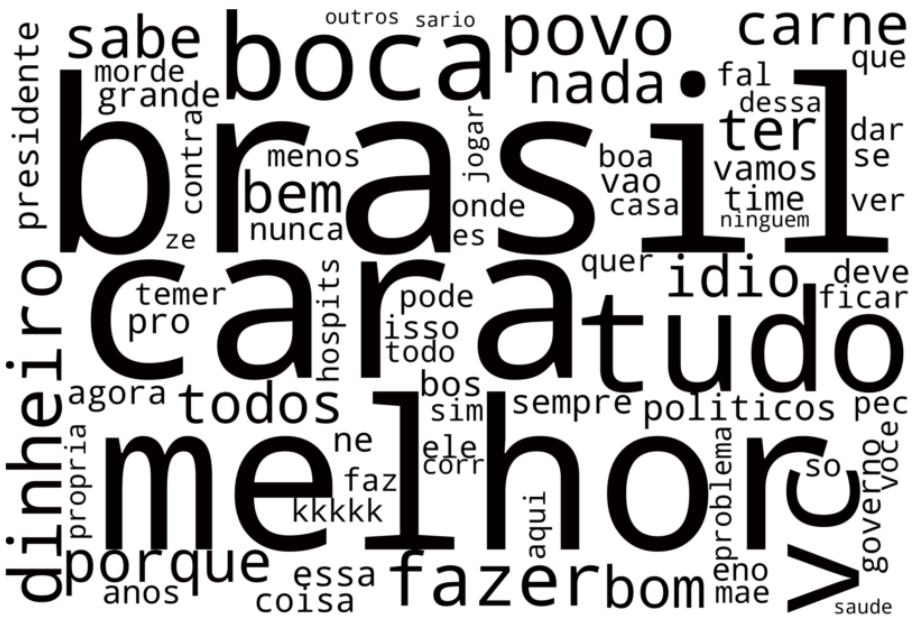

Figura 4.2 – OffComBr-2: Nuvem de palavras.

Fonte: Elaborado pelo autor, 2022.

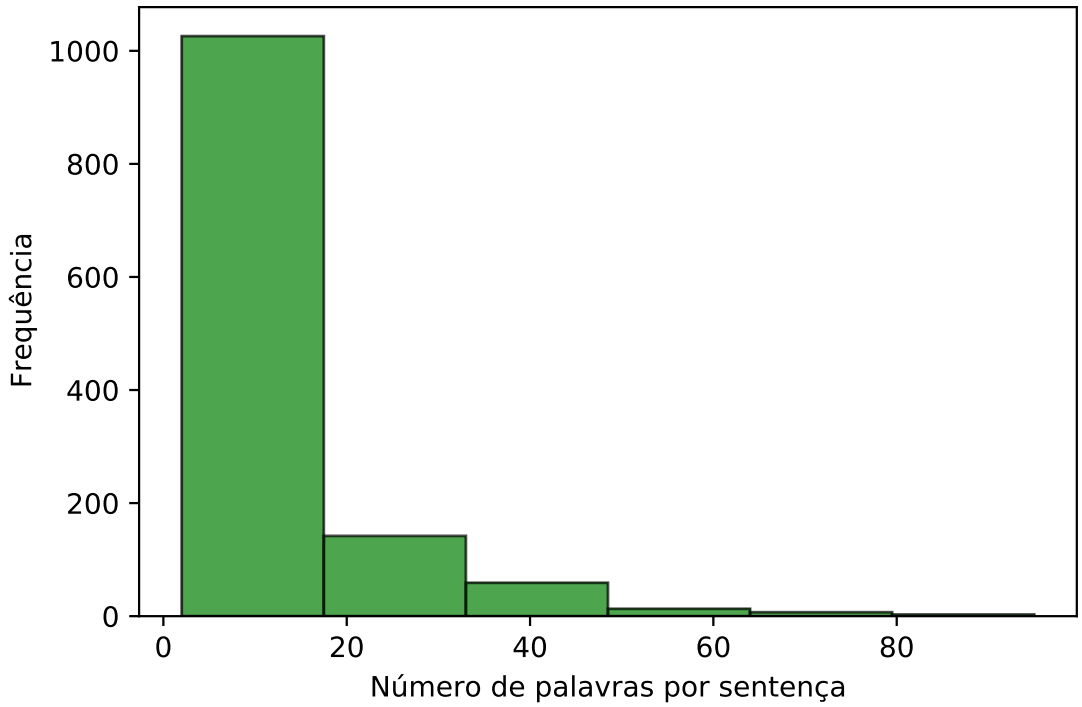

Figura 4.3 – OffComBr-2: Histograma de número de palavras por sentença.

Fonte: Elaborado pelo autor, 2022.

Além dos dados do Facebook, no Evalita 2018 também existia um outro corpus extraídos do Twitter. Este foi desenvolvido pela Universidade de Turin, e faz parte do Programa de Monitoramento contra o Discurso de Ódio[1]. Este corpus foi desenvolvido por (POLETTO et al., 2017) e (SANGUINETTI et al., 2018). Os textos contidos neste conjunto de dados estão relacionados a três grupos: imigrantes, muçulmanos e romanos. Assim sendo, o segundo corpus em italiano utilizado nesta tese consistiu da junção dos dados do Facebook com os dados do Twitter.

O Quadro 4.2 exibe uma amostra de comentários dos corpora italianos utilizados. Pode-se ver as palavras mais frequentes contidas nos corpora italianos na Figura 4.4. 'Mateo', 'Salvini', 'voto' e 'Renzi' são alguns dos termos com maior aparição nos conjuntos de dados. *Matteo Salvini* estava à frente da Liga Norte, enquanto *Matteo Renzi* era o líder do Partido Democrático (PD) na época em que os conjuntos de dados foram coletados. As páginas do Facebook utilizadas para coletar comentários tinham cunho político em sua

[1] http://hatespeech.di.unito.it/

maioria, por isso estes foram os termos mais frequentes. Na Figura 4.5, tem-se o histograma com o número de palavras em cada sentença. O conjunto de dados do Twitter colaborou para um número pequeno de palavras por sentença. Cerca de 70% dos textos dos corpora contém entre 1 e 20 palavras, e o restante acima de 20, tendo em vista que no Facebook não há restrição quanto ao tamanho dos textos postados.

Quadro 4.2 – Amostra dos corpora Evalita 2018.

Texto original	**Tradução**	**Discurso de ódio**
Sta cacchio di Malpezzi quando parla con quel mezzo sorriso da Ebete mi fa venire su i nervi. Quanto la odio! Sta rincoglionita lei e tutti i babbioni che la pensano uguale a lei	Maldita Malpezzi quando ela fala com aquele meio sorriso de bobo me dá nos nervos. Como eu a odeio! Ela está chapada e todos os babuínos que pensam o mesmo que ela	Sim
Sempre a fare servizi strappa lacrime sugli immigrati,come se fossero tutti dei martiri. Quasi tutti criminali #gabbiaopen @user	Sempre prestando serviços, arranca lágrimas dos imigrantes, como se fossem todos mártires. Quase todos os criminosos #gabbiaopen @user	Sim
diose..la malpezzi vive su un altro pianeta...e la castaldini fa parte del partito più ridicolo d Italia ed è irritante..da prendere a schiaffi x ore	Deus.. malpezzi mora em outro planeta ... e castaldini faz parte do partido mais ridículo da Itália e é irritante .. deveria ser esbofeteado por horas.	Sim
La Malpezzi vuole farci credere che esiste Babbo Natale e la Castaldini appartiene a un partito fantasma	Malpezzi quer que acreditemos que Papai Noel existe e Castaldini pertence a um partido fantasma	Não
Siamo arrivati a un punto di non ritorno. POVERI NOI!!!!	Chegamos a um ponto sem volta. POBRES DE NÓS!!!!	Não

Fonte: Elaborado pelo autor, 2022.

Figura 4.4 – Evalita 2018: Nuvem de palavras.

Fonte: Elaborado pelo autor, 2022.

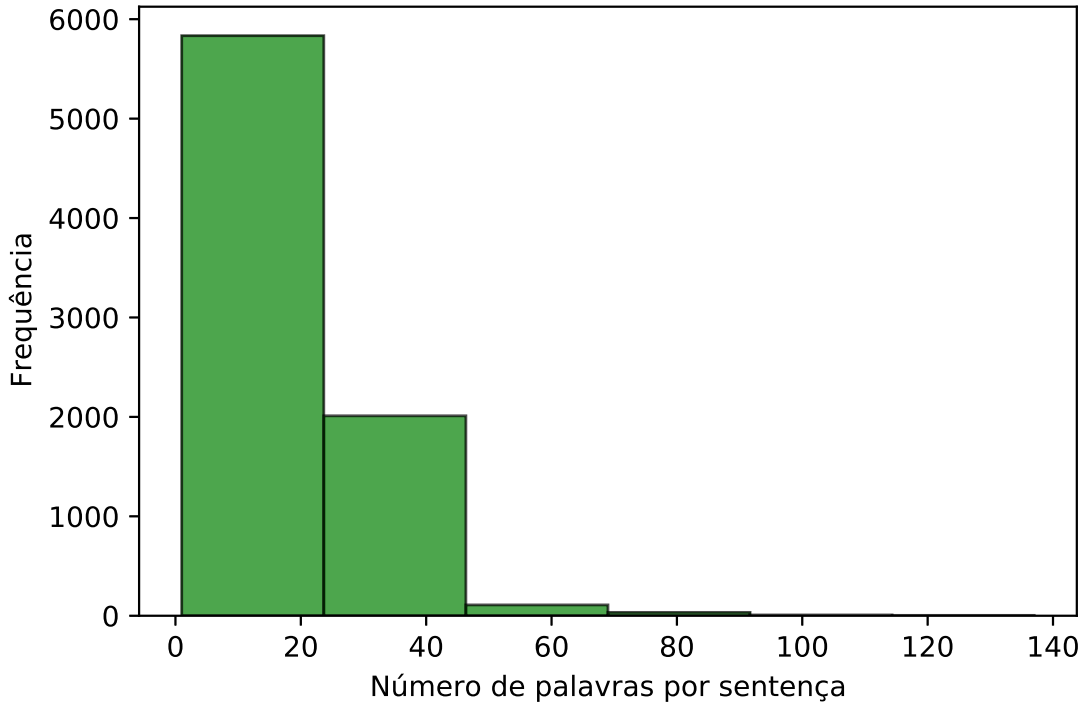

Figura 4.5 – Evalita 2018: Histograma de número de palavras por sentença.

Fonte: Elaborado pelo autor, 2022.

4.2.3 Corpus *Hate Speech Dataset* - HSD

Fortuna et al. (2019) desenvolveram o *Hate Speech Dataset* (HSD), um conjunto de dados anotado de forma hierárquica composto por micro-textos em português que foram extraídos do Twitter. Foram obtidos 42.390 *tweets* ao todo, mas o número foi reduzido para 5.670 após a realização de pré-processamentos. Dois juízes anotaram os dados. Apesar do HSD ter sido anotado com hierarquia de classes de ódio (sexismo, racismo, dentre outros), estes também possuem classificação binária. Dos 5.670 *tweets*, 1.228 foram classificados como discurso de ódio - 22% do conjunto de dados, e os 4.442 restantes (88%), como neutros.

No Quadro 4.3, tem-se uma amostra deste corpus. A Figura 4.6 mostra a nuvem de palavras geradas para o corpus HSD. Uma vez que o corpus foi coletado com textos sobre tipos de discurso de ódio variados, vê-se que há uma grande diversidade nas palavras mais frequentes. É possível observar palavras como 'mulher' e 'burra' - que são termos que podem estar relacionados a sexismo -, e também termos como 'refugiados' - um termo que pode estar atrelado a xenofobia, e 'branco' - termo que pode ser usado para combater racismo. O histograma gerado (Figura 4.7) mostra que a maioria das sentenças neste corpus possui entre 20 e 30 palavras, o que mostra textos maiores em comparação com o corpus OffComBr-2 (PELLE; MOREIRA, 2017), por exemplo.

4.2.4 Corpus WH

Waseem e Hovy (2016) apresentaram um corpus (a ser chamado WH a partir de então) com cerca de 16.000 *tweets*, dos quais 3.383 foram anotados como tendo conteúdo sexista e 1.972, como tendo conteúdo racista. Os *tweets* sexistas e racistas foram juntados em um só conjunto nesta pesquisa. Assim, no total 5.355 *tweets* foram considerados como discurso de ódio, sendo equivalente a 33% do total do corpus. Os autores realizaram uma busca manual inicial de injúrias e termos usados para se referir a minorias religiosas, sexuais, de gênero e/ou étnicas. Foram identificados termos frequentes que contêm referências a entidades específicas, como "#MKR", a *hashtag* do programa de TV australiano *My Kitchen Rules*, que muitas vezes gera *tweets* sexistas direcionados a mulheres. Os *tweets* foram anotados pelos próprios autores e por uma anotadora externa.

O Quadro 4.4 apresenta uma amostra do corpus WH. As palavras mais frequentes deste corpus podem ser vistas na Figura 4.8. 'Kat' foi o termo mais citado, seguido por 'like', 'women' e 'sexist'. 'Kat' foi uma participante do programa MKR em 2015, que

demonstrava ter problemas de temperamento e foi uma das maiores vítimas de comentários sexistas pelos telespectadores. O histograma de número de palavras por sentença (Figura 4.9) mostra que a maior parte das sentenças do corpus WH possui entre 20 e 40 palavras, apresentando um comportamento diferente dos outros corpora que contêm *tweets*.

Figura 4.6 – HSD: Nuvem de palavras.

Fonte: Elaborado pelo autor, 2022.

Quadro 4.3 – Amostra do corpus HSD.

Texto original	**Discurso de ódio**
'SOU MOCIDADE INDEPENDENTE' FALOU O MACONHERO, 32 ANOS, SÓ TEXTÃO NO FEICE E MORA COM A MÃE... NÃO É MOCIDADE MUITO MENOS INDEPENDENTE	Sim
Vim trabalhar com uma camiseta da minha namorada hoje, tô me sentindo bem sapatão	Não
Totalitários, cobardes-queixinhas! Nojento @user vive dos nossos impostos e é racista anti-nacional e branco.	Sim
VAI TOMAR NO **!!! JUDICIÁRIO É A VERGONHA DO BRASIL!!! REVOLTANTE!!	Sim
VOU PROCURAR MINHA FOTO CM ELE MAIS EU ERA MT FEIA E GORDA	Não

Fonte: Elaborado pelo autor, 2022.

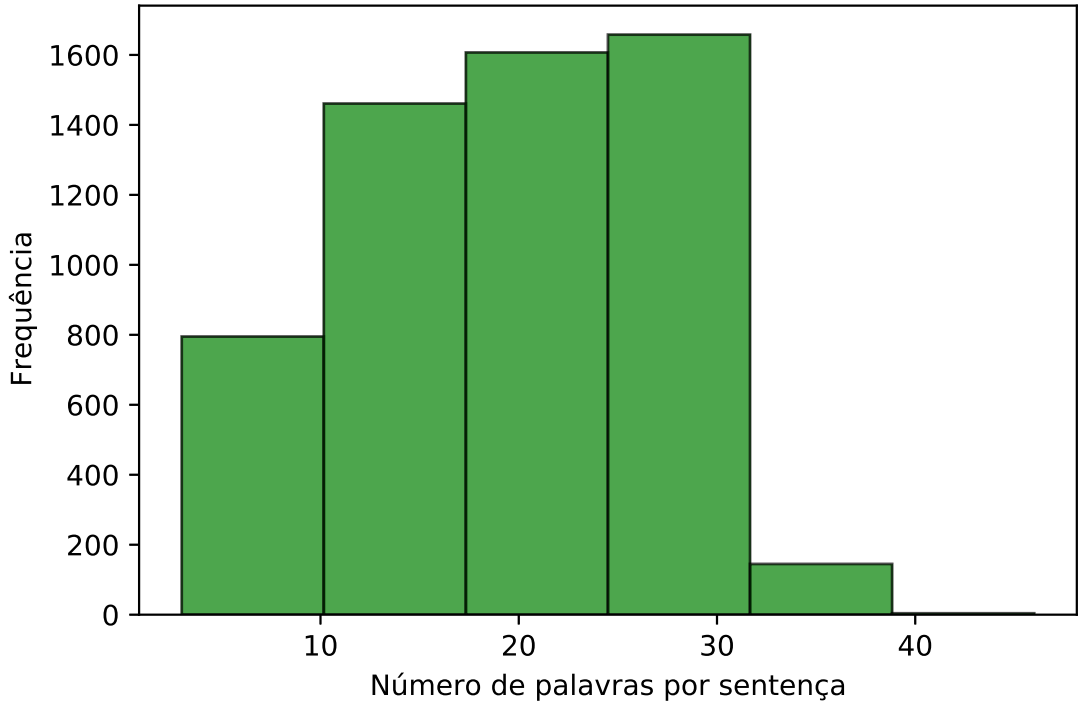

Figura 4.7 – HSD: Histograma de número de palavras por sentença.

Fonte: Elaborado pelo autor, 2022.

Quadro 4.4 – Amostra do corpus WH.

Texto original	**Tradução**	**Discurso de ódio**
@user Mosque that is still standing. Typical Muslim terrorist asshole.	@user Mesquita que ainda está de pé. Típico idiota terrorista muçulmano.	Sim
Not sure if I'm speaking there this year or not, but they were the first conference at which I ever presented.	Não tenho certeza se vou falar lá este ano ou não, mas foi a primeira conferência em que apresentei	Não
Wish these blondes were in that How To Get Away With Murder show....#MKR	Queria que essas loiras estivessem naquela série 'How To Get Away With Murder'....#MKR	Sim
@user that's why they really can't effect us. gaming industry is involved, sure. But it's a much bigger conversation.	@user é por isso que eles realmente não podem nos afetar. indústria de jogos está envolvida, com certeza. Mas é uma conversa muito maior.	Não
#mkr deconstructed by girls that have deconstructed brains ! Nearly brought up my dinner when I saw that crap on the plate	#mkr desconstruído por garotas que têm cérebros desconstruídos! Quase vomitei meu jantar quando vi aquela porcaria no prato	Sim

Fonte: Elaborado pelo autor, 2022.

Figura 4.8 – WH: Nuvem de palavras.

Fonte: Elaborado pelo autor, 2022.

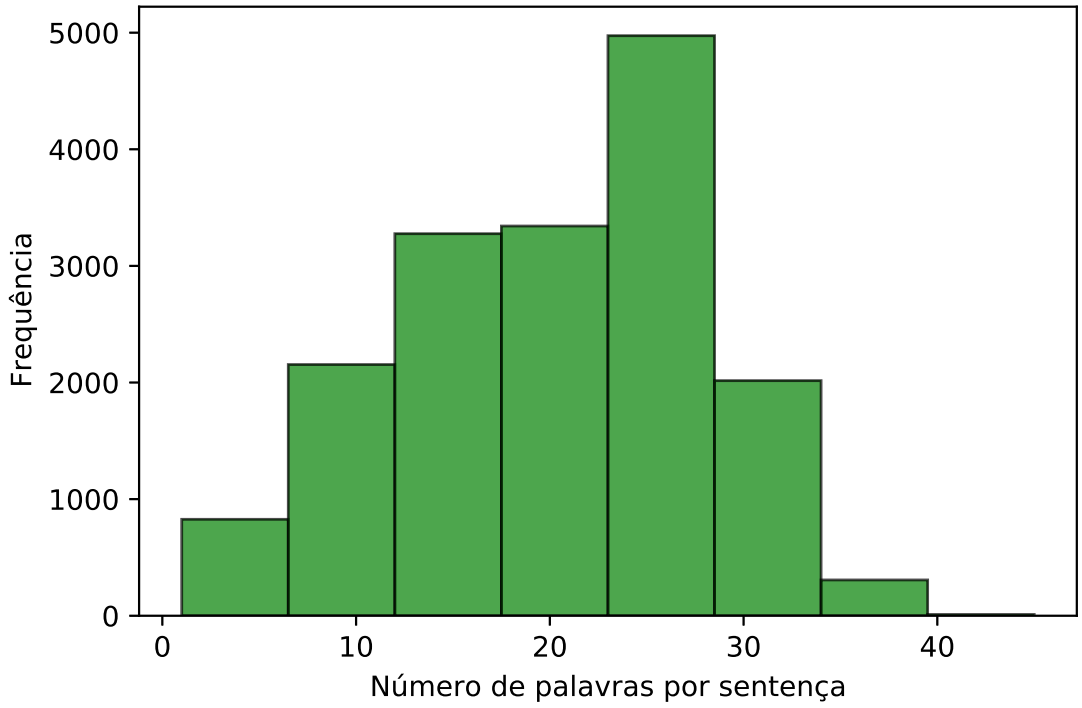

Figura 4.9 – WH: Histograma de número de palavras por sentença.

Fonte: Elaborado pelo autor, 2022.

Na Tabela 4.1, tem-se um resumo das principais estatísticas dos corpora utilizados. É possível observar que os corpora são de tamanhos diferentes, e que a proporção de classes também difere neles. O corpus Evalita 2018 (Facebook) contém dados balanceados, enquanto os demais corpora (OffComBr-2, HSD, WH e Evalita 2018 (Twitter)) contêm majoritariamente textos neutros (correspondendo ao comportamento do mundo real).

Tabela 4.1 – Estatísticas dos corpora utilizados.

Corpus	**Textos neutros**	**Textos com discurso de ódio**	**Total**	**Fonte**	**Contexto**
OffComBr-2	831 (67,5%)	419 (32,5%)	1.250	g1.globo.com	Política e Esportes
Evalita 2018 (Facebook)	1.941 (48,5%)	2.059 (51,5%)	4.000	Facebook	Política
Evalita 2018 (Twitter)	2.704 (67%)	1.296 (33%)	4.000	Twitter	Religião e Xenofobia
HSD	3.882 (69%)	1.788 (31%)	5.670	Twitter	Geral
WH	5.038 (62%)	3.098 (38%)	8.136	Twitter	Sexismo e Racismo

Fonte: Elaborado pelo autor, 2022.

4.2.5 Considerações Finais do Capítulo

Neste capítulo foi apresentada a metodologia desenvolvida para detecção de discurso de ódio em português utilizando CLL. Além disso, foram apresentados os corpora utilizados nesta pesquisa. Foram mostradas informações relevantes sobre cada corpus, como as palavras mais recorrentes e o número de palavras por sentença. No próximo capítulo, serão detalhados os experimentos realizados e os resultados obtidos serão discutidos.

5 AVALIAÇÃO EXPERIMENTAL

Este capítulo apresenta os experimentos para avaliar a metodologia proposta para detecção de discurso de ódio de textos em português. Além disso, os resultados dos experimentos realizados são discutidos e comparados com trabalhos existentes na literatura.

5.1 DESCRIÇÃO DOS EXPERIMENTOS

Para investigar e avaliar os problemas de pesquisa conforme a metodologia proposta, foram realizados diversos experimentos com cada uma das estratégias de treinamento. Foram realizados seis experimentos. Em todos os experimentos, foram utilizadas as estratégias experimentais citadas na Seção 4.1:

- ZST: Zero-Shot Transfer;
- JL: Joint Learning;
- CL: Cascade Learning;
- CL/JL;
- CL/JL+.

O primeiro experimento, a ser chamado Evalita/OffComBr-2, consistiu na utilização de quatro MLPTs e das cinco estratégias de experimentação descritas no Capítulo 4.1. Para este experimento, utilizou-se o corpus Evalita (em italiano) como I_f e o corpus OffComBr-2 (em português) como I_a. No segundo experimento - estudo de ablação -, utilizou-se BERTimbau e XLM-R, e as estratégias ZST e CL. Os corpora Evalita e OffComBr-2 foram utilizados neste experimento.

O terceiro experimento - balanceamento de dados - foi realizado com a utilização dos MLPTs XLM-R e BERTimbau, e utilizaram-se os corpora Evalita e OffComBr-2. As estratégias de treinamento utilizadas no terceiro experimentos foram JL e CL. No quarto experimento, a ser chamado Evalita/HSD, consistiu na utilização dos MLPTs XML-R e BERTimbau e dos corpora Evalita (Facebook e Twitter) como I_f e HSD (em português) como I_a. Todas as estratégias de treinamento foram utilizadas no quarto experimento.

No quinto experimento - WH/OffComBr-2, foram utilizados os MLPT XLM-R e BERTimbau. Todas as estratégias de treinamento foram usadas, e o corpus WH (em inglês)

foi utilizado como I_f, e o corpus OffComBr-2, como I_a. No sexto e último experimento (WH/OffComBr-2-EN) consistiu na utilização das cinco estratégias de treinamento e dos MLPTs BERT e XLM-R. Neste experimento, utilizaram-se os corpora WH e o corpus OffComBr-2-EN (originalmente em português traduzido para a língua inglesa).

As mesmas configurações foram adotadas para todos os MLPT utilizados nesta tese. Para a execução dos experimentos, utilizou-se uma taxa de aprendizado de 1×10^{-5} e número de épocas igual a 3 (DEVLIN et al., 2019; CONNEAU et al., 2020). O otimizador utilizado foi o AdamW com epsilon $= 1 \times 10^{-8}$. Como função de perda, foi utilizada a função de entropia cruzada binária, e softmax como função de ativação.

Em todos os experimentos, utilizou-se o Google Colab com uma GPU Nvidia Tesla K80. Para a execução dos experimentos com os MLPTs adotados, utilizou-se a biblioteca PyTorch.

Tendo em vista que os MLPTs usados aceitam apenas entradas de tamanho fixo, fez-se necessário definir um tamanho máximo para os dados utilizados. Então, realizou-se o mapeamento das entradas em um vetor de dimensão igual a 128. Nos casos em que as entradas eram menores que este valor, um preenchimento com 0's foi feito; nos casos em que as entradas eram maiores, foi feito um truncamento para o tamanho adequado, ignorando os elementos com índice maiores que 128. Nos corpora utilizados, houve apenas um registro que ultrapassou o limite de 128 palavras.

Para todos os casos, ao utilizar a estratégia ZST, no treinamento dos MLPTS utilizou-se 90% dos dados do corpus I_f e os 10% restantes foram usados para validação. Na etapa de teste, apenas o corpus do idioma I_a foi utilizado. Já quando a estratégia JL foi utilizada, utilizou-se o corpus do idioma I_f com a mesma distribuição usada na abordagem ZST (90% para treinamento e 10% para validação). Além disso, um subconjunto do corpus do idioma I_a foi utilizado no treinamento, e o modelo foi testado usando apenas os dados restantes do corpus do idioma I_a. A porcentagem de dados do corpus do idioma I_a a serem adicionados ao treinamento dos MLPTs variou de 10 a 70% nos experimentos realizados. Os melhores resultados foram obtidos quando 30% do corpus do idioma I_a foi adicionado no treinamento.

Quando a estratégia CL foi usada, utilizou-se apenas o corpus do idioma I_f com a seguinte divisão: 70% dos dados treinamento, 10% para a etapa de validação e 20% para a etapa de teste. Depois disso, foi realizado um ajuste fino dos MLPTs. Desta vez, apenas dados do idioma I_a foram utilizados. Seguiu-se então o mesmo padrão utilizado na primeira etapa, com 70% dos dados do idioma I_a sendo usados para treinamento, 10% para validação e 20% para testes.

Na utilização da estratégia CL/JL, utilizou-se 70% dos dados do corpus do idioma I_f juntamente com 30% dos dados do corpus do idioma I_a. Para a validação, 20% dos dados do idioma I_f foram utilizados, somados com 10% dos dados do idioma I_a e o restante dos dados do idioma I_f foram usados no teste. O restante dos dados do idioma I_a foram utilizados para o ajuste fino dos MLPTs. Nesta etapa, com os dados restantes seguiu-se a mesma divisão utilizada na primeira etapa (70/20/10).

Por fim, nos experimentos em que a estratégia CL/JL+ foi utilizada, adotou-se a mesma divisão da estratégia anterior no treinamento inicial, em que tem-se 70% dos dados do idioma I_f somados com 30% dos dados do idioma I_a para o treinamento. O mesmo se aplica nas etapas de validação e teste. Para os ajustes finos posteriores, o corpus do idioma I_a restante foi dividido de acordo com o número de ajustes finos realizados, utilizando validação cruzada *k-fold* para garantir a proporcionalidade das classes ao longo das iterações.

Para todos os testes de significância realizados, utilizou-se um nível de confiança de 95% (alfa = 0.05). Em todos os casos, considera-se que a hipótese nula pode ser rejeitada caso o *epsilon* mínimo seja menor que 0.5.

Nas subseções a seguir, serão exibidos os resultados dos experimentos realizados para detecção de discurso de ódio usando CLL.

5.1.1 Experimento 1: Evalita/OffComBr-2

O primeiro experimento consistiu em utilizar o corpus Evalita (Facebook) (BOSCO et al., 2018) - italiano - como idioma fonte I_f e o corpus OffComBr-2 (PELLE; MOREIRA, 2017) - português - como idioma alvo I_a. Para este experimento foram utilizados como MLPTs: o BERTimbau (SILVA, 2020), o BERT italiano (SCHWETER, 2020), o BERT (DEVLIN et al., 2019) e o XLM-R (CONNEAU et al., 2020).

A primeira parte deste experimento consistiu em comparar o desempenho dos MLPTs XLM-R (CONNEAU et al., 2020) e BERT (DEVLIN et al., 2019) quanto às versões base e grande, utilizando a estratégia de treinamento ZST. Na Tabela 5.1, tem-se os resultados destes experimentos. É importante ressaltar que o corpus OffComBr-2 é desbalanceado (Tabela 4.1), possuindo cerca de 68% dos dados sendo neutros, e o restante sendo discurso de ódio.

Tabela 5.1 – Experimento 1: Resultados da utilização da estratégia ZST comparando as versões do XLM-R e BERT.

MLPT	Precisão	Revocação	Medida F1
BERT (base)	51%	48%	49%
XLM-R (base)	58%	62%	60%
BERT (grande)	62%	54%	58%
XLM-R (grande)	72%	71%	**71%**

Fonte: Elaborado pelo autor, 2022.

Ainda sobre a primeira parte do experimento 1, foram feitos testes de significância quanto ao uso das versões base e grande dos MLPTs BERTimbau e XLM-R. Os resultados para estes testes de significância podem ser vistos na Tabela 5.2. Uma vez que o *epsilon* mínimo foi menor que 0.5 em ambos os casos, rejeita-se a hipótese nula e assume-se que o desempenho dos modelos grandes foram estocasticamente dominantes sobre o desempenho dos base.

Tabela 5.2 – Resultados dos testes de hipóteses sobre a utilização dos modelos grandes nos MLPTs BERT e XLM-R (experimento 1).

Hipótese	***Epsilon* mínimo**
H0-0: O desempenho do BERT grande não é superior ao do modelo base	0
H0-1: O desempenho do BERT grande é superior ao do modelo base	
H1-0: O desempenho do XLM-R grande não é superior ao do modelo base	0
H1-1: O desempenho do XLM-R grande é superior ao do modelo base	

Na Tabela 5.3 são apresentados os resultados da utilização da estratégia ZST para os MLPTs BERTimbau, BERT italiano, BERT e XLM-R. Tendo em vista que houve melhora significativa ao utilizar o modelo grande em detrimento do base, optou-se pelo primeiro em cada um dos MLPTs.

Tabela 5.3 – Experimento 1: Resultados da utilização da estratégia ZST comparando todos os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	72%	71%	**71%**
BERTimbau	70%	61%	66%
BERT italiano	59%	62%	60%
BERT	60%	54%	58%

Fonte: Elaborado pelo autor, 2022.

Sobre a estratégia de treinamento JL, na Tabela 5.4 são apresentados os resultados da utilização usando esta estratégia para os quatro MLPTs já citados anteriormente.

Tabela 5.4 – Experimento 1: Resultados da utilização da estratégia JL comparando todos os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	77%	78%	**77%**
BERTimbau	77%	75%	76%
BERT italiano	63%	58%	61%
BERT	65%	52%	56%

Fonte: Elaborado pelo autor, 2022.

Os resultados da utilização da estratégia CL são apresentados na Tabela 5.5. Mais uma vez, os mesmos MLPTs foram utilizados aqui. No treinamento, foi utilizado apenas o corpus Evalita. Já no ajuste fino, apenas o corpus OffComBr-2 foi usado.

Tabela 5.5 – Experimento 1: Resultados da utilização da estratégia CL comparando todos os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	82%	79%	80%
BERTimbau	83%	84%	**83%**
BERT italiano	75%	74%	74%
BERT	72%	73%	72%

Fonte: Elaborado pelo autor, 2022.

Na Tabela 5.6 são apresentados os resultados da utilização da estratégia CL/JL para os quatro MLPTs. Uma parte dos dados do corpus OffComBr-2 é utilizado também no treinamento dos MLPTs, juntamente com o corpus Evalita. No ajuste fino, apenas o restante do corpus OffComBr-2 é utilizado.

Tabela 5.6 – Experimento 1: Resultados da utilização da estratégia CL/JL comparando todos os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	83%	81%	82%
BERTimbau	84%	85%	**84%**
BERT italiano	79%	78%	78%
BERT	73%	75%	74%

Fonte: Elaborado pelo autor, 2022.

Utilizando a abordagem CL/JL+, foi experimentado o uso de mais de dois ajustes finos nos MLPTs. Para este sub-experimento, o corpus OffComBr-2 foi dividido de acordo

com o número de ajustes finos realizados, utilizando validação cruzada *k-fold* para garantir a proporcionalidade das classes ao longo das iterações. O MLPT utilizado aqui foi apenas o XLM-R. A Tabela 5.7 apresenta os resultados deste sub-experimento.

Tabela 5.7 – Experimento 1: Resultados da utilização da estratégia CL/JL+ em relação ao número de ajustes finos.

Número de iterações	Precisão	Revocação	Medida F1
3	87%	83%	85%
4	88%	86%	87%
5	92%	89%	**90%**

Fonte: Elaborado pelo autor, 2022.

Na Tabela 5.8 são apresentados os resultados da utilização da estratégia CL/JL+ para os quatro MLPTs utilizados com cinco ajustes finos.

Tabela 5.8 – Experimento 1: Resultados da utilização da estratégia CL/JL+ comparando todos os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	92%	89%	90%
BERTimbau	92%	93%	**92%**
BERT italiano	83%	80%	81%
BERT	80%	81%	80%

Fonte: Elaborado pelo autor, 2022.

Os resultados para os quatro MLPTs utilizados com cinco ajustes finos adicionais estão listados na Tabela 5.8. O resultado apresentado pelo MLPT BERTimbau atingiu uma medida F1 de 92%, alcançando o melhor resultado do estado da arte entre os trabalhos que usaram o corpus OffComBr2 (PELLE; MOREIRA, 2017). Silva et al. (2019), Lima e Bianco (2019) e Pari et al. (2019) também utilizaram o corpus OffComBr2 (PELLE; MOREIRA, 2017). Na Tabela 5.9, é apresentada uma comparação com a medida F1 desses trabalhos.

Tabela 5.9 – Experimento 1: Comparação com o estado da arte.

Trabalho	Medida F1
Lima e Bianco (2019)	72%
Baseline - Pelle e Moreira (2017)	77%
Pari et al. (2019)	86%
Silva et al. (2019)	89%
Metodologia proposta	**92%**

Fonte: Elaborado pelo autor, 2022.

Também foram feitos testes de significância sobre os resultados da abordagem proposta e de trabalhos relacionados. Comparou-se apenas o resultado da abordagem proposta com os segundo e terceiro melhores resultados (SILVA et al., 2019; PARI et al., 2019), visto que a diferença entre o desempenho da abordagem proposta para os demais trabalhos é maior (mais de 10%). Foram feitos dois testes unilaterais, e o *epsilon* mínimo obtido em ambos foi menor que 0.5, logo rejeita-se a hipótese nula e assume-se que o desempenho da abordagem proposta foi estocasticamente dominantes sobre o desempenho de Silva et al. (2019), de Pari et al. (2019) e, por consequência, dos demais trabalhos (Tabela 5.10).

Tabela 5.10 – Resultados dos testes de hipóteses sobre o desempenho dos trabalhos que utilizaram o corpus OffComBr-2 (experimento 1).

Hipótese	***Epsilon* mínimo**
H0-0: O desempenho da abordagem proposta não é superior ao de Silva et al. (2019)	0
H0-1: O desempenho da abordagem proposta é superior ao de Silva et al. (2019)	
H1-0: O desempenho da abordagem proposta não é superior ao de Pari et al. (2019)	0
H1-1: O desempenho da abordagem proposta é superior ao de Pari et al. (2019)	

5.1.2 Experimento 2: Estudos de ablação

Para o segundo experimento, utilizou-se como MLPTs o XLM-R e o BERTimbau. Um estudo de ablação consiste em remover um componente de um sistema para entender melhor o seu funcionamento ((MEYES et al., 2019)). Apenas o corpus OffComBr-2 foi utilizado neste experimento. Um dos objetivos deste trabalho é entender se a utilização de CLL traz algum benefício na metodologia proposta. Para isto, foram realizados experimentos sem utilizar um idioma I_a. Ou seja, removeu-se a utilização de um corpus auxiliar na metodologia. Assim, o corpus OffComBr-2 foi dividido de acordo com as abordagens empregadas.

Neste experimento, foram utilizadas as estratégias de treinamento ZST e CL. No caso da utilização da estratégia CL, parte dos dados foi usada para realizar o treinamento dos MLPTs, e a outra parte foi guardada para o ajuste fino e para teste. Os resultados deste experimento podem ser conferidos na Tabela 5.11.

Nas Figuras 5.1 e 5.2, é possível observar as diferenças entre os resultados dos experimentos com e sem a utilização de CLL. Especialmente, o MLPT XLM-R demonstra que o uso de CLL trouxe grandes ganhos em relação ao seu desempenho, pois é possível

visualizar a disparidade dos resultados quanto ao uso de CLL.

Tabela 5.11 – Experimento 2: Resultados da utilização das estratégia ZST e CL.

MLPT	Estratégia	Usou CLL?	Precisão	Revocação	Medida F1
XLM-R	ZST	Sim	72%	71%	**71%**
		Não	50%	44%	48%
	CL	Sim	82%	79%	**80%**
		Não	46%	68%	55%
BERTimbau	ZST	Sim	70%	61%	**66%**
		Não	51%	50%	50%
	CL	Sim	83%	84%	**83%**
		Não	62%	59%	60%

Fonte: Elaborado pelo autor, 2022.

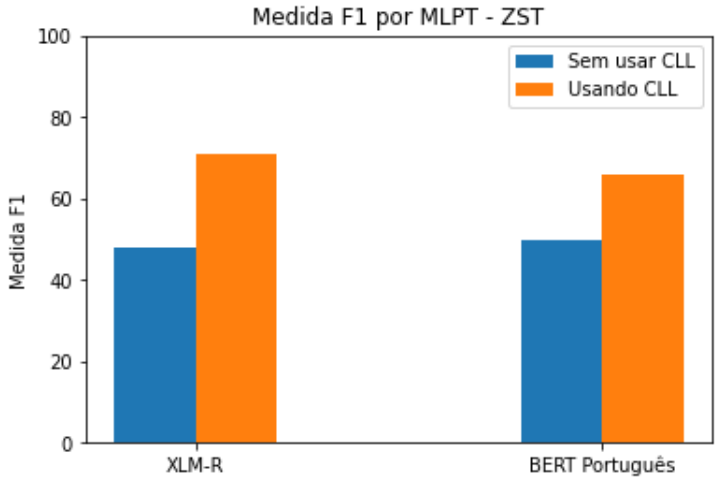

Figura 5.1 – Experimento 2: Resultados da utilização da estratégia ZST

Fonte: Elaborado pelo autor, 2022.

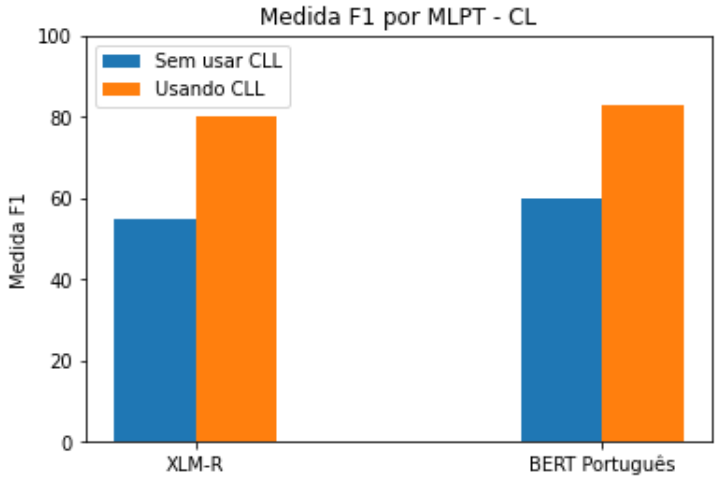

Figura 5.2 – Experimento 2: Resultados da utilização da estratégia CL

Fonte: Elaborado pelo autor, 2022.

Ademais, foram feitos testes de significância acerca do desempenho dos MLPTs XLM-R e BERTimbau quanto ao uso de CLL. Foram realizados dois testes unilaterais para cada MLPT: um sobre a estratégia de treinamento ZST, e outro sobre a estratégia CL. Para todos os casos, o *epsilon* mínimo obtido foi 0. Deste modo, rejeita-se a hipótese nula e assume-se que o desempenho dos MLPTs quando usando CLL é estocasticamente dominante sobre não utilizar CLL. Na Tabela 5.12, tem-se as hipóteses e os respectivos resultados.

Tabela 5.12 – Resultados dos testes de hipóteses sobre os estudos de ablação (experimento 2).

MLPT	**Hipótese**	***Epsilon* mínimo**
XLM-R	H0-0: O desempenho da abordagem proposta utilizando CLL na abordagem ZST não é superior ao da não utilização de CLL	0
	H0-1: O desempenho da abordagem proposta utilizando CLL na abordagem ZST é superior ao da não utilização de CLL	
	H1-0: O desempenho da abordagem proposta utilizando CLL na abordagem CL não é superior ao da não utilização de CLL	
	H1-1: O desempenho da abordagem proposta utilizando CLL na abordagem CL é superior ao da não utilização de CLL	
BERTimbau	H2-0: O desempenho da abordagem proposta utilizando CLL na abordagem ZST não é superior ao da não utilização de CLL	0
	H2-1: O desempenho da abordagem proposta utilizando CLL na abordagem ZST é superior ao da não utilização de CLL	
	H3-0: O desempenho da abordagem proposta utilizando CLL na abordagem CL não é superior ao da não utilização de CLL	
	H3-1: O desempenho da abordagem proposta utilizando CLL na abordagem CL é superior ao da não utilização de CLL	

5.1.3 Experimento 3: Balanceamento de dados

Este experimento consistiu em verificar se o balanceamento de dados impactaria no desempenho dos MLPTs na metodologia proposta. Como visto na Tabela 4.1, o corpus mais utilizado nesta pesquisa (OffComBr-2) apresenta desbalanceamento entre as classes,

tendo aproximadamente 70% dos dados pertencendo à classe neutra, e os 30% restantes pertencendo à classe de discurso de ódio.

Neste experimento, foram utilizados como MLPTs o BERTimbau e o XLM-R. As estratégias de treinamento utilizadas foram JL e CL. Aqui, novamente utilizou-se o corpus Evalita (Facebook) como idioma fonte I_f, e o corpus OffComBr-2 como idioma alvo I_a.

O experimento foi dividido em duas partes. A primeira delas consistiu em deixar o corpus OffComBr-2 na mesma proporção do corpus Evalita. Para tanto, a técnica de subamostragem foi utilizada para realizar o balanceamento dos dados e foram removidas instâncias da classe majoritária. Assim, a proporção entre as classes do corpus OffComBr-2 ficou igual à proporção entre as classes do corpus Evalita.

A segunda parte do experimento consistiu em deixar os dados do corpus Evalita na mesma proporção do corpus OffComBr-2. Novamente, utilizou-se de subamostragem para remover instâncias da classe positiva para ter um balanceamento equivalente a 70/30, no qual a classe majoritária é a neutra (sem discurso de ódio).

Na Tabela 5.13 são exibidos os resultados do terceiro experimento. Os resultados da primeira parte deste experimento estão listados nas linhas em que a proporção dos dados consta como 50%/50%. Nestes casos, os dados do corpus OffComBr-2 seguem a mesma proporção dos dados do corpus Evalita.

Já os resultados da segunda parte deste experimento estão representados nas linhas em que a proporção dos dados está como 70%/30%. Neste experimento, o corpus Evalita seguiu a mesma proporção do corpus OffComBr-2.

Tabela 5.13 – Experimento 3: Resultados da utilização das estratégia ZST e CL.

MLPT	Estratégia	Proporção dos dados	Precisão	Revocação	Medida F1
XLM-R	JL	Original	77%	78%	**77%**
		50/50	75%	75%	75%
		70/30	74%	74%	74%
	CL	Original	82%	79%	**80%**
		50/50	80%	79%	79%
		70/30	79%	77%	78%
BERTimbau	JL	Original	77%	75%	**76%**
		50/50	77%	75%	**76%**
		70/30	76%	76%	**76%**
	CL	Original	83%	84%	**83%**
		50/50	83%	82%	82%
		70/30	80%	81%	81%

Fonte: Elaborado pelo autor, 2022.

Para o experimento 3, foram realizados testes de significância a fim de comparar o desempenho dos MLPTs quanto ao balanceamento dos corpora utilizados (Tabela 5.14 e Tabela 5.15). Foi comparado o desempenho dos MLPTs XLM-R e BERTimbau, e foram comparadas as três distribuições de dados citadas nos experimentos: original, 70% para a classe neutra e 30% para a positiva, e 50% para ambas as classes. Para todos os casos com o XLM-R (Tabela 5.14), o *epsilon* mínimo foi menor do que 0.5, dando indícios de que a hipótese nula deve ser rejeitada. Então, assume-se que o desempenho deste MLPT nas estratégias de treinamento usadas, com as distribuições originais dos corpora usados foram estocasticamente dominantes sobre o desempenho com a utilização de dados balanceados.

Já no caso dos resultados com o BERTimbau, em alguns casos não houve indícios para refutar a hipótese nula. Conforme vê-se na Tabela 5.15, nos experimentos JL o *epsilon* mínimo obtido foi maior do que 0.5 para as duas distribuições de balanceamento. Nestes casos, não se pôde refutar a hipótese nula. O desempenho do BERTimbau não foi melhorado ao usar-se a distribuição original dos dados. Porém, nos experimentos CL, foi possível rejeitar as hipóteses nulas e assumir o desempenho utilizando a distribuição original dos dados como sendo estocasticamente dominante sobre a utilização de dados balanceados.

Tabela 5.14 – Resultados dos testes de hipóteses sobre os estudos de balanceamento de dados com o MLPT XLM-R (experimento 3).

Estratégia	**Hipótese**	***Epsilon* mínimo**
JL	H0-0: O desempenho da abordagem proposta utilizando a proporção original dos corpora não é superior ao da utilização da divisão 50/50	0
	H0-1: O desempenho da abordagem proposta utilizando a proporção original dos corpora é superior ao da utilização da divisão 50/50	
	H1-0: O desempenho da abordagem proposta utilizando a proporção original dos corpora não é superior ao da utilização da divisão 70/30	0
	H1-1: O desempenho da abordagem proposta utilizando a proporção original dos corpora é superior ao da utilização da divisão 70/30	
CL	H2-0: O desempenho da abordagem proposta utilizando a proporção original dos corpora não é superior ao da utilização da divisão 50/50	0.1326
	H2-1: O desempenho da abordagem proposta utilizando a proporção original dos corpora é superior ao da utilização da divisão 50/50	
	H3-0: O desempenho da abordagem proposta utilizando a proporção original dos corpora não é superior ao da utilização da divisão 70/30	0
	H3-1: O desempenho da abordagem proposta utilizando a proporção original dos corpora é superior ao da utilização da divisão 70/30	

Tabela 5.15 – Resultados dos testes de hipóteses sobre os estudos de balanceamento de dados com o MLPT BERTimbau (experimento 3).

Estratégia	Hipótese	*Epsilon* mínimo
JL	H4-0: O desempenho da abordagem proposta utilizando a proporção original dos corpora não é superior ao da utilização da divisão 50/50	0.5648
	H4-1: O desempenho da abordagem proposta utilizando a proporção original dos corpora é superior ao da utilização da divisão 50/50	
	H5-0: O desempenho da abordagem proposta utilizando a proporção original dos corpora não é superior ao da utilização da divisão 70/30	0.5535
	H5-1: O desempenho da abordagem proposta utilizando a proporção original dos corpora é superior ao da utilização da divisão 70/30	
CL	H6-0: O desempenho da abordagem proposta utilizando a proporção original dos corpora não é superior ao da utilização da divisão 50/50	0
	H6-1: O desempenho da abordagem proposta utilizando a proporção original dos corpora é superior ao da utilização da divisão 50/50	
	H7-0: O desempenho da abordagem proposta utilizando a proporção original dos corpora não é superior ao da utilização da divisão 70/30	0
	H7-1: O desempenho da abordagem proposta utilizando a proporção original dos corpora é superior ao da utilização da divisão 70/30	

5.1.4 Experimento 4: Evalita/HSD

O quarto experimento consistiu em utilizar os corpora Evalita (Facebook + Twitter) como idioma fonte I_f e o corpus HSD - português - como idioma alvo I_a. Para este experimento foram utilizados apenas dois MLPTs: BERTimbau e XLM-R, por estes terem apresentado um melhor desempenho no experimento 1. Além disso, utilizou-se todas as estratégias de treinamento neste experimento.

Para este experimento, não foi possível utilizar apenas o corpus Evalita (Facebook) como idioma fonte I_f. Isso porque a principal motivação do uso de CLL é ter um corpus com muitos dados anotados para ser utilizado na etapa de treinamento, e ter um corpus com poucos dados para avaliar o modelo treinado. No caso dos corpora utilizados aqui, esta premissa não teria validade, uma vez que corpus HSD é maior que o corpus Evalita (Facebook) (Tabela 4.1) - o corpus Evalita (Facebook) contém apenas 4.000 dados, enquanto o corpus HSD tem 5.670 dados.

Então, adotou-se a estratégia de incrementar o corpus Evalita (Facebook). Para tanto, foram acrescentados mais dados do mesmo evento (BOSCO et al., 2018). Os autores disponibilizaram um corpus contendo 4.000 postagens do Facebook e 4.000 tweets. O corpus Evalita (Facebook) foi composto apenas pelas postagens do Facebook. Optou-se então por mesclar os corpora Evalita (Facebook) e Evalita (Twitter) em um só, tendo assim um corpus resultante com 8.000 textos. Deste modo, utilizou-se este corpus como idioma fonte I_f e o corpus HSD como idioma alvo I_a.

Na Tabela 5.16 são exibidos os resultados do experimento utilizando a estratégia ZST para os dois MLPTs utilizados. Na Tabela 5.17, podem ser vistos os resultados da utilização da estratégia JL, em que uma parte do corpus do HSD é adicionado ao treinamento.

Tabela 5.16 – Experimento 4: Resultados da utilização da estratégia ZST comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	72%	74%	**70%**
BERTimbau	68%	69%	68%

Fonte: Elaborado pelo autor, 2022.

Tabela 5.17 – Experimento 4: Resultados da utilização da estratégia JL comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	77%	78%	**77%**
BERTimbau	76%	77%	76%

Fonte: Elaborado pelo autor, 2022.

Os resultados da utilização da estratégia CL estão listados na Tabela 5.18. Já os resultados da utilização da estratégia CL/JL podem ser vistos na Tabela 5.19. É importante ressaltar que nesta última estratégia, parte dos dados do corpus HSD também são adicionados no treinamento do modelo.

Tabela 5.18 – Experimento 4: Resultados da utilização da estratégia CL comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	77%	77%	77%
BERTimbau	80%	80%	**80%**

Fonte: Elaborado pelo autor, 2022.

Tabela 5.19 – Experimento 4: Resultados da utilização da estratégia CL/JL comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	80%	81%	80%
BERTimbau	82%	82%	**82%**

Fonte: Elaborado pelo autor, 2022.

A última estratégia utilizada neste experimento foi a CL/JL+, na qual se tem cinco ajustes finos adicionais nos MLPTs. Os resultados podem ser vistos na Tabela 5.20.

Tabela 5.20 – Experimento 4: Resultados da utilização da estratégia CL/JL+ comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	83%	82%	82%
BERTimbau	90%	91%	**90%**

Fonte: Elaborado pelo autor, 2022.

Na Tabela 5.21, são mostrados os resultados de trabalhos relacionados que também utilizaram o corpus HSD para realizar a detecção de discurso de ódio em textos. A abordagem proposta nesta tese obteve o segundo melhor resultado para este corpus, perdendo apenas para Soto et al. (2022), que conseguiu 92% de medida F1, enquanto obteve-se 90% para a abordagem proposta.

Tabela 5.21 – Experimento 4: Comparação com o estado da arte.

Trabalho	Medida F1
Wiegand e Ruppenhofer (2021)	64%
Pagmunkas et al. (2021)	66%
Huang et al. (2020)	67%
Aluru et al. (2020)	69%
Silva e Roman (2020)	72%
Venturott e Ciarelli (2020)	74%
Metodologia proposta	90%
Soto et al.(2022)	**92%**

Fonte: Elaborado pelo autor, 2022.

Por fim, foram feitos testes de significância sobre os resultados da abordagem proposta e de trabalhos relacionados para o corpus HSD (2019). Comparou-se apenas o resultado da abordagem proposta com os primeiro e terceiro melhores resultados (VENTUROTT; CIARELLI, 2020; SOTO et al., 2022), visto que a diferença entre o desempenho da abordagem proposta para os demais trabalhos é maior (quase 20%). Os testes realizados

demonstraram que a abordagem proposta foi estocasticamente dominante sobre a de Venturott e Ciarelli (2020) - e, por consequência, as abordagens com resultados inferiores -, e conforme esperado, a abordagem proposta não foi estocasticamente dominante sobre a de Soto et al. (2022), obtendo *epsilon* mínimo de 1. (Tabela 5.22).

Tabela 5.22 – Resultados dos testes de hipóteses sobre o desempenho dos trabalhos que utilizaram o corpus HSD (experimento 4).

Hipótese	***Epsilon* mínimo**
H0-0: O desempenho da abordagem proposta não é superior ao de Venturott e Ciarelli (2019)	0
H0-1:O desempenho da abordagem proposta é superior ao de Venturott e Ciarelli (2019)	
H1-0: O desempenho da abordagem proposta não é superior ao de Soto et al. (2022)	1.0
H1-1: O desempenho da abordagem proposta é superior ao de Soto et al. (2022)	

5.1.5 Experimento 5: WH/OffComBr-2

O quinto experimento consistiu em utilizar o corpus WH - inglês - como idioma fonte I_f e o corpus OffComBr-2 como idioma alvo I_a. Para este experimento foram utilizados os MLPTs BERTimbau e XLM-R. Tendo em vista que esta pesquisa também teve como objetivo verificar o impacto do idioma alvo no resultado da detecção de discurso de ódio, este experimento se baseou em usar um idioma de tronco linguístico diferente do latino para ser o idioma fonte.

Na Tabela 5.23, pode-se ver os resultados da utilização da estratégia ZST para os MLPTs utilizados neste experimento. Além disso, na Tabela 5.24, podem ser vistos os resultados da utilização de JL.

Tabela 5.23 – Experimento 5: Resultados da utilização da estratégia ZST comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	72%	69%	**59%**
BERTimbau	59%	66%	55%

Fonte: Elaborado pelo autor, 2022.

Tabela 5.24 – Experimento 5: Resultados da utilização da estratégia JL comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	75%	76%	**74%**
BERTimbau	80%	68%	68%

Fonte: Elaborado pelo autor, 2022.

Os resultados dos experimentos CL estão listados na Tabela 5.25. Já os resultados dos experimentos CL/JL podem ser vistos na Tabela 5.26.

Tabela 5.25 – Experimento 5: Resultados da utilização da estratégia CL comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	79%	80%	80%
BERTimbau	82%	81%	**81%**

Tabela 5.26 – Experimento 5: Resultados da utilização da estratégia CL/JL comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	89%	89%	**89%**
BERTimbau	85%	85%	85%

No quinto experimento, a última estratégia testada foi a CL/JL+. Nestes experimentos, tem-se cinco ajustes finos adicionais nos MLPTs. Na Tabela 5.27, podem ser vistos os resultados destes experimentos.

Tabela 5.27 – Experimento 5: Resultados da utilização da estratégia CL/JL+ comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	92%	91%	91%
BERTimbau	92%	92%	**92%**

Ainda sobre o quinto experimento, foram realizados testes de hipóteses para averiguar se o desempenho dos MLPTs usando italiano foi estocasticamente superior ao uso do inglês (experimento 5). Foram comparados os resultados do BERTimbau utilizando inglês e italiano como idiomas fonte, e o português como idioma alvo. Os resultados demonstraram que não foi possível refutar a hipótese nula ($epsilon = 0.783$); assim não se pode afirmar que o desempenho do modelo ao usar italiano como idioma fonte é estocasticamente dominante sobre o do modelo ao usar inglês (Tabela 5.28).

Tabela 5.28 – Resultados dos testes de hipóteses sobre o desempenho do BERTimbau utilizando diferentes idiomas fonte (experimento 5).

Hipótese	*Epsilon* mínimo
H0-0: O desempenho do modelo utilizando italiano como idioma fonte não é superior ao da utilização do inglês como idioma fonte	0.783
H0-1:O O desempenho do modelo utilizando italiano como idioma fonte é superior ao da utilização do inglês como idioma fonte	

5.1.6 Experimento 6: WH/OffComBr-2-EN

No sexto experimento, foram utilizados o corpus WH como corpus auxiliar e o corpus OffComBr-2 traduzido para o inglês como corpus principal. Neste experimento, não é utilizada a ideia CLL propriamente dita. A ideia é não ter corpora de diferentes idiomas em um mesmo experimento, mas ter corpora auxiliares do mesmo idioma. Estes experimentos podem servir também como estudos de ablação, uma vez que CLL não é utilizado.

Neste experimento, fez-se necessário a utilização de uma máquina de tradução para traduzir o corpus OffComBr-2 para o inglês. Assim, os textos em português foram traduzidos para a língua inglesa usando a Máquina de Tradução Neural da Google (WU et al., 2016). Para este experimento, BERT e XLM-R foram utilizados como MLPTs, e utilizou-se todas as estratégias de treinamento.

Os resultados da estratégia ZST estão listados na Tabela 5.29 para os dois MLPTs utilizados. Ao utilizar a estratégia ZST, utilizou-se o corpus WH para realizar o treinamento, e o corpus OffComBr-2(EN) para a etapa de teste. Já os resultados da estratégia JL podem ser vistos na Tabela 5.30. Na utilização desta estratégia, uma parte do corpus OffComBr-2(EN) é adicionado ao treinamento dos MLPTs.

Tabela 5.29 – Experimento 6: Resultados da utilização da estratégia ZST comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	46%	68%	**55%**
BERT	55%	68%	**55%**

Tabela 5.30 – Experimento 6: Resultados da utilização da estratégia JL comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	74%	74%	74%
BERT	80%	75%	**78%**

Na Tabela 5.31 estão listados os resultados da utilização de CL. Apenas o corpus WH foi utilizado no treinamento dos MLPTs. No ajuste fino, somente o corpus OffComBr-2 (EN) foi usado. A Tabela 5.32 apresenta os resultados da utilização de CL/JL, em que uma parte dos dados do corpus OffComBr-2 (EN) é usado no treinamento. Por outro lado, no ajuste fino o restante deste corpus é utilizado para treinamento, validação e teste.

Tabela 5.31 – Experimento 6: Resultados da utilização da estratégia CL comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	84%	85%	84%
BERT	85%	85%	**85%**

Tabela 5.32 – Experimento 6: Resultados da utilização da estratégia CL/JL comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	85%	86%	85%
BERT	86%	86%	**86%**

Na Tabela 5.33 são apresentados os resultados da estratégia CL/JL+. Nesta estratégia, uma parte dos dados do corpus OffComBr-2 (EN) é anexada ao treinamento do primeiro ajuste fino dos MLPTs, assim como nos experimentos CL/JL. Nos ajustes finos, o restante do corpus OffComBr-2 (EN) é dividido de acordo com o número de iterações.

Tabela 5.33 – Experimento 6: Resultados da utilização da estratégia CL/JL+ comparando os MLPTs usados.

MLPT	Precisão	Revocação	Medida F1
XLM-R	91%	90%	**90%**
BERT	91%	90%	**90%**

Também foram realizados testes de hipóteses neste experimento, para averiguar se o desempenho dos MLPTs usando o corpus de Pelle e Moreira (2017) foi estocasticamente superior ao uso de uma versão deste corpus traduzida para o inglês. Foram comparados os resultados de MLPTs diferentes: para o caso do corpus original, utilizou-se os resultados do BERTimbau, e no caso do corpus traduzido, utilizou-se os resultados do BERT original. Os resultados demonstraram que pode-se refutar a hipótese nula ($epsilon = 0.01$); seguindo-se então com a hipótese alternativa, de que o desempenho do modelo ao usar o corpus em português foi estocasticamente dominante sobre a utilização da versão traduzida do corpus (Tabela 5.34).

Tabela 5.34 – Resultados dos testes de hipóteses sobre o desempenho dos MLPTs sobre o uso dos corpora OffComBr-2 original e traduzido para o inglês (experimento 6).

Hipótese	*Epsilon* mínimo
H0-0: O desempenho do modelo utilizando o corpus original não é superior ao da utilização da versão traduzida para inglês	0.01
H0-1:O O desempenho do modelo utilizando o corpus original é superior ao da utilização da versão traduzida para inglês	

5.2 DISCUSSÃO DOS RESULTADOS

Como pode-se ver na Tabela 5.1, em todos os casos, os resultados dos modelos de tamanho grande são melhores do que os de tamanho base. Os testes de significância realizados reforçam isto (Tabela 5.2). No primeiro experimento, foram utilizados quatro MLPTs: BERTimbau (SILVA, 2020), BERT italiano (SCHWETER, 2020), BERT (DEVLIN et al., 2019) e XLM-R (CONNEAU et al., 2020). O XLM-R foi utilizado em todos os experimentos por ter sido treinado com um grande corpus multilíngue. O BERTimbau foi utilizado já que o idioma alvo nestes experimentos é o português. Já a utilização dos MLPTs BERT e BERT italiano teve apenas caráter puramente investigativo. Usou-se o BERT italiano já que os dados do idioma fonte eram em italiano. Já o BERT (inglês) foi utilizado devido a este modelo ter sido o primeiro MLPT a ser disponibilizado, tendo alcançado os melhores resultados em diversas tarefas de Processamento de Linguagem Natural.

Para todos os casos no experimento 1, os MLPTs BERT italiano e BERT obtiveram um desempenho ruim, quando comparado ao desempenho do XLM-R e BERTimbau. Isso deve-se ao fato de que o português foi o idioma de mais enfoque no primeiro experimento. Em todos os casos, o corpus em português era usado para avaliar o modelo treinado; e em alguns experimentos, este corpus também era utilizado na etapa de treinamento. Tendo em vista que os dois MLPTs citados não foram treinados inicialmente com dados em português, eles apresentaram um resultado inferior em relação ao XLM-R e BERTimbau. O BERT italiano ainda apresentou um resultado levemente superior ao BERT. Possivelmente, isto deve-se ao fato de o idioma italiano ser do mesmo tronco linguístico do idioma português.

Na utilização da estratégia ZST do experimento 1, foi visto que o MLPT XLM-R apresentou melhor desempenho (Tabela 5.3). Esse comportamento repetiu-se em todas os demais experimentos. Uma possível explicação para este fato é que o XLM-R foi treinado inicialmente com corpora multilíngues. Assim, ele consegue apresentar um bom desempenho em tarefas que envolvem CLL.

Ainda no experimento 1, na Tabela 5.4 (experimentos JL) vê-se que os resultados foram superiores aos da estratégia ZST. Vê-se então que, ao utilizar dados do idioma alvo I_a no treinamento do primeiro ajuste fino dos MLPTs traz benefícios na performance dos modelos usados. A partir da Tabela 5.5 (experimentos CL), observa-se que o BERTimbau começa a apresentar um resultado superior ao XLM-R. Nos experimentos CL, o ajuste fino dos MLPTs é feito unicamente com o corpus OffComBr-2. Nesta situação, o BERTimbau leva vantagem por ter sido treinado apenas com dados neste idioma.

Na Tabela 5.6, pode-se ver os resultados dos experimentos CL/JL. Novamente, é possível observar que a adição de dados do idioma I_a ao treinamento do primeiro ajuste fino do modelo melhora o desempenho dos MLPTs.

Sobre os experimentos CL/JL+, tem-se a Tabela 5.7 mostrando os resultados obtidos por número de ajustes finos adicionais no MLPT utilizado - neste caso, apenas o XLM-R. É possível ver que o melhor resultado obtido entre todos os experimentos foi quando foram realizados cinco ajustes finos adicionais.

Pelle e Moreira (2017) forneceram um *baseline* para o corpus OffComBr-2 e utilizaram SVM para realizar a classificação e obter seu melhor resultado. Pari et al. (2019) e Silva et al. (2019) utilizaram abordagens com aprendizado profundo. Ambos usaram redes neurais convolucionais; Pari et al. (2019) utilizaram *word2vec* com 100 dimensões em conjunto com CNN, enquanto Silva et al. (2019) utilizaram *wang2vec* com 100 dimensões, obtendo o melhor resultado no estado da arte até então. Vale a pena ressaltar que a metodologia proposta nesta tese é a única que utilizou CLL, alcançando o melhor resultado dentre os trabalhos que utilizaram o corpus OffComBr-2, conforme pode ser visto na Tabela 5.9. Os testes de significância realizados reforçam este resultado (Tabela 5.10).

Sobre os estudos de ablação realizados (experimento 2), observou-se que a utilização de CLL melhorou substancialmente a performance dos modelos utilizados (Tabela 5.11), e os testes de significância realizados dão suporte a esse resultado (Tabela 5.12). Ao realizar detecção de discurso de ódio utilizando apenas o corpus OffComBr-2, o resultado foi muito insatisfatório, tanto para o BERTimbau, quanto para o XLM-R. Até mesmo o resultado mais baixo obtido nos experimentos utilizando CLL - 66% com o BERTimbau e usando a estratégia ZST - conseguiu-se um resultado superior ao da não utilização de CLL. Adicionalmente, é possível visualizar tais discrepâncias nas Figuras 5.1 e 5.2 .

O terceiro experimento foi sobre o balanceamento dos dados utilizados. O corpus OffComBr-2 apresenta dados desbalanceados; então foram realizados experimentos tanto para deixá-los na mesma proporção que o corpus Evalita (Facebook), quanto para deixar os dados do corpus Evalita na mesma proporção que o corpus OffComBr-2. Os resultados

mostraram que o balanceamento não forneceu um resultado superior; pelo contrário, em alguns casos houve um decréscimo nas métricas de avaliação ao se realizar balanceamento nos corpora utilizados. Houve uma variação entre 0-3% de diferença entre os resultados do balanceamento e da proporção original dos corpora.

A diminuição da performance dos modelos pode ser explicada devido ao fato de se ter reduzido o conjunto de treinamento nos experimentos, tendo em vista que foi utilizada a técnica subamostragem, que consiste em remover instâncias da classe majoritária (Tabela 5.16). Os resultados dos testes de significância para o MLPT XLMR-R mostraram que o desempenho do modelo foi melhor sem a realização de balanceamento nos dados (Tabela 5.14). Já para o BERTimbau, na estratégia CL não houve diferença significativa entre os resultados com os balanceamentos realizados (Tabela 5.15).

No quarto experimento, um novo corpus em português foi utilizado: HSD (FORTUNA et al., 2019). Este corpus continha mais dados do que o corpus Evalita (Facebook), usado nos experimentos anteriores. Então, fez-se necessário obter mais dados para este corpus para que a premissa de que o idioma fonte precisa ter mais recursos (ou dados) fosse mantida. Assim, o corpus Evalita utilizada neste experimento consistiu de 4.000 postagens no Facebook + 4.000 postagens no Twitter. A partir deste de experimento, utilizaram-se apenas os MLPT XLM-R e BERTimbau, tendo em vista que a performance dos modelos BERT e BERT italiano não se mostraram eficazes. Assim como no primeiro experimento, o BERTimbau obteve o melhor resultado em comparação com o XLM-R, e o melhor resultado obtido foi utilizando a estratégia CL/JL+.

Como pôde ser visto na Tabela 5.21, a metodologia proposta apresentou resultados promissores para o corpus HSD, tendo sido inferior apenas ao resultado de Soto et al. (2022). Os testes de significância reforçaram este resultado (Tabela 5.22). Soto et al. (2022) alcançaram 92% de medida F1 utilizando a representação *Glove* com 300 dimensões, com os *embeddings* NILC (HARTMANN et al., 2017), e uma rede CNN para classificar os textos.

O quinto experimento baseou-se na investigação de utilizar um idioma fonte cuja base não era a latina (neste caso, o inglês) para auxiliar na detecção de discurso de ódio em português. Os resultados dos experimentos mostraram que a utilização da língua inglesa contribuiu positivamente na performance dos MLPTs utilizados. Mais uma vez, obteve-se o resultado de 92% de medida F1 para o corpus OffComBr-2. Em relação aos MLPTs utilizados (BERTimbau e XLM-R), observou-se que os resultados de ambos foram próximos em quase todos os experimentos. Nos experimentos CL/JL+, a diferença foi de apenas 1% (Tabela 5.27).

Foram realizados testes de significância para averiguar se houve diferença significativa entre os resultados de utilizar italiano ou inglês como idioma fonte na abordagem proposta. Os resultados mostraram que não houve diferença significativa; assim, ambos os idiomas contribuem de forma positiva como idioma fonte, utilizando o português como idioma alvo (Tabela 5.28).

No sexto experimento, o corpus de WH foi utilizado como corpus auxiliar, e o corpus OffComBr-2 traduzido para o inglês foi usado como corpus principal. Os MLPTs utilizados foram XLM-R e BERT. Os resultados dos MLPTs foram bem semelhantes um ao outro, e o melhor resultado obtido aqui (90% com a abordagem CL/JL+ - Tabela 5.2) não superou o já obtido anteriormente - utilizando italiano como idioma fonte. Os resultados dos testes de significância reafirmam o resultado obtido (Tabela 5.34).

Em relação ao número de épocas adotado nos experimentos, utilizou-se o número recomendado por Devlin et al. (2019) - entre 2 e 4 épocas. Porém, como em algumas estratégias, são feitos vários ajustes finos a soma total de épocas utilizadas chega até 15. Desse modo, foram realizados experimentos com todas as estratégias experimentais aumentando o número de épocas. Porém, foi visto que o aumento no desempenho dos modelos não era significativo, e o custo computacional para a utilização de um número maior de épocas se tornou relativamente alto. Para experimentos utilizando a estratégia JL, por exemplo, o treinamento durou mais de 1 hora com 15 épocas. Deste modo, optou-se por continuar a utilizar poucas épocas em cada treinamento.

No tocante ao custo computacional no desenvolvimento da metodologia proposta, foi utilizado o Google Colab com uma GPU Nvidia Tesla K80 nos experimentos realizados. Na maioria das abordagens experimentais, foram utilizados cerca de 4 GB de memória RAM e 40 GB de armazenamento em disco. O custo computacional torna-se diretamente proporcional ao número de ajustes finos realizados. Para os experimentos CL/JL+, quase todos os recursos disponíveis no Google Colab foram utilizados, tanto em termos de memória RAM, quanto em termos de armazenamento em disco.

Como apontado por Liu et al. (2019) e Branco et al. (2016), um dos problemas que ocorre com soluções que utilizam aprendizado profundo é o super ajustamento do modelo aos dados (*overfitting*). Uma das formas de se observar isto é verificando o comportamento da função de perda ao longo das épocas. Nos experimentos CL/JL+, notou-se que havia uma tendência ao *overfitting* do modelo com um número de ajustes finos posteriores maior que cinco.

A Figura 5.3 exibe o comportamento das funções de perda de treino e validação acumuladas, utilizando a estratégia CL/JL+. Como mencionado, adotou-se o número

de épocas como sendo igual a 3 em cada treinamento realizado. Tendo em vista que na estratégia CL/JL+ são feitos cinco ajustes finos, tem-se então 15 épocas no total. Na Figura 5.3, vê-se que tanto a função de perda no treino quanto na validação apresentam um comportamento semelhante. Deste modo, observa-se que não há uma tendência ao *overfitting* nos modelos utilizados.

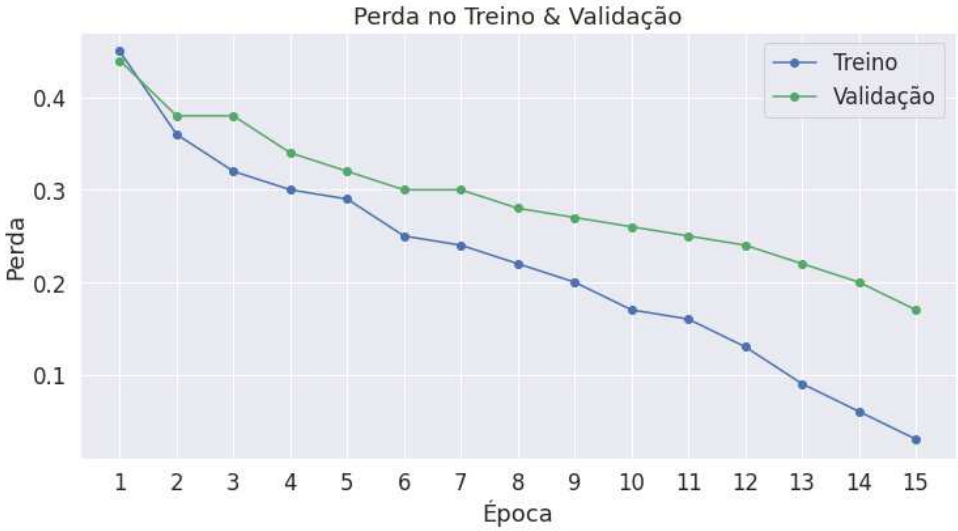

Figura 5.3 – Gráfico das funções de perda por número de épocas.

Fonte: Elaborado pelo autor, 2022.

5.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo, foram detalhados todos os experimentos realizados utilizando a metodologia proposta para detecção de discurso de ódio utilizando CLL. Além disso, foram apresentados os testes de significância realizados e os resultados dos experimentos foram discutidos.

Foram experimentados diferentes idiomas como fonte na abordagem proposta - inglês e italiano. Ambos os idiomas apresentaram bons resultados com a utilização de português como idioma alvo. Além disso, os resultados demonstraram que o MLPT BERTimbau obteve o melhor desempenho geral para os experimentos utilizando o português como idioma alvo. Com os resultados obtidos, foi possível responder às questões de pesquisa elencadas no Capítulo 1, as quais serão melhor discutidas no próximo capítulo.

No capítulo seguinte, serão vistas as conclusões obtidas neste tese, bem como as vias de pesquisa a serem seguidas futuramente.

6 CONSIDERAÇÕES FINAIS

Este capítulo apresenta as conclusões extraídas a partir desta tese, além das vias de pesquisa a serem seguidas como continuação deste trabalho.

Esta pesquisa endereçou o problema de detecção de discurso de ódio em textos. Mais especificamente, foi abordada a dificuldade em desenvolver soluções que realizem detecção de discurso de ódio em idiomas com poucos dados disponíveis. Para este fim, a abordagem proposta baseia-se no uso de CLL para realizar detecção de discurso de ódio em português. Foram utilizados corpora em italiano e inglês como idiomas fonte, e corpora em português como idioma alvo.

Um dos objetivos dessa tese foi o desenvolvimento de uma metodologia para a realização de detecção de discurso de ódio em português utilizando CLL. Os resultados obtidos mostraram que o uso de CLL com idiomas de base latina como idiomas fonte é promissor quando o idioma alvo é o português. O desempenho dos MLPTs utilizados na tese foi melhorado com o uso do italiano como idioma fonte. O mesmo se aplica a idiomas de base anglo-saxônica, uma vez que a utilização do inglês como idioma fonte também melhorou o desempenho dos MLPTs utilizados. Os testes de significância realizados em ambos os casos dão suporte a estas conclusões. É importante pontuar também que a metodologia apresentada nesta tese contribuiu com o estado-da-arte no tocante à detecção de discurso de ódio em português.

Dos MLPTs utilizados, observou-se que ao utilizar o português como idioma alvo, o BERT e o BERT italiano não obtiveram um bom desempenho na metodologia proposta. Assim, o BERTimbau e o XLM-R tiveram um melhor desempenho do que aqueles; e o BERTimbau demonstrou um melhor desempenho em todos os cenários experimentados, com o suporte dos testes de significância realizados.

Das estratégias de treinamento utilizadas, pode-se observar que a estratégia CL/JL+ obteve o melhor desempenho em todos os cenários. Além disso, os resultados dos estudos de ablação mostraram que o desempenho dos MLPTs utilizados foi melhorado com o uso de CLL. Isso também se reflete em relação ao estado da arte; uma vez que foi obtido o melhor desempenho para o corpus OffComBr-2 e o segundo melhor desempenho para o corpus HSD. Os testes de significância realizados reforçam o desempenho da metodologia para ambos os corpora.

Considerando as questões de pesquisa expostas no Capítulo 1:

Q1: O uso de CLL melhora o desempenho do modelo proposto?

Para responder à esta questão, foram realizados estudos de ablação (experimento 2). Os resultados demonstraram que o uso de CLL melhorou significativamente o desempenho dos MLPTs utilizados na metodologia proposta. Os resultados podem ser visualizados na Tabela 5.11.

Q2: O uso de CLL traz resultados relevantes em relação ao estado da arte?

Além do fato do uso de CLL melhorar o desempenho da abordagem proposta, outra preocupação foi concernente aos resultados encontrados no estado da arte. Para responder à questão de pesquisa **Q2**, buscaram-se resultados de outros trabalhos que utilizaram os corpora OffComBr-2 e HSD. Assim, foi possível realizar uma comparação com os demais trabalhos no estado da arte, e foi possível verificar que o uso de CLL trouxe resultados relevantes. A metodologia proposta obteve o melhor desempenho dentre os trabalhos que utilizaram o corpus OffComBr-2, e o segundo melhor desempenho dentre os trabalhos que utilizaram o corpus HSD. É importante ressaltar que, neste trabalho, foi utilizada a métrica F1 ponderada, enquanto que na maioria dos trabalhos relacionados não foi mencionado o tipo de média computada para a medida F1.

Q3: O uso de CLL com um idioma de base latina como idioma fonte melhora o desempenho do modelo proposto?

Uma das contribuições desta tese consistiu da investigação de quais idiomas seriam úteis quando utilizados como idiomas fonte na metodologia proposta, ao utiliza-se o português como idioma alvo. Para responder à questão de pesquisa **Q3**, utilizou-se o italiano como idioma fonte nos experimentos 1 e 2. No experimento 1, utilizando-se a estratégia experimental CL/JL+, obteve-se uma medida F1 de 92% para o MLPT BERTimbau (Tabela 5.8), utilizando-se o português como idioma alvo (corpus OffComBr-2). Os resultados do experimento 2 também demonstraram a contribuição do uso do italiano como idioma fonte (Tabela 5.11).

Q4: O uso de CLL com um idioma de base anglo-saxônica como idioma fonte melhora o desempenho do modelo proposto?

Para responder à esta questão, foi realizado o experimento 5. Neste experimento, foi utilizado o inglês como idioma fonte, dada a utilização do português como idioma alvo. Os resultados deste experimento demonstraram que o desempenho da metodologia proposta foi melhorada ao utilizar-se o inglês como idioma fonte. Com o MLPT BERTimbau e com o uso da estratégia experimental CL/JL+, obteve-se uma medida F1 de 92% - tendo

o mesmo resultado da utilização do italiano como idioma fonte, e alcançando o melhor resultado dentre os trabalhos que utilizaram o corpus OffComBr-2.

6.1 LIMITAÇÕES

Nesta seção, serão abordadas algumas das limitações encontradas na pesquisa realizada.

Como descrito no Capítulo 4.1, os corpora utilizados nesta pesquisa têm focos diferentes. Os dois corpora principais utilizados foram OffComBr-2 e Evalita (Facebook). Um deles contém textos do Facebook, enquanto o outro é composto por comentários extraídos de páginas de notícias.

Existem alguns termos que podem ser considerados ofensivos em um idioma, mas em outro não. Além disso, existem termos que estão atrelados a um contexto geopolítico. É o que acontece no tocante a textos relacionados a partidos políticos. Um exemplo disto está numa postagem contida no corpus italiano, a qual foi rotulada pelos autores como discurso de ódio (Quadro 6.1). Os alvos da mensagem ofensiva contidos neste texto (*pd* e *ncd*) não auxiliam na tarefa de detecção de discurso de ódio em português, visto que tratam-se de partidos políticos italianos que não existem no Brasil.

Quadro 6.1 – Mensagem extraída do corpus Evalita 2018.

Mensagem original	Tradução	Discurso de ódio
Ho cambiato canale...le 2 del pd e ncd mi fanno schifo	Eu mudei de canal ... os 2 do pd e ncd me deixam doente	Sim

Fonte: Elaborado pelo autor, 2022.

Um outro questionamento neste mesmo exemplo é que a classificação do discurso de ódio pode ser subjetiva. Assim, para um dado anotador, essa postagem pode não ser considerada como discurso de ódio, visto que não contempla nenhuma ofensa direta a um grupo específico de pessoas. Já para os anotadores do corpus, a mensagem continha discurso de ódio.

Como mencionado por Bender et al. (2021), modelos de linguagem têm suas limitações. Além do custo computacional e ambiental para a criação destes modelos, deve-se levar em consideração que existem erros inerentes aos mesmos. Bender et al. (2021) relatam os vieses presentes nos corpora utilizados por MLPTs como BERT e GPT-3. Assim, estas limitações também se aplicam a este trabalho, tendo em vista que MLPTs são utilizados na metodologia proposta.

6.2 TRABALHOS FUTUROS

Os trabalhos futuros a serem realizados como continuação desta pesquisa são:

- Utilização de outros idiomas como idioma fonte na metodologia proposta. Podem ser utilizados idiomas de base latina como espanhol ou galego, por exemplo, para afirmar a utilização destes idiomas como fonte. Além disso, podem ser utilizados idiomas de outras famílias linguísticas - como árabe ou chinês -, para verificar o comportamento da metodologia proposta nestes casos;
- Abordagem de outros temas semelhantes, como detecção de ofensas, por exemplo. Deste modo, a metodologia proposta poderia ser validada em outros domínios;
- Utilização de outros MLPTs na metodologia proposta, como ALBERT, DistilBERT, Electra, dentre outros;
- Realizar um estudo com a metodologia proposta em uma aplicação no mundo real;
- Exploração de outros *word embeddings* para verificar o impacto do uso destes na metodologia proposta (como os do NILC, por exemplo). Deste modo, não seriam utilizados os *word embeddings* dos próprios MLPTs;
- Utilização de mais de um corpus como idioma fonte e/ou mais de um corpus como idioma alvo. Deste modo, poderia-se identificar se a utilização de mais de um corpus de mesmo idioma, ou de mais de um corpus de idiomas distintos poderia trazer benefícios à metodologia proposta;
- Realizar experimentos utilizando *bootstraping* para verificar impacto do balanceamento de dados em diferentes corpora;
- Verificar a utilização da metodologia desenvolvida para outros contextos, como detecção de *fake news*, por exemplo.

REFERÊNCIAS

ALAMMAR, J. The illustrated transformer. **The Illustrated Transformer–Jay Alammar–Visualizing Machine Learning One Concept at a Time**, v. 27, 2018.

ALURU, S. S.; MATHEW, B.; SAHA, P.; MUKHERJEE, A. Deep learning models for multilingual hate speech detection. **arXiv preprint arXiv:2004.06465**, 2020.

ANAGNOSTOU, A.; MOLLAS, I.; TSOUMAKAS, G. Hatebusters: A web application for actively reporting youtube hate speech. In: **IJCAI**. [S.l.: s.n.], 2018. p. 5796–5798.

ARCO, F. M. P. del; MOLINA-GONZáLEZ, M. D.; UREñA-LóPEZ, L. A.; MARTíN-VALDIVIA, M. T. Comparing pre-trained language models for spanish hate speech detection. **Expert Systems with Applications**, v. 166, p. 114120, 2021. ISSN 0957-4174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S095741742030868X>.

BADJATIYA, P.; GUPTA, M.; VARMA, V. Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In: **The World Wide Web Conference**. [S.l.: s.n.], 2019. p. 49–59.

BADJATIYA, P.; GUPTA, S.; GUPTA, M.; VARMA, V. Deep learning for hate speech detection in tweets. In: **Proceedings of the 26th international conference on World Wide Web companion**. [S.l.: s.n.], 2017. p. 759–760.

BARRIO, E. D.; CUESTA-ALBERTOS, J. A.; MATRÁN, C. An optimal transportation approach for assessing almost stochastic order. In: **The Mathematics of the Uncertain**. [S.l.]: Springer, 2018. p. 33–44.

BASILE, V.; BOSCO, C.; FERSINI, E.; DEBORA, N.; PATTI, V.; PARDO, F. M. R.; ROSSO, P.; SANGUINETTI, M. et al. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **13th International Workshop on Semantic Evaluation**. [S.l.], 2019. p. 54–63.

BASSIGNANA, E.; BASILE, V.; PATTI, V. Hurtlex: A multilingual lexicon of words to hurt. In: CEUR-WS. **5th Italian Conference on Computational Linguistics, CLiC-it 2018**. [S.l.], 2018. v. 2253, p. 1–6.

BENDER, E. M.; GEBRU, T.; MCMILLAN-MAJOR, A.; SHMITCHELL, S. On the dangers of stochastic parrots: Can language models be too big? In: **Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency**. [S.l.: s.n.], 2021. p. 610–623.

BHASKARAN, J.; BHALLAMUDI, I. Good secretaries, bad truck drivers? occupational gender stereotypes in sentiment analysis. p. 62–68, 2019.

BIGOULAEVA, I.; HANGYA, V.; FRASER, A. Cross-lingual transfer learning for hate speech detection. In: **Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion**. [S.l.: s.n.], 2021. p. 15–25.

BLOMMAERT, J. **Grassroots literacy: Writing, identity and voice in Central Africa**. [S.l.]: Routledge, 2008.

BOSCO, C.; FELICE, D.; POLETTO, F.; SANGUINETTI, M.; MAURIZIO, T. Overview of the evalita 2018 hate speech detection task. In: CEUR. **EVALITA 2018-Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian**. [S.l.], 2018. v. 2263, p. 1–9.

BOURGONJE, P.; MORENO-SCHNEIDER, J.; SRIVASTAVA, A.; REHM, G. Automatic classification of abusive language and personal attacks in various forms of online communication. In: SPRINGER, CHAM. **International Conference of the German Society for Computational Linguistics and Language Technology**. [S.l.], 2017. p. 180–191.

BRANCO, P.; TORGO, L.; RIBEIRO, R. P. A survey of predictive modeling on imbalanced domains. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 49, n. 2, p. 1–50, 2016.

BURNAP, P.; WILLIAMS, M. L. Us and them: identifying cyber hate on twitter across multiple protected characteristics. **EPJ Data science**, Springer, v. 5, p. 1–15, 2016.

CHOWDHURY, A. G.; DIDOLKAR, A.; SAWHNEY, R.; SHAH, R. Arhnet-leveraging community interaction for detection of religious hate speech in arabic. In: **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop**. [S.l.: s.n.], 2019. p. 273–280.

CHOWDHURY, G. G. Natural language processing. **Annual review of information science and technology**, Wiley Online Library, v. 37, n. 1, p. 51–89, 2003.

CHUNG, Y.-L.; KUZMENKO, E.; TEKIROĞLU, S. S.; GUERINI, M. Conan-counter narratives through nichesourcing: a multilingual dataset of responses to fight online hate speech. p. 2819–2829, 2019.

CICHY, R. M.; KAISER, D. Deep neural networks as scientific models. **Trends in cognitive sciences**, Elsevier, v. 23, n. 4, p. 305–317, 2019.

CLARK, K.; LUONG, M.-T.; LE, Q. V.; MANNING, C. D. Electra: Pre-training text encoders as discriminators rather than generators. 2019.

CONNEAU, A.; KHANDELWAL, K.; GOYAL, N.; CHAUDHARY, V.; WENZEK, G.; GUZMÁN, F.; GRAVE, É.; OTT, M.; ZETTLEMOYER, L.; STOYANOV, V. Unsupervised cross-lingual representation learning at scale. p. 8440–8451, 2020.

CORAZZA, M.; MENINI, S.; CABRIO, E.; TONELLI, S.; VILLATA, S. A multilingual evaluation for online hate speech detection. **ACM Transactions on Internet Technology (TOIT)**, ACM New York, NY, USA, v. 20, n. 2, p. 1–22, 2020.

COUNCIL, U. H. R. 2013. (https://www.refworld.org/docid/50f925cf2.html).

DAVIDSON, T.; WARMSLEY, D.; MACY, M.; WEBER, I. Automated hate speech detection and the problem of offensive language. In: **Proceedings of the International AAAI Conference on Web and Social Media**. [S.l.: s.n.], 2017. v. 11, n. 1.

DEAN, J.; GHEMAWAT, S. Mapreduce: simplified data processing on large clusters. **Communications of the ACM**, ACM New York, NY, USA, v. 51, n. 1, p. 107–113, 2008.

DENG, L.; LIU, Y. **Deep learning in natural language processing**. [S.l.]: Springer, 2018.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 4171–4186. Disponível em: <https://aclanthology.org/N19-1423>.

DJURIC, N.; ZHOU, J.; MORRIS, R.; GRBOVIC, M.; RADOSAVLJEVIC, V.; BHAMIDIPATI, N. Hate speech detection with comment embeddings. In: **Proceedings of the 24th international conference on world wide web**. [S.l.: s.n.], 2015. p. 29–30.

DONG, L.; XU, S.; XU, B. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In: IEEE. **2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2018. p. 5884–5888.

DROR, R.; PELED-COHEN, L.; SHLOMOV, S.; REICHART, R. Statistical significance testing for natural language processing. **Synthesis Lectures on Human Language Technologies**, Morgan & Claypool Publishers, v. 13, n. 2, p. 1–116, 2020.

DROR, R.; SHLOMOV, S.; REICHART, R. Deep dominance-how to properly compare deep neural models. In: **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**. [S.l.: s.n.], 2019. p. 2773–2785.

EISENSTEIN, J. **Natural language processing**. [S.l.]: MIT press, 2018.

EL-KASSAS, W. S.; SALAMA, C. R.; RAFEA, A. A.; MOHAMED, H. K. Automatic text summarization: A comprehensive survey. **Expert Systems with Applications**, Elsevier, v. 165, p. 113679, 2021.

ENGELEN, J. E. V.; HOOS, H. H. A survey on semi-supervised learning. **Machine Learning**, Springer, v. 109, n. 2, p. 373–440, 2020.

FIRMINO, A. A.; BAPTISTA, C. S. de; PAIVA, A. C. de. Using cross lingual learning for detecting hate speech in portuguese. In: STRAUSS, C.; KOTSIS, G.; TJOA, A. M.; KHALIL, I. (Ed.). **Database and Expert Systems Applications**. Cham: Springer International Publishing, 2021. p. 170–175. ISBN 978-3-030-86475-0.

FORTUNA, P.; NUNES, S. A survey on automatic detection of hate speech in text. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 51, n. 4, p. 1–30, 2018.

FORTUNA, P.; SILVA, J. R. da; WANNER, L.; NUNES, S. et al. A hierarchically-labeled portuguese hate speech dataset. In: **Proceedings of the Third Workshop on Abusive Language Online**. [S.l.: s.n.], 2019. p. 94–104.

FORTUNA, P.; SOLER-COMPANY, J.; WANNER, L. How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? **Information Processing & Management**, Elsevier, v. 58, n. 3, p. 102524, 2021.

FORTUNA, P. C. T. Automatic detection of hate speech in text: an overview of the topic and dataset annotation with hierarchical classes. 2017.

FRENDA, S.; GHANEM, B.; GÓMEZ, M. Montes-y; ROSSO, P. Online hate speech against women: Automatic identification of misogyny and sexism on twitter. **Journal of Intelligent & Fuzzy Systems**, IOS Press, v. 36, n. 5, p. 4743–4752, 2019.

GOOGLE. **Embeddings: Translating to a Lower-Dimensional Space**. 2022. (https://developers.google.com/machine-learning/crash-course/embeddings/translating-to-a-lower-dimensional-space). [Online; acesso em 21 mar. 2022].

GORDON, M.; KOCHEN, M. Recall-precision trade-off: A derivation. **Journal of the American Society for Information Science**, Wiley Online Library, v. 40, n. 3, p. 145–151, 1989.

GOUTTE, C.; GAUSSIER, E. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In: **Proceedings of the 27th European Conference on Advances in Information Retrieval Research**. Berlin, Heidelberg: Springer-Verlag, 2005. (ECIR'05), p. 345–359. ISBN 3540252959. Disponível em: <https://doi.org/10.1007/978-3-540-31865-1_25>.

HARTMANN, N.; FONSECA, E.; SHULBY, C.; TREVISO, M.; SILVA, J.; ALUÍSIO, S. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. p. 122–131, 2017.

HEIJDEN, N. van der; YANNAKOUDAKIS, H.; MISHRA, P.; SHUTOVA, E. Multilingual and cross-lingual document classification: A meta-learning approach. p. 1966–1976, 2021.

HEWITT, S.; TIROPANIS, T.; BOKHOVE, C. The problem of identifying misogynist language on twitter (and other online social spaces). In: **Proceedings of the 8th ACM Conference on Web Science**. [S.l.: s.n.], 2016. p. 333–335.

HU, J.; RUDER, S.; SIDDHANT, A.; NEUBIG, G.; FIRAT, O.; JOHNSON, M. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2020. p. 4411–4421.

HUANG, X.; XING, L.; DERNONCOURT, F.; PAUL, M. Multilingual twitter corpus and baselines for evaluating demographic bias in hate speech recognition. p. 1440–1448, 2020.

IBA, H.; NOMAN, N. **Deep Neural Evolution**. [S.l.]: Springer, 2020.

JIANG, T.; GRADUS, J. L.; ROSELLINI, A. J. Supervised machine learning: A brief primer. **Behavior Therapy**, v. 51, n. 5, p. 675–687, 2020. ISSN 0005-7894. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0005789420300678>.

JOULIN, A.; GRAVE, É.; BOJANOWSKI, P.; MIKOLOV, T. Bag of tricks for efficient text classification. In: **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers**. [S.l.: s.n.], 2017. p. 427–431.

KARIM, M. R.; DEY, S. K.; ISLAM, T.; SARKER, S.; MENON, M. H.; HOSSAIN, K.; HOSSAIN, M. A.; DECKER, S. Deephateexplainer: Explainable hate speech detection in under-resourced bengali language. In: IEEE. **2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)**. [S.l.], 2021. p. 1–10.

KEMP, S. 2021. (https://datareportal.com/reports/digital-2021-global-overview-report).

KOTTASOVá, I. 2017. (https://money.cnn.com/2017/06/01/technology/twitter-facebook-hate-speech-europe/index.html).

KUIPERS, G.; ENT, B. Van der. The seriousness of ethnic jokes: Ethnic humor and social change in the netherlands, 1995–2012. **Humor**, De Gruyter, v. 29, n. 4, p. 605–633, 2016.

LAMPLE, G.; CONNEAU, A. Cross-lingual language model pretraining. **arXiv preprint arXiv:1901.07291**, 2019.

LAMPLE, G.; CONNEAU, A.; RANZATO, M.; DENOYER, L.; JÉGOU, H. Word translation without parallel data. 2018.

LAN, Z.; CHEN, M.; GOODMAN, S.; GIMPEL, K.; SHARMA, P.; SORICUT, R. Albert: A lite bert for self-supervised learning of language representations. 2019.

LEE, S.; MIN, S.-J.; EIGENMANN, R. Openmp to gpgpu: a compiler framework for automatic translation and optimization. **ACM Sigplan Notices**, ACM New York, NY, USA, v. 44, n. 4, p. 101–110, 2009.

LEWIS, W. Haitian creole: How to build and ship an mt engine from scratch in 4 days, 17 hours, & 30 minutes. In: CITESEER. **EAMT**. [S.l.], 2010.

LIMA, C.; BIANCO, G. D. Extração de característica para identificação de discurso de ódio em documentos. In: SBC. **Anais da XV Escola Regional de Banco de Dados**. [S.l.], 2019. p. 61–70.

LING, W.; DYER, C.; BLACK, A. W.; TRANCOSO, I. Two/too simple adaptations of word2vec for syntax problems. In: **Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. [S.l.: s.n.], 2015. p. 1299–1304.

LITSCHKO, R.; GLAVAŠ, G.; PONZETTO, S. P.; VULIĆ, I. Unsupervised cross-lingual information retrieval using monolingual data only. In: **The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval**. [S.l.: s.n.], 2018. p. 1253–1256.

LIU, Y.; LAPATA, M. Text summarization with pretrained encoders. p. 3730–3740, 2019.

LIU, Y.; OTT, M.; GOYAL, N.; DU, J.; JOSHI, M.; CHEN, D.; LEVY, O.; LEWIS, M.; ZETTLEMOYER, L.; STOYANOV, V. Roberta: A robustly optimized bert pretraining approach. **arXiv preprint arXiv:1907.11692**, 2019.

LUQMAN, H.; MAHMOUD, S. A. Automatic translation of arabic text-to-arabic sign language. **Universal Access in the Information Society**, Springer, v. 18, n. 4, p. 939–951, 2019.

MATHEW, B.; DUTT, R.; GOYAL, P.; MUKHERJEE, A. Spread of hate speech in online social media. In: **Proceedings of the 10th ACM conference on web science**. [S.l.: s.n.], 2019. p. 173–182.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.

MEYES, R.; LU, M.; PUISEAU, C. W. de; MEISEN, T. **Ablation Studies in Artificial Neural Networks**. 2019.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013.

MONDAL, M.; SILVA, L. A.; CORREA, D.; BENEVENUTO, F. Characterizing usage of explicit hate expressions in social media. **New Review of Hypermedia and Multimedia**, Taylor & Francis, v. 24, n. 2, p. 110–130, 2018.

NASSIF, A. B.; SHAHIN, I.; ATTILI, I.; AZZEH, M.; SHAALAN, K. Speech recognition using deep neural networks: A systematic review. **IEEE access**, IEEE, v. 7, p. 19143–19165, 2019.

NETO, E. F.; RODRIGUES, M. L. B. B. Z. Liberdade de expressão e discurso de ódio: o direito brasileiro à procura de um modelo. **Espaço Jurídico Journal of Law [EJJL]**, p. 1–36, 2021.

NOBATA, C.; TETREAULT, J.; THOMAS, A.; MEHDAD, Y.; CHANG, Y. Abusive language detection in online user content. In: **Proceedings of the 25th international conference on world wide web**. [S.l.: s.n.], 2016. p. 145–153.

PAETZOLD, G.; ZAMPIERI, M.; MALMASI, S. Utfpr at semeval-2019 task 5: Hate speech identification with recurrent neural networks. p. 519–523, 2019.

PAMUNGKAS, E. W.; BASILE, V.; PATTI, V. A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. **Information Processing and Management**, v. 58, n. 4, p. 102544, 2021. ISSN 0306-4573. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0306457321000510>.

PAMUNGKAS, E. W.; PATTI, V. Cross-domain and cross-lingual abusive language detection: A hybrid approach with deep learning and a multilingual lexicon. In: **Proceedings of the 57th annual meeting of the association for computational linguistics: Student research workshop**. [S.l.: s.n.], 2019. p. 363–370.

PARI, C.; NUNES, G.; GOMES, J. Avaliação de técnicas de word embedding na tarefa de detecção de discurso de ódio. In: **Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional**. Porto Alegre, RS, Brasil: SBC, 2019. p. 1020–1031. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/eniac/article/view/9354>.

PELICON, A.; SHEKHAR, R.; ŠKRLJ, B.; PURVER, M.; POLLAK, S. Investigating cross-lingual training for offensive language detection. **PeerJ Computer Science**, PeerJ Inc., v. 7, p. e559, 2021.

PELLE, R. P. de; MOREIRA, V. P. Offensive comments in the brazilian web: a dataset and baseline results. In: SBC. **Anais do VI Brazilian Workshop on Social Network Analysis and Mining**. [S.l.], 2017.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**. [S.l.: s.n.], 2014. p. 1532–1543.

PETERS, M. E.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K.; ZETTLEMOYER, L. Deep contextualized word representations. In: **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**. New Orleans, Louisiana: Association for Computational Linguistics, 2018. p. 2227–2237. Disponível em: <https://aclanthology.org/N18-1202>.

PIKULIAK, M.; ŠIMKO, M.; BIELIKOVA, M. Cross-lingual learning for text processing: A survey. **Expert Systems with Applications**, Elsevier, v. 165, p. 113765, 2021.

PITSILIS, G. K.; RAMAMPIARO, H.; LANGSETH, H. Effective hate-speech detection in twitter data using recurrent neural networks. **Applied Intelligence**, Springer, v. 48, n. 12, p. 4730–4742, 2018.

POLETTO, F.; BASILE, V.; SANGUINETTI, M.; BOSCO, C.; PATTI, V. Resources and benchmark corpora for hate speech detection: a systematic review. **Language Resources and Evaluation**, Springer, v. 55, n. 2, p. 477–523, 2021.

POLETTO, F.; STRANISCI, M.; SANGUINETTI, M.; PATTI, V.; BOSCO, C. Hate speech annotation: Analysis of an italian twitter corpus. In: CEUR-WS. **4th Italian Conference on Computational Linguistics, CLiC-it 2017**. [S.l.], 2017. v. 2006, p. 1–6.

RADFORD, A.; NARASIMHAN, K.; SALIMANS, T.; SUTSKEVER, I. Improving language understanding by generative pre-training. 2018.

RANASINGHE, T.; ZAMPIERI, M. Multilingual offensive language identification with cross-lingual embeddings. p. 5838–5844, 2020.

SALTON, G.; WONG, A.; YANG, C.-S. A vector space model for automatic indexing. **Communications of the ACM**, ACM New York, NY, USA, v. 18, n. 11, p. 613–620, 1975.

SAMMUT, C.; WEBB, G. I. **Encyclopedia of machine learning**. [S.l.]: Springer Science & Business Media, 2011.

SANGUINETTI, M.; POLETTO, F.; BOSCO, C.; PATTI, V.; STRANISCI, M. An italian twitter corpus of hate speech against immigrants. In: **Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)**. [S.l.: s.n.], 2018.

SAP, M.; CARD, D.; GABRIEL, S.; CHOI, Y.; SMITH, N. A. The risk of racial bias in hate speech detection. In: **Proceedings of the 57th annual meeting of the association for computational linguistics**. [S.l.: s.n.], 2019. p. 1668–1678.

SCHMIDT, A.; WIEGAND, M. A survey on hate speech detection using natural language processing. In: **Proceedings of the fifth international workshop on natural language processing for social media**. [S.l.: s.n.], 2017. p. 1–10.

SCHÜTZE, H.; MANNING, C. D.; RAGHAVAN, P. **Introduction to information retrieval**. [S.l.]: Cambridge University Press Cambridge, 2008. v. 39.

SCHWETER, S. **Italian BERT and ELECTRA models**. Zenodo, 2020. Disponível em: <https://doi.org/10.5281/zenodo.4263142>.

SHARMA, V. K.; MITTAL, N. Cross-lingual information retrieval: A dictionary-based query translation approach. In: **Advances in computer and computational sciences**. [S.l.]: Springer, 2018. p. 611–618.

SHEARER, E.; MITCHELL, A. 2021. (https://www.journalism.org/2021/01/12/news-use-across-social-media-platforms-in-2020/).

SILVA, A.; ROMAN, N. Hate speech detection in portuguese with naïve bayes, svm, mlp and logistic regression. In: SBC. **Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional**. [S.l.], 2020. p. 1–12.

SILVA, B. M. da. Liberdade de expressão e discurso de ódio no supremo tribunal federal: uma análise do rhc nº 146.303/rj à luz da crítica hermenêutica do direito. **Revista de Argumentação e Hermenêutica Jurídica, Virtual**, p. 01–20, 2020.

SILVA, S. C.; SERAPIÃO, A. B.; PARABONI, I. Hate-speech detection in portuguese using cnn and psycho-linguistic dictionary. **J. Inf. Data Manage.**, v. 5, p. 1–12, 2019.

SOARES, M. A. C.; PARREIRAS, F. S. A literature review on question answering techniques, paradigms and systems. **Journal of King Saud University-Computer and Information Sciences**, Elsevier, v. 32, n. 6, p. 635–646, 2020.

SOTO, C. P.; NUNES, G.; GOMES, J. G. R.; NEDJAH, N. Application-specific word embeddings for hate and offensive language detection. **Multimedia Tools and Applications**, Springer, p. 1–26, 2022.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Bertimbau: pretrained bert models for brazilian portuguese. In: SPRINGER. **Brazilian Conference on Intelligent Systems**. [S.l.], 2020. p. 403–417.

SRBA, I.; LENZINI, G.; PIKULIAK, M.; PECAR, S. Addressing hate speech with data science: an overview from computer science perspective. **Hate Speech-Multidisziplinäre Analysen und Handlungsoptionen**, Springer, p. 317–336, 2021.

STAPPEN, L.; BRUNN, F.; SCHULLER, B. Cross-lingual zero-and few-shot hate speech detection utilising frozen transformer language models and axel. **arXiv preprint arXiv:2004.13850**, 2020.

ULMER, D.; HARDMEIER, C.; FRELLSEN, J. **deep-significance - Easy and Meaningful Statistical Significance Testing in the Age of Neural Networks**. arXiv, 2022. Disponível em: <https://arxiv.org/abs/2204.06815>.

VAKULENKO, S.; LONGPRE, S.; TU, Z.; ANANTHA, R. Question rewriting for conversational question answering. In: **Proceedings of the 14th ACM International Conference on Web Search and Data Mining**. [S.l.: s.n.], 2021. p. 355–363.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2017. p. 5998–6008.

VENTUROTT, L. I.; CIARELLI, P. M. Data augmentation for improving hate speech detection on social networks. In: **Proceedings of the Brazilian Symposium on Multimedia and the Web**. [S.l.: s.n.], 2020. p. 249–252.

VIGNA, F. D.; CIMINO, A.; DELL'ORLETTA, F.; PETROCCHI, M.; TESCONI, M. Hate me, hate me not: Hate speech detection on facebook. In: **Proceedings of the First Italian Conference on Cybersecurity (ITASEC17)**. [S.l.: s.n.], 2017. p. 86–95.

WAGNER, K. 2020. (https://www.bloomberg.com/news/articles/2020-08-11/facebook-pulls-22-5-million-hate-speech-posts-in-second-quarter).

WANG, S.; ZHOU, W.; JIANG, C. A survey of word embeddings based on deep learning. **Computing**, Springer, v. 102, n. 3, p. 717–740, 2020.

WASEEM, Z.; HOVY, D. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In: **Proceedings of the NAACL student research workshop**. [S.l.: s.n.], 2016. p. 88–93.

WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D. A survey of transfer learning. **Journal of Big data**, SpringerOpen, v. 3, n. 1, p. 1–40, 2016.

WIEGAND, M.; RUPPENHOFER, J. Exploiting emojis for abusive language detection. 2021.

WU, Y.; SCHUSTER, M.; CHEN, Z.; LE, Q. V.; NOROUZI, M.; MACHEREY, W.; KRIKUN, M.; CAO, Y.; GAO, Q.; MACHEREY, K. et al. Google's neural machine translation system: Bridging the gap between human and machine translation. **arXiv preprint arXiv:1609.08144**, 2016.

YANG, Z.; DAI, Z.; YANG, Y.; CARBONELL, J.; SALAKHUTDINOV, R. R.; LE, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. **Advances in neural information processing systems**, v. 32, 2019.

YU, D.; DENG, L. **Automatic speech recognition**. [S.l.]: Springer, 2016. v. 1.

ZHANG, E.; ZHANG, Y. F-measure. In: _____. **Encyclopedia of Database Systems**. Boston, MA: Springer US, 2009. p. 1147–1147. ISBN 978-0-387-39940-9. Disponível em: <https://doi.org/10.1007/978-0-387-39940-9_483>.

ZHANG, Z.; ROBINSON, D.; TEPPER, J. Detecting hate speech on twitter using a convolution-gru based deep neural network. In: SPRINGER. **European semantic web conference**. [S.l.], 2018. p. 745–760.

ZHU, Y.; LU, S.; ZHENG, L.; GUO, J.; ZHANG, W.; WANG, J.; YU, Y. Texygen: A benchmarking platform for text generation models. In: **The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval**. [S.l.: s.n.], 2018. p. 1097–1100.

ZHUANG, F.; QI, Z.; DUAN, K.; XI, D.; ZHU, Y.; ZHU, H.; XIONG, H.; HE, Q. A comprehensive survey on transfer learning. **Proceedings of the IEEE**, IEEE, v. 109, n. 1, p. 43–76, 2020.