UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

PEDRO MORAIS FERNANDES

ANÁLISE DE SAÚDE MENTAL DE PACIENTES COM CÂNCER USANDO IA

Ouro Preto
2026

PEDRO MORAIS FERNANDES

ANÁLISE DE SAÚDE MENTAL DE PACIENTES COM CÂNCER USANDO IA

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Pedro Henrique Lopes Silva

Ouro Preto
2026

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

F363a Fernandes, Pedro Morais.
Análise de saúde mental de pacientes com câncer usando IA. [manuscrito] / Pedro Morais Fernandes. - 2026.
65 f.

Orientador: Prof. Dr. Pedro Henrique Lopes Silva Silva.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Biológicas. Graduação em Ciência da Computação .

1. Inteligência Artificial. 2. Câncer. 3. Dados - Análise. I. Silva, Pedro Henrique Lopes Silva. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004.8

Bibliotecário(a) Responsável: Renata Mara de Almeida - CRB-7: 6328

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
DEPARTAMENTO DE COMPUTACAO

FOLHA DE APROVAÇÃO

Pedro Morais Fernandes

Análise de Saúde Mental de Pacientes com Câncer Usando IA

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 27 de Fevereiro de 2026

Membros da banca

Pedro Henrique Lopes Silva (Orientador) - Doutor - Universidade Federal de Ouro Preto
Ederson Naves Fernandes Gonçalves Junior (Examinador) - Bacharel - Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Ouro Preto
Arthur Negrão de Faria Martins da Costa (Examinador) - Bacharel - Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Ouro Preto

Pedro Henrique Lopes Silva, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 27/02/2026

Documento assinado eletronicamente por **Pedro Henrique Lopes Silva**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 27/02/2026, às 14:07, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **1062473** e o código CRC **AB747FE3**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591692 - www.ufop.br

Dedico este trabalho, primeiramente a Deus, por todos esses anos de força que me deu, e aos meus pais Vanessa e Rodrigo.

Agradecimentos

Gostaria de dedicar este espaço para todas as pessoas que de alguma maneira contribuíram para que eu chegasse ate aqui.

Primeiramente, agradeço a Deus pela possibilidade de estar aqui para a realização deste trabalho, por toda força e sabedoria que me foi concedida.

Agradeço aos meus pais, Vanessa e Rodrigo, que com todo amor, suporte e apoio que me criaram, me possibilitaram chegar ate aqui, sou eternamente grato por todo o esforço e sacrifício feito para que eu tivesse uma educação de qualidade, obrigado por todo apoio e força que me deram, tento honrar cada esforço feito por vocês, todos os dias, para que eu me torne uma pessoa e um homem melhor, obrigado por serem os melhores pais do mundo, por terem formado quem eu sou hoje, amo muito vocês.

Agradeço também, minha namorada Maria Beatriz, que chegou na minha vida para agregar ainda mais, obrigado pelo suporte, por todo apoio, por cada conversa, me torno uma pessoa e um homem melhor e mais forte tendo você ao meu lado, te amo.

Agradeço muito aos meus avós, Luiz, Ení e Alice, que deram todo o suporte aos meus pais na minha criação, muito obrigado por tudo que fizeram e fazem por mim, amo vocês.

Um Agradecimento também aos meus tios, Hellyezer, Érica e Fabrício, muito obrigado por todo o apoio e carinho.

Um agradecimento a todos da minha família, que de certa forma contribuíram para eu chegar até onde estou, amo todos de coração.

Agradeço ao meu irmão de família diferente Gabriel, estamos nessa jornada de estudos e amizade a muito tempo, muito bom crescer junto de você meu irmão, e pelas conversas e com a parceria de sempre, sei que vamos chegar longe, tenho certeza, que de certa forma, também estou aqui pela nossa amizade e por toda a maluquice de estudos que passamos no ensino médio com matemática, física e química, e agora aos poucos estamos crescendo profissionalmente, uma alegria imensa observar este progresso. Ademais, essa amizade me levou a conhecer pessoas incríveis da família como, Miquelina, minha madrinha de crisma e Daniel meu padrinho, que tanto me ajudaram e agregaram pelo meu crescimento, como segundos pais na minha vida, sempre me recebendo bem e oferecendo todo suporte sempre que eu precisasse e sempre fazendo questão de me ter por perto, obrigado por todo o carinho e por se tornarem minha família também e me considerarem parte da família. Um agradecimento a outras pessoas dessa família tao especial como Mirla, Camille, Magno e Gessica e Maria Vitoria por todo o carinho de sempre.

Agradecimento também a meus dois outros irmãos de família diferente Vitor e Eduardo e todos os meus amigos de Manhuaçu, muito obrigado pela parceria de sempre, todos esses anos.

Agradeço aos meus irmãos de graduação, Nicolas, Felipe, Matheus, Lucas e Pedro Henrique, espero levar a amizade de vocês pelo resto da vida, com certeza contribuíram para que a graduação fosse mais agradável, que a parceria dure pra sempre!

Por fim, agradeço ao meu orientador Pedro Silva, por ter aceitado fazer parte deste trabalho comigo, um dos professores que com certeza contribuíram muito bem para meu conhecimento e interesse na área, principalmente de Inteligencia Artificial, muito obrigado por todo suporte e ajuda, por fazer parte da minha jornada, agradeço imensamente.

Esquecendo-me das coisas que para trás ficam, eu prossigo para o alvo.
(BíBLIA SAGRADA, 2011, Filipenses 3:13)

Resumo

O câncer é uma doença grave que compromete tanto a saúde física quanto a mental dos pacientes, e com isso, tem-se um grande desafio mediante a mitigação da doença fisicamente, mas ao mesmo tempo, há a necessidade de manter a pessoa forte mentalmente, para que a recuperação seja realizada. Visando auxiliar no fortalecimento mental de pacientes com câncer, a Inteligência Artificial tem-se mostrado eficaz para processar grandes cargas de dados e obter um modelo que auxilie nesta tarefa. Em muitos casos, os conjuntos de dados encontram-se desbalanceados, o que pode comprometer o treinamento e a generalização do modelo treinado. Portanto, este trabalho propõe um modelo de Inteligência Artificial para classificar sentimentos expressos em textos de pacientes oncológicos, aplicando-se técnicas de balanceamento de dados, para uma diminuição do viés do modelo em apenas um determinado grupo de classificações, visando auxiliar no monitoramento do estado emocional e no suporte psicológico durante o tratamento com eficiência, a fim de garantir a qualidade de vida psicológica do paciente em um momento de extrema vulnerabilidade. Utilizando técnicas para lidar com o desbalanceamento da base de dados utilizada, o modelo alcançou 82% de acurácia com o modelo LLM.

Palavras-chave: Câncer. Balanceamento de dados. Inteligência Artificial.

Abstract

Cancer is a serious disease that affects both the physical and mental health of patients. Thus, there is a major challenge not only in mitigating the disease physically but also in ensuring that patients remain mentally strong throughout the recovery process. To support the mental resilience of cancer patients, artificial intelligence has proven effective in processing large amounts of data and developing models to assist in this task. In many cases, however, datasets are imbalanced, which may compromise the training and generalization of the resulting model. Therefore, this work proposes an AI-based model to classify sentiments expressed in the texts of oncology patients, applying data balancing techniques to reduce model bias toward specific classes. The goal is to contribute to the monitoring of emotional states and provide psychological support during treatment effectively, thereby ensuring patients' psychological well-being in a moment of extreme vulnerability. By employing techniques to handle data imbalance, the proposed model achieved an accuracy of 82% with LLM model.

Keywords: Cancer; Data balancing; Artificial Intelligence.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

ABSA *Aspect-Based Sentiment Analysis*. 29

BERT *Bidirectional Encoder Representations from Transformers*. x, xv, xvi, 1–3, 21, 22, 29, 35, 38, 39, 42–44, 46, 47, 49, 51, 53, 54

BiLSTM *Bidirectional Long Short Term Memory Neural Network*. 28

BPTT *Backpropagation Through Time*. 16

CNN *Convolutional Neural Network*. x, xv, xvi, 2, 3, 18, 19, 28, 29, 35, 38, 39, 42–44, 46, 49, 51, 54

DAM *Domain Attention Model*. 30

DAMA *Dynamic Domain Information Modulation Algorithm*. 30

EDA *Easy Data Augmentation*. x, xii, xvi, 2, 25, 32, 33, 47, 49, 50, 53, 54

FN *Falso Negativo* . 26

FP *Falso Positivo* . 26

GELU *Gaussian Error Linear Unit*. 12, 37, 38, 40

GenAI *Generative Artificial Intelligence*. xi, xvi, 25, 32, 34, 49, 52–54

GRU *Gated Recurrent Unit*. 29

IA Inteligência Artificial. vi, xv, 1, 2, 4, 7

LLM *Large Language Model*. vi, vii, x, xv, xvi, 2, 3, 21–24, 35, 39, 40, 42–44, 46, 47, 49, 51, 53–55

LoRA *Low-Rank Adaptation*. xv, 23, 24, 39, 55

LSTM *Long Short Term Memory Neural Network*. x, xv, 2, 3, 16–18, 28–30, 35–37, 42–44, 46, 47, 49, 51, 54

MSE *Mean Square Error* . 13

MTL *Multi Task Learning*. 30

NER *Named Entity Recognition*. 29

OPAS *Organização Pan-Americana da Saúde*. 1

PLN Processamento de Linguagem Natural. x, xv, 1, 2, 7, 8, 22, 28

RNN *Recurrent Neural Network*. xv, 2, 14–17, 20

RoPE *Rotary Position Embedding*. 23

SMOTE *Synthetic Minority Over-sampling Technique*. 25, 29

VN *Verdadeiro Negativo* . 26

VP *Verdadeiro Positivo* . 26

xLSTM *Extended Long Short Term Memory Neural Network*. x, xv, 2, 3, 16, 18, 30, 35, 37, 42–45, 49, 51, 54

Sumário

1 Introdução

O termo câncer refere-se a um grupo de doenças graves e potencialmente letais que despertam grande preocupação tanto nos pacientes quanto em seus familiares. Essa designação abrange uma ampla variedade de enfermidades malignas, caracterizadas principalmente pela capacidade de disseminação das células cancerígenas por diferentes partes do organismo, afetando tecidos e órgãos diversos (INCA, 2022). A doença representa uma das principais causas de morbidade e mortalidade nas Américas. De acordo com a Organização Pan-Americana da Saúde (OPAS), no ano de 2022, foram registrados aproximadamente 4,2 milhões de novos casos de câncer na região, sendo projetado um crescimento de 60% nesse número até o ano de 2045 (OPAS, 2025).

De maneira geral, o câncer não apenas compromete fisicamente os pacientes, mas também exerce um forte impacto emocional, afetando profundamente seu bem-estar psicológico. Entre os transtornos mais recorrentes em indivíduos diagnosticados com a doença, destaca-se a depressão, que se manifesta com frequência diante do sofrimento físico, das incertezas quanto ao prognóstico e das alterações na qualidade de vida (BOTTINO; FRáGUAS; GATTAZ, 2009). Diante desse cenário, torna-se evidente a necessidade de suporte psicológico especializado, capaz de amparar os pacientes durante as diferentes etapas do tratamento.

Nesse contexto, diversas abordagens podem ser utilizadas para avaliar o estado emocional do paciente, sendo a Inteligência Artificial (IA) e o Processamento de Linguagem Natural (PLN) ferramentas promissoras na tarefa de análise de sentimentos. Esses recursos têm-se mostrado eficazes na detecção de padrões emocionais a partir de grandes volumes de texto, como no relevante estudo de Yazdani et al. (2023), no qual foi desenvolvido um sistema de análise de sentimentos voltado para pacientes hospitalizados no Irã. A proposta alcançou acurácias de 89,3%, 92,6% e 90,8% em três categorias distintas: serviços gerais, cuidados de saúde e expectativa de vida, respectivamente, utilizando dados extraídos de formulários preenchidos pelos próprios pacientes.

Embora existam iniciativas promissoras baseadas em dados controlados e estruturados, o uso de técnicas de IA e PLN ainda apresenta um vasto campo a ser explorado, especialmente no que tange à análise emocional de pacientes oncológicos. A automatização do processamento de textos livres, como depoimentos, comentários e registros pessoais, oferece uma alternativa eficaz à escuta clínica tradicional, possibilitando intervenções psicológicas mais ágeis e direcionadas. Esse potencial é evidenciado no estudo de Wu et al. (2024), que demonstrou o desempenho expressivo do modelo *Bidirectional Encoder Representations from Transformers* (BERT) em tarefas de análise de sentimentos, alcançando acurácias de 91,3% em sua versão básica e 92,7% em uma arquitetura mais robusta. Paralelamente, modelos baseados em redes neurais recorrentes,

como a *Long Short Term Memory Neural Network* (LSTM), também demonstram resultados promissores nessa tarefa. É valido ressaltar que tais ferramentas compostas por IA não visam substituir o papel de psicólogos e psiquiatras, mas sim servem como um importante auxilio.

Além disso, conforme apresentado por Hussain e Naseer (2023), a utilização de uma arquitetura com duas camadas LSTM permitiu a obtenção de acurácias de 86,23% e 87,65% na classificação de sentimentos em avaliações de filmes, evidenciando a eficiência desses modelos em cenários de linguagem natural. Além disso, trabalhos como (DONG et al., 2020), (YUE; LI, 2025) e (ZERKOUK; MIHOUBI; CHIKHAOUI, 2025) evidenciam certa eficiência e resultados promissores no campo de análise de sentimento ao utilizar modelagem híbrida, xLSTM e LLMs.

Diante dessas considerações em relação as arquiteturas de PLN e ao panorama clínico relacionado ao câncer, este trabalho propõe o desenvolvimento de um modelo de classificação de sentimentos voltado para pacientes oncológicos. A proposta baseia-se no uso de técnicas de PLN com o objetivo de identificar padrões emocionais que possam subsidiar a oferta de cuidados paliativos mais eficientes. Para isso, foram conduzidos experimentos utilizando um *corpus* disponibilizado pela plataforma *Kaggle* (ORCHI et al., 2023), que reúne *posts* de redes sociais voltadas à saúde, como *Reddit*, *DailyStrength* e *HealthBoards*, o que possibilitou o treinamento de modelos baseados em *Recurrent Neural Network* (RNN) como *Long Short Term Memory Neural Network* (LSTM) e *Extended Long Short Term Memory Neural Network* (xLSTM), assim como modelos baseados na arquitetura de *Transformers*, como *Bidirectional Encoder Representations from Transformers* (BERT) e uma arquitetura hibrida, ou seja, que une a estrutura do modelo BERT com *Convolutional Neural Networks* (CNN), além do uso do *Llama*, um *Large Language Model* (LLM).

Os resultados evidenciam que os modelos baseados em mecanismos de atenção alcançaram as maiores acurácias ao longo dos experimentos. O BERT obteve acurácia máxima de 81,83% com o uso de dados sintéticos e *Class Weights*, enquanto o LLM atingiu 82% no cenário que combinou técnicas de aumento de dados (*Easy Data Augmentation* (EDA)) e ponderação de classes, representando o melhor resultado global observado. O modelo híbrido apresentou acurácias próximas, variando entre 78% e 80%, dependendo da estratégia aplicada. Em contraste, os modelos recorrentes apresentaram desempenho inferior, com a xLSTM alcançando acurácias entre 69% e 75%, e a LSTM tradicional permanecendo abaixo de 72% em todos os cenários. Esses resultados confirmam a superioridade dos modelos baseados em atenção em termos de acurácia para a tarefa avaliada.

1.1 Justificativa

De acordo com (OMS, 2020), os cuidados paliativos são de extrema importância quando uma pessoa possui um quadro clínico extremamente grave. Uma parte essencial do processo de recuperação é oferecer suporte e alívio ao paciente, seja físico ou psicológico. Essa atitude é

uma responsabilidade ética global de qualquer profissional de saúde. Além disso, esta instituição estima que apenas 14% dos pacientes que necessitam de atendimentos paliativos realmente os recebem, o que demonstra a necessidade de mudança nessa estatística notavelmente baixa.

À vista disso, este trabalho busca colaborar no combate ao câncer e contribuir com o auxílio e a melhoria da eficiência no fornecimento de cuidados paliativos, como o tratamento psicológico, por meio do estudo e aprimoramento de técnicas de análise sentimental, para que haja o devido cuidado com o paciente e que sua saúde mental esteja fortalecida para a cura física.

1.2 Objetivos

O objetivo central deste trabalho é investigar e comparar o desempenho de diferentes arquiteturas de Redes Neurais na classificação de sentimentos em textos de pacientes oncológicos. A pesquisa abrange desde modelos baseados em recorrência, como LSTM e sua evolução xLSTM, até o estado da arte em processamento de linguagem natural, representado pela arquitetura *Transformers* com o uso do BERT. Adicionalmente, avalia-se a eficácia de uma abordagem híbrida, integrando BERT e CNN, bem como a capacidade de inferência de um *Large Language Model* (LLM) por meio do modelo *Llama*.

Para atingir o objetivo principal, os seguintes objetivos secundários são necessários:

- Avaliar diferentes abordagens para a classificação do problema em questão utilizando LSTM e sua evolução, a xLSTM.
- Avaliar abordagens que utilizam a arquitetura dos *Transformers*, como o BERT, uma abordagem hibrida utilizando CNN em sua composição estrutural e por fim, uma abordagem utilizando um LLM como *Llama*.
- Aplicar métricas que permitam ter clareza sobre a qualidade do modelo investigado em relação ao problema.
- Investigar e realizar experimentos com técnicas de balanceamento de dados.

1.3 Organização do Trabalho

Este trabalho é organizado com a seguinte estrutura: Capítulo 2 apresenta o referencial teórico que serve como apoio a cada termo utilizado para a compreensão do que está sendo abordado; uma revisão da literatura sobre trabalhos similares à problemática exposta é apresentada no Capítulo 3; no Capítulo 4 é apresentada toda a metodologia a ser aplicada no trabalho; no Capítulo 5 são mostrados os resultados obtidos com as experimentações; por fim, no Capítulo 6 são apresentadas as conclusões obtidas e os trabalhos futuros.

2 Referencial Teórico

Este capítulo apresenta os conceitos teóricos referentes ao conteúdo abordado nesta monografia, como Inteligência Artificial na seção 2.1, com um contexto sobre o âmbito de IA e *Machine Learning*; na seção 2.2, explica-se sobre o uso de linguagem natural em modelos de IA e detalhes acerca do tema; na seção 2.3, explica-se sobre componentes e conceitos básicos acerca do que constitui uma rede neural artificial;Por fim, na seção 2.4 e na seção 2.5, são apresentadas técnicas de balanceamento de dados e algumas métricas de avaliação que são utilizadas para avaliar modelos de aprendizado de máquina.

2.1 Inteligência Artificial e Aprendizado de máquina

O termo Inteligência Artificial esta atrelado a automação de tarefas que são comumente feitas pelo ser humano. Portanto, dentro deste contexto, aprendizado de maquina seria uma série de técnicas e métodos que o computador usa para aprender a fazer determinadas tarefas (ASHENDEN et al., 2021).

Dentre as subáreas envoltas pela IA, *Machine Learning* se destaca como uma de suas principais áreas, desempenhando um papel importante na sociedade contemporânea, abrangendo desde mecanismos de busca na internet até sistemas de recomendação em plataformas de *e-commerce* e a filtragem de conteúdos em redes sociais. Historicamente, a construção desses sistemas dependia de um trabalho cuidadoso de engenharia e de conhecimento especializado do domínio, necessário para criar extratores de características capazes de converter dados brutos em representações apropriadas para algoritmos de classificação. Mas com a evolução do *Deep learning*, esse paradigma foi transformado pela introdução de técnicas de aprendizagem de representação, nas quais os próprios modelos passam a identificar automaticamente as representações mais adequadas diretamente a partir dos dados originais. Esse aprendizado é viabilizado por arquiteturas compostas por diversas camadas de módulos não lineares, que refinam progressivamente a informação em representações cada vez mais abstratas, possibilitando a modelagem e o aprendizado de funções de alta complexidade(LECUN YOSHUA BENGIO, 2015).

Diante desse cenário, o *Machine Learning* engloba diferentes paradigmas de aprendizado, entre os quais se destacam o aprendizado supervisionado, o aprendizado semi-supervisionado, o aprendizado não supervisionado e o aprendizado por reforço(BERGMANN, c).

O aprendizado supervisionado é a abordagem mais amplamente utilizada em *machine learning* (BELSIC; STRYKER, 2024). Ele baseia-se no treinamento de modelos a partir de grandes conjuntos de dados previamente rotulados. Durante esse processo, o sistema recebe uma entrada e produz uma saída, como um vetor de pontuações associado às classes possíveis.

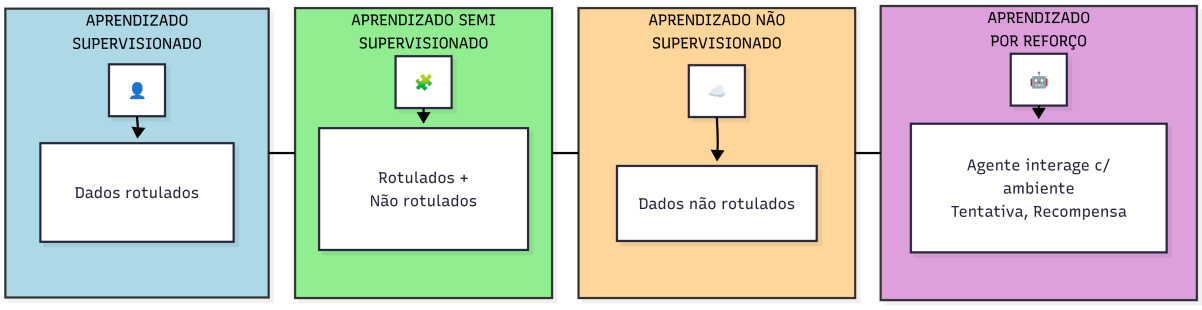
Essa saída é comparada ao resultado esperado por meio de uma função de erro, que quantifica a discrepância entre ambos. Para minimizar esse erro, o algoritmo ajusta seus parâmetros internos, conhecidos como pesos, com base nas informações do gradiente. Ao final do treinamento, o desempenho do modelo é avaliado por sua capacidade de generalização, isto é, sua habilidade de produzir respostas corretas para dados inéditos que não foram apresentados durante a fase de treinamento (LECUN YOSHUA BENGIO, 2015).

Por sua vez, o aprendizado não supervisionado ocorre quando o sistema recebe apenas os dados de entrada, sem qualquer rótulo ou saída desejada associada. Nesse contexto, a rede neural aprende de forma autônoma a identificar padrões, estruturas ou relações ocultas nos dados. O objetivo principal não é prever uma saída específica, mas compreender a distribuição dos dados e organizá-los de acordo com suas similaridades (GUPTA GAURAV KHATRI, 2025).

O aprendizado semi-supervisionado pode ser entendido como uma abordagem intermediária entre os paradigmas supervisionado e não supervisionado. Nesse método, o treinamento é realizado utilizando uma pequena quantidade de dados rotulados em conjunto com um grande volume de dados não rotulados, buscando explorar as vantagens de ambos os cenários para melhorar o desempenho do modelo (BERGMANN, b).

Por fim, o aprendizado por reforço baseia-se na interação contínua de um agente com um ambiente. Diferentemente do aprendizado supervisionado, não existem respostas corretas explícitas para cada entrada. Em vez disso, o sistema aprende por meio de recompensas e penalidades recebidas após a execução de ações. O objetivo é maximizar a recompensa acumulada ao longo do tempo, ajustando o comportamento do agente com base no *feedback* obtido (GUPTA GAURAV KHATRI, 2025).

Figura 2.1 – Imagem que ilustra como os tipos de aprendizado funcionam em aprendizado de máquina



Fonte: Próprio Autor.

Em suma, tais tipos de aprendizado servem como a base do treinamento e aprendizado de modelos de aprendizado de máquina mas a eficácia desses paradigmas de aprendizado, entretanto, não depende apenas do volume de dados, mas da capacidade do modelo em equilibrar a memorização e a generalização, evitando fenômenos como o *overfitting* e o *underfitting*.

2.1.1 *Overfitting* e *Underfitting*

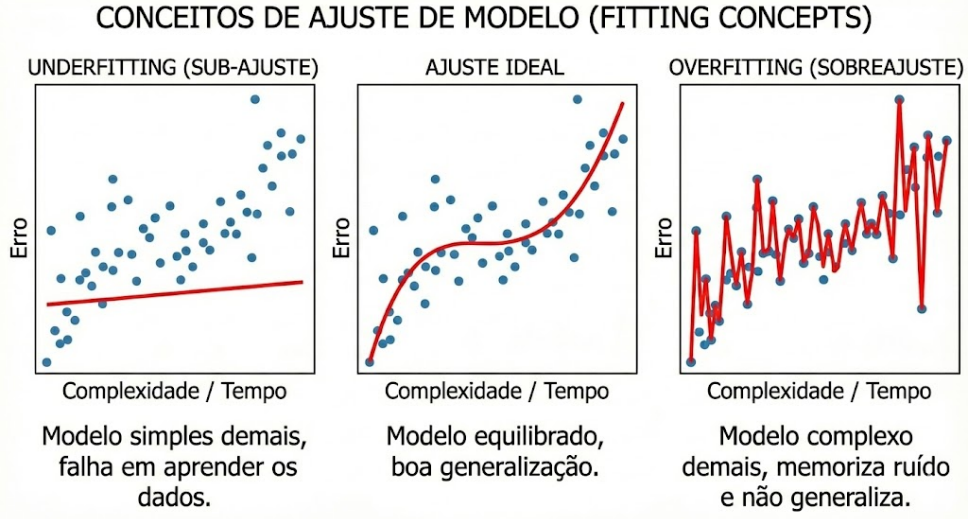
O *overfitting* é um problema recorrente no aprendizado supervisionado e ocorre quando um modelo apresenta excelente desempenho sobre os dados de treinamento, mas falha ao generalizar adequadamente para dados novos ou não vistos. Esse fenômeno indica que o modelo aprendeu não apenas os padrões relevantes presentes nos dados, mas também características específicas e ruídos do conjunto de treinamento, comprometendo sua capacidade de generalização (YING, 2019).

Estes modelos que sofrem de *overfitting* tendem a memorizar os dados de treinamento em vez de aprender as relações subjacentes que governam o problema. Como consequência, pequenas variações nos dados de entrada podem resultar em grandes variações na saída do modelo, tornando-o instável e pouco confiável em cenários reais. Esse comportamento é particularmente comum em modelos excessivamente complexos, que possuem um grande número de parâmetros em relação à quantidade ou à qualidade dos dados disponíveis(YING, 2019).

As principais causas do *overfitting* estão relacionadas a três fatores centrais: a presença de ruído nos dados de treinamento, o tamanho limitado ou pouco representativo do conjunto de dados e a elevada complexidade do modelo. Conjuntos de dados pequenos ou ruidosos aumentam a probabilidade de o modelo aprender padrões irrelevantes, enquanto modelos com grande capacidade expressiva tendem a se ajustar excessivamente aos dados observados, reduzindo sua consistência quando expostos a novos exemplos (YING, 2019), sendo necessário o uso de estratégias que foquem em mitigar esse problema e a adaptar-se melhor ao problema enfrentado no treinamento do modelo.

Já o *underfitting*, é o oposto do que o *overfitting* significa. Este problema acontece quando o modelo é incapaz de capturar as variações de dados no conjunto(JABBAR; KHAN, 2015), fazendo com que o treinamento, teste e validação sejam comprometidos.

Figura 2.2 – Representações gráficas de *underfitting*, *overfitting* e o estado ideal.



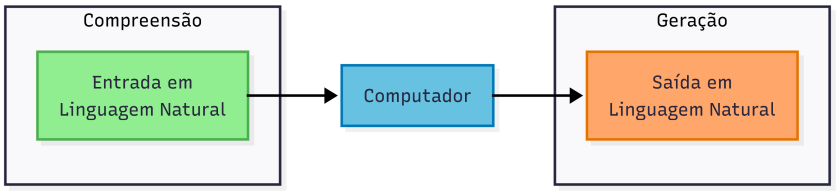
Fonte: Próprio Autor.

Portanto, uma vez estabelecido o equilíbrio no aprendizado do modelo, torna-se possível aplicá-lo em domínios complexos e específicos, como o processamento de dados textuais, discutido na seção seguinte sob a ótica da Processamento de Linguagem Natural.

2.2 Processamento de Linguagem Natural

A área de Processamento de Linguagem Natural (PLN) constitui um subcampo da Ciência da Computação e da IA, cujo objetivo é possibilitar que computadores processem a linguagem humana e, a partir desse processamento, sejam capazes comunicar-se. Tal comunicação é viabilizada por meio da combinação da linguagem com modelos estatísticos amplamente utilizados no domínio da IA, tais como *Machine Learning* e *Deep Learning* (IBM, 2024b). Em síntese, o PLN é composto por duas tarefas fundamentais: o processamento da linguagem e a geração de uma saída, seja uma classificação ou um texto (JOSHI et al., 2025).

Figura 2.3 – Imagem que representa o entendimento e a geração de textos em PLN.



Fonte: Próprio Autor.

A partir dessa perspectiva, a computação linguística que fundamenta o PLN abrange duas dimensões analíticas principais: análise sintática e análise semântica (IBM, 2024b). Ambas operam de forma complementar para garantir a interpretação coerente do conteúdo textual. A análise sintática é responsável por determinar a estrutura das palavras, frases ou sentenças, aplicando regras gramaticais para organizar o texto. Em sequência, a análise semântica utiliza os resultados obtidos na etapa anterior para extrair o significado do conteúdo e interpretar as informações de maneira contextualizada (IBM, 2024b).

Para que esse processo seja efetivado, torna-se imprescindível aplicar técnicas de pré-processamento textual, favorecendo a compreensão por parte dos algoritmos de *Machine Learning*(TABASSUM; PATIL, 2020), sendo a tokenização uma das etapas iniciais. Essa técnica consiste em segmentar o texto em unidades menores — como palavras, frases ou sentenças — com o intuito de reduzir a complexidade do processamento. Subsequentemente, realizam-se procedimentos adicionais, tais como a conversão de todos os *tokens* para letras minúsculas, a fim de evitar distinções entre termos idênticos que diferem apenas pelo uso de maiúsculas, como “Cabelo” e “cabelo”. Ademais, são eliminados elementos irrelevantes, como espaços em branco, pontuações e *stop words*, que não possuem relevância semântica (THANAKI, 2017).

Após essa etapa, procede-se à transformação dos *tokens* em representações numéricas, possibilitando a aplicação de algoritmos de aprendizado e técnicas de análise textual. Essa

conversão viabiliza que os modelos baseados em PLN adquiram conhecimento a partir das informações processadas, permitindo sua utilização em cenários práticos e a geração de dados relevantes (CHOPRA; PRASHAR; SAIN, 2013).

Não obstante os avanços e a eficiência das arquiteturas que utilizam PLN, essa abordagem ainda enfrenta desafios significativos, especialmente no que se refere a aspectos como viés e imparcialidade (BANSAL, 2022), fatores que podem comprometer a equidade e a confiabilidade das aplicações no mundo real.

Em conclusão, o PLN revela-se uma tecnologia essencial no contexto contemporâneo, uma vez que possibilita a comunicação entre humanos e máquinas por meio da linguagem natural, recurso universal que permeia as interações sociais (IBM, 2024b). Apesar da complexidade semântica da linguagem humana e as interações sociais, o sucesso das aplicações de PLN reside na capacidade de traduzir textos em representações numéricas processáveis por arquiteturas de Redes Neurais Artificiais.

2.3 Redes Neurais Artificiais

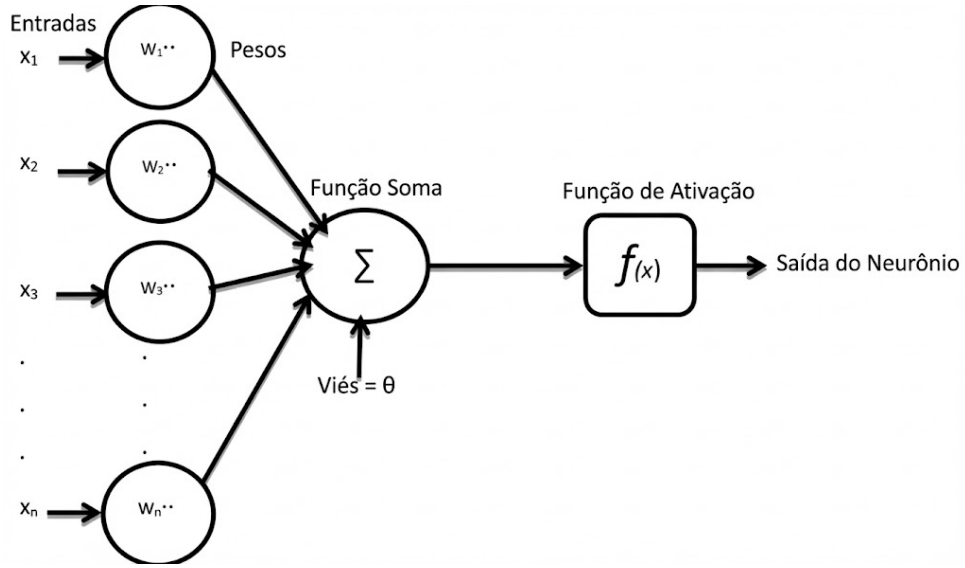
As redes neurais artificiais configuram-se como estruturas computacionais inspiradas na organização e no funcionamento do cérebro humano (GOODFELLOW; BENGIO; COURVILLE, 2018). Essas redes são constituídas por unidades de processamento denominadas neurônios artificiais, organizadas em camadas e conectadas por meio de ligações ponderadas. Tais modelos têm-se destacado como ferramentas eficazes na realização de tarefas relacionadas ao reconhecimento de padrões (KHAN; AFZAL; LEE, 2022), à previsão (BOCHENEK; USTRNUL, 2022) e à classificação de dados (YAQOOB; AZIZ; VERMA, 2023). De modo análogo ao cérebro humano, essas estruturas matemáticas requerem dados como insumo para a realização do processo de aprendizado, o que lhes permite executar tarefas com maior eficiência. Dessa forma, as redes neurais constituem a base estrutural dos algoritmos de *Deep Learning*.

Para compreender como essas redes realizam tais abstrações, é necessário detalhar o funcionamento de sua unidade fundamental, o neurônio artificial, e como sua organização em múltiplas camadas define a profundidade do aprendizado.

2.3.1 Neurônios

Cada neurônio de uma rede neural é composto por uma entrada, pesos, um viés, que representa um limite imposto, e uma saída. Os neurônios encontram-se interligados, e esse tipo de conexão permite que a rede transmita informações entre os nós. Quando a saída de um neurônio ultrapassa determinado valor de limiar, essa saída pode ser encaminhada como entrada para outro neurônio. Esse mecanismo de processamento, no qual uma entrada gera uma saída, é regido pela função de ativação presente em cada neurônio (HAN et al., 2019).

Figura 2.4 – Exemplo de um neurônio e seus componentes.



Fonte: Adaptado de (YACIM; BOSHOFF, 2018).

Tendo em vista tais neurônios, pode-se conceituar o significado de uma rede multicamadas, formada por diversos neurônios. No contexto das redes neurais multicamadas, entre a camada de entrada e a camada de saída, encontram-se as chamadas camadas ocultas (NIELSEN, 2015). Essas camadas são compostas por diversos neurônios, responsáveis por transformar os dados brutos em representações progressivamente mais abstratas. A principal função das camadas ocultas é permitir que a rede aprenda relações complexas e não lineares entre as variáveis de entrada e a saída desejada. Quanto maior o número de camadas ocultas, maior a profundidade da rede(GOODFELLOW; BENGIO; COURVILLE, 2018), o que possibilita o desenvolvimento de modelos mais expressivos e capazes de resolver tarefas de alta complexidade, como reconhecimento de padrões visuais, processamento de linguagem natural, entre outras aplicações em inteligência artificial.

A cada etapa desse processo direto, os pesos e o viés de cada neurônio são ajustados por meio do algoritmo de *backpropagation*.

2.3.2 *Forward Propagation* e *Back Propagation*

O processo de *forward propagation* constitui uma etapa fundamental no funcionamento das redes neurais artificiais, sendo responsável pela geração das predições do modelo. Nesse procedimento, os dados de entrada são propagados sequencialmente através das camadas da rede, desde a camada de entrada até a camada de saída. Em cada camada, os sinais recebidos são combinados por meio de operações envolvendo pesos e vieses associados aos neurônios, e o resultado dessa combinação é submetido a uma função de ativação. A aplicação dessa função é essencial para introduzir não linearidades no modelo, permitindo que a rede represente e aprenda relações complexas e não lineares presentes nos dados(LECUN YOSHUA BENGIO, 2015).

Backpropagation tem como principal objetivo calcular, de forma eficiente, o quanto cada peso da rede contribuiu para o erro observado, o que significa alterar tanto o peso, quanto o viés de um modelo contribui para mudanças na rede neural como um todo(NIELSEN, 2015). Esse ajuste é realizado por meio de métodos do cálculo diferencial, que permitem identificar como o erro da rede varia em função de seus parâmetros internos. A partir dessa análise, os pesos e os vieses são gradualmente atualizados, normalmente com o auxílio de algoritmos de otimização como o Gradiente Descendente. O objetivo desse processo é reduzir o erro total do modelo, de modo que, a cada iteração do treinamento, a rede aprimore seu desempenho e produza predições cada vez mais próximas dos valores esperados(BERGMANN, a).

O treinamento de uma rede neural, portanto, é caracterizado por um ciclo contínuo, no qual os dados são processados na etapa de *forward propagation*, o erro é calculado, e o *backpropagation* ajusta os parâmetros do modelo. Esse ciclo se repete por diversas épocas (ou iterações), permitindo que a rede aprenda progressivamente a resolver a tarefa para a qual foi projetada.

Tendo o entendimento deste ajuste de pesos, torna-se necessário a introdução de não linearidade no processo, para a aprendizagem de padrões mais complexos em conjunto com este procedimento, o que é devidamente empregado pelas funções de ativação.

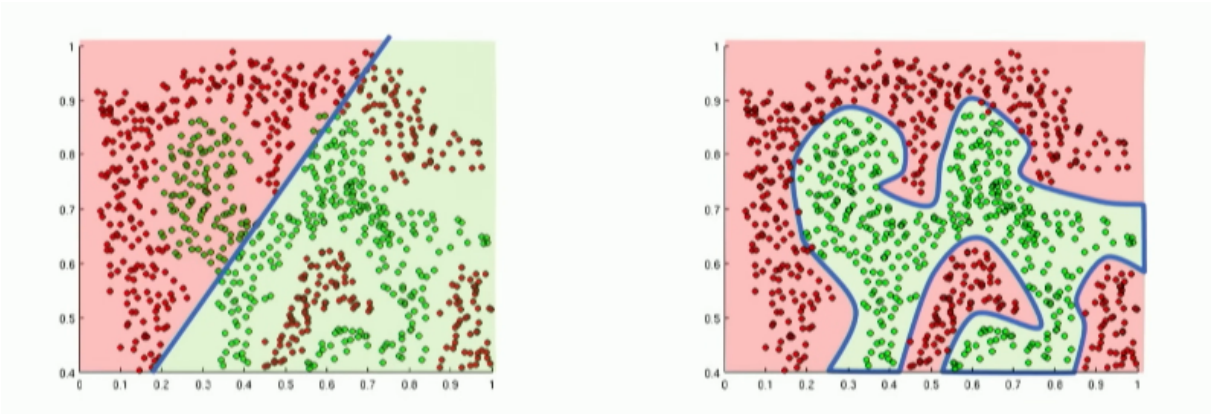
2.3.3 Função de Ativação

As funções de ativação exercem um papel fundamental no funcionamento das redes neurais (DUBEY; SINGH; CHAUDHURI, 2022). Tais funções são responsáveis por determinar se um neurônio será ativado ou não, o que implica que, caso a informação recebida por determinado neurônio seja considerada relevante, este será ativado. Essa ativação promove uma transformação não linear dos dados de entrada, possibilitando que a saída gerada por esse neurônio seja utilizada como entrada para o neurônio subsequente. Esse processo se repete até que a informação alcance as camadas de saída da rede neural, resultando em uma resposta correspondente à entrada inicial fornecida.

Essencialmente, a função de ativação permite que a rede neural seja capaz de aprender e executar tarefas mais complexas, especialmente aquelas de natureza não linear. Dessa forma, a rede não se limita apenas à resolução de tarefas lineares e simples, mas torna-se apta a lidar com problemas de maior complexidade, ampliando significativamente sua capacidade de generalização e aprendizado.

À vista disso, para o algoritmo de *backpropagation*, que é amplamente utilizado em arquiteturas de aprendizado profundo para realizar o processo de aprendizagem entre neurônios, é importante que as funções de ativação possuam derivação, pois o processo de atualização é através das derivadas dos parâmetros nas arquiteturas dos modelos (RAMACHANDRAN; ZOPH; LE, 2017).

Figura 2.5 – Exemplo de funções lineares (a) e funções não lineares(b).As imagens mostram a diferença existente entre as diferentes funções, retratando o contexto da importância das funções de ativação.



(a) (b)

Fonte: Amini (2023).

Dessa forma, (GUSTINELI, 2022) apresenta em seu trabalho, exemplos de funções de ativação bastante utilizadas:

- *Sigmoid*: A função *sigmoid* é uma função matemática que, ao receber qualquer número real como entrada, transforma-o em um valor pertencente ao intervalo entre 0 e 1. Embora possua um custo computacional relativamente elevado devido à presença de uma operação exponencial em sua composição, essa desvantagem é compensada por dois fatores relevantes: a introdução de não-linearidade à arquitetura e a simplicidade computacional do seu gradiente, o que favorece a etapa de *backpropagation* durante o treinamento da rede neural. Ela é definida matematicamente por:

$$\mathrm{f(x)} = \frac{1}{1 + e^{-x}}. \tag{2.1}$$

- *Tahn*: A função *tanh* (tangente hiperbólica) é uma função contínua, diferenciável e não linear, cujo intervalo de saída está compreendido entre -1 e 1. Uma de suas principais vantagens em relação à função *sigmoid* é o fato de ser centrada em zero, ou seja, seus valores se distribuem em torno do ponto zero. Essa característica contribui para um gradiente mais equilibrado durante o processo de *backpropagation*, favorecendo a convergência do modelo e a estabilidade no treinamento de redes neurais. Ela é definida matematicamente por:

$$\mathrm{f(x)} = \frac{e^{x} + e^{-x}}{e^{x} + e^{-x}}. \tag{2.2}$$

- *ReLU*: A função *ReLU* é extremamente usada no contexto de redes neurais, devido as suas propriedades, que são: é contínua e não centrada em zero. Além disso, possui um baixo custo computacional por não ser exponencial e também força valores negativos a serem 0. Ela é definida matematicamente por:

$$\mathrm{f(x)} = \max(0, x) \tag{2.3}$$

A função de ativação *Gaussian Error Linear Unit* (GELU) é amplamente utilizada em redes neurais profundas por combinar propriedades lineares com um comportamento probabilístico suave. A GELU pondera a entrada pela probabilidade de ela ser positiva sob uma distribuição normal padrão, em vez de aplicar um corte rígido como a *ReLU*(LEE, 2023). Sua equação pode ser definida por:

$$\mathrm{GELU}(x) \approx 0.5\,x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}\left(x + 0.044715\,x^3\right)\right)\right). \tag{2.4}$$

Além das funções apresentadas acima, há uma função que também é uma escolha bastante popular em modelos de classificação multi-classe (BANERJEE et al., 2020): a função *Softmax*. Ela é extremamente utilizada, pois transforma um vetor de números reais em um vetor de probabilidades, ou seja, todos os valores ficam entre 0 e 1 e somam exatamente 1. Quanto maior o elemento contido em z, maior a probabilidade associada a ele, mas sempre de forma proporcional aos outros valores do vetor. Sobre seu uso, e muito comum no último estágio de uma rede neural para classificação multi-classe, pois converte as saídas da rede em probabilidades interpretáveis. Ela é definida matematicamente por:

$$sm(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K), \tag{2.5}$$

onde K é o número total de classes e z, um vetor de valores reais.

A escolha da função de ativação determina a saída do neurônio, mas a qualidade dessa saída frente ao objetivo final do modelo só pode ser quantificada por meio da função de custo.

2.3.4 Função de custo

As funções de custo em modelos de redes neurais constituem um componente fundamental para a obtenção de otimização em resultados, uma vez que sua principal função é avaliar o desempenho da arquitetura, ou seja, mensurar a eficiência do modelo em relação à saída esperada (TERVEN1 et al., 2025). O principal objetivo durante o processo de treinamento é a minimização da função de custo, pois essa medida oferece um panorama sobre a diferença entre os valores previstos pelo modelo e os valores reais desejados. Quanto menor o valor retornado pela função de custo, maior é a precisão do modelo em suas predições. Por essa razão, tal função é amplamente utilizada no processo de otimização dos parâmetros da rede, uma vez que, a partir da análise dos resultados obtidos durante o treinamento, é possível realizar ajustes nos pesos e vieses visando à melhoria contínua do desempenho da arquitetura frente à tarefa proposta.

Existem diversas funções de custo, sendo que, em cenários voltados para problemas de regressão — nos quais se busca prever uma variável contínua —, a função mais comumente empregada é a *Mean Square Error* (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{2.6}$$

O erro $(y_i - \hat{y}_i)$ é elevado ao quadrado para evitar sinais negativos e destacar erros maiores, somado para todas as n amostras e dividido por n para obter a média.

Já para problemas de classificação multi-classe, nos quais os rótulos serão categóricos, não formados apenas por uma decisão binária, pode-se utilizar *a Cross-Entropy*, definida por:

$$\text{CE} = -\sum_{i=1}^{n} y_i \log(\hat{y}_i). \tag{2.7}$$

Na Equação (2.7), realiza-se a multiplicação de cada probabilidade real de uma amostra pertencer a uma determinada classe pelo logaritmo da probabilidade prevista. Em essência, essa operação avalia a probabilidade de a previsão estar correta em relação ao rótulo correspondente da amostra. Esta fórmula consiste no termo n representa o número de classes existentes no problema. Quanto a y_i e $\hat{y}_i$, representam a classificação verdadeira e a predição do modelo, respectivamente, sendo que o logaritmo encontrado com $\hat{y}_i$ faz com que amplifique erros que são muito distantes da classe verdadeira.

Além dessas abordagens, existe também o uso da técnica *Class Weight* (GHOSH; BELLINGER; CORIZZO, 2022), na qual a função de custo do modelo é ajustada para atribuir maior importância às classes menos representadas, equilibrando assim o processo de aprendizagem e evitando o favorecimento da classe majoritária.

Portanto, destaca-se a relevância da função de custo em diferentes contextos de aplicação. Cada função de custo possui um propósito específico, sendo utilizada para otimizar e ajustar os parâmetros do modelo em que é aplicada, promovendo o equilíbrio entre os valores obtidos (ELHARROUSSA et al., 2025).

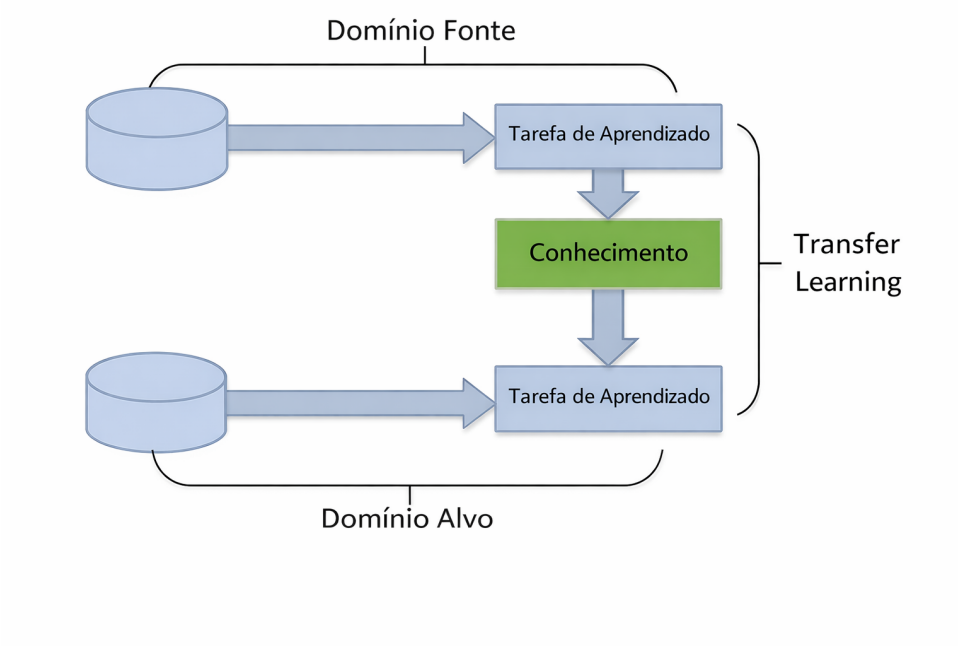
Embora funções de custo otimizadas permitam o treinamento do zero, cenários com escassez de dados ou recursos computacionais exigem estratégias mais eficientes, como o aproveitamento de modelos pré-existentes via *Transfer Learning*.

2.3.5 *Transfer Learning* e *Fine Tuning*

O conceito de *transfer learning* esta atrelado a um aprendizado de alta performance para um determinado domínio, que foi treinado de um outro domínio, mas que de certa maneira se relacionam(WEISS; KHOSHGOFTAAR; WANG, 2016). Basicamente, o processo de *transfer learning* é utilizado para melhorar o treino de um domínio, transferindo informações através de um domínio relacionado (WEISS; KHOSHGOFTAAR; WANG, 2016).

A necessidade de utilização do *transfer learning* deriva da limitação de dados de treino, que pode ocorrer, por exemplo, devido a escassez de dados, que é caro e difícil de ser coletado em grande quantidade, como em situações médicas (KIM et al., 2022).

Figura 2.6 – Fluxograma de um processo de *Tranfer Learning*.



Fonte: Adaptado de (TAN et al., 2018).

O *fine-tuning* é uma técnica de aprendizado por transferência que consiste em adaptar um modelo de aprendizado profundo já treinado em grandes bases de dados para uma nova tarefa específica, utilizando um conjunto menor de dados. Essa abordagem é especialmente útil em problemas como a detecção de pedestres, pois reduz o tempo de treinamento e o custo computacional, ao mesmo tempo em que mantém um bom desempenho (AMISSE; JIJÓN-PALMA; CENTENO, 2021).

Dessa forma, o uso conjunto do *transfer learning* e do *fine-tuning* estabelece um paradigma de eficiência no desenvolvimento de sistemas inteligentes. Ao reaproveitar camadas de extração de características já validadas, é possível superar a barreira da insuficiência de dados e acelerar a convergência dos modelos. Assim, essa metodologia não apenas viabiliza projetos em nichos com poucos recursos informacionais, mas também democratiza o acesso a arquiteturas robustas de aprendizado profundo em aplicações de tempo real e alta precisão.

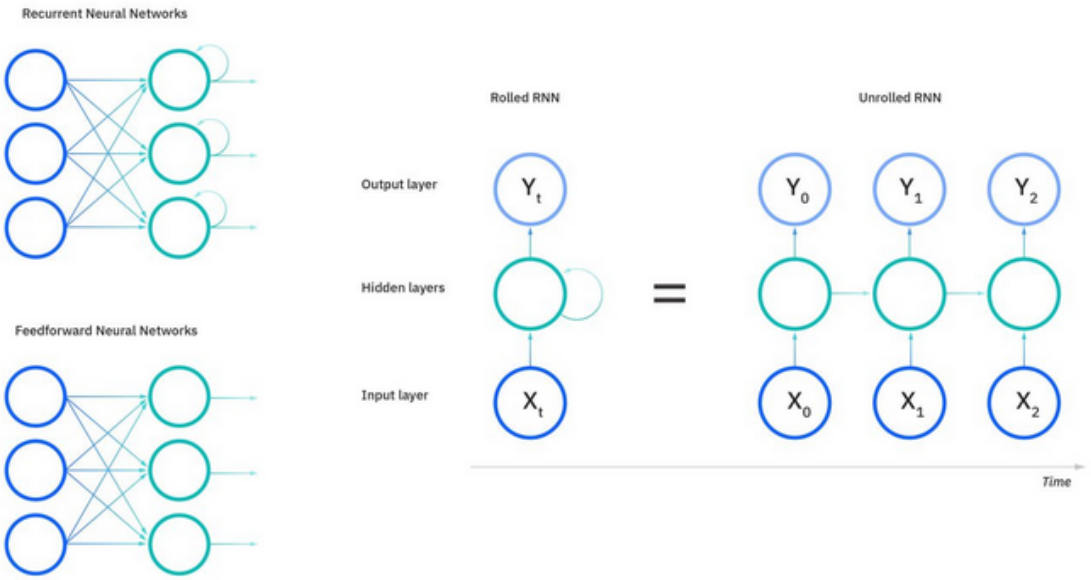
Após a adaptação de modelos através de *fine-tuning*, deve-se considerar que certas tarefas exigem arquiteturas especializadas na natureza dos dados, como no caso de informações sequenciais tratadas pelas *Recurrent Neural Network*.

2.3.6 *Recurrent Neural Network*

As *Recurrent Neural Networks* (RNNs) constituem uma classe de modelos que se destacam pela capacidade de processar dados sequenciais ou séries temporais, ou seja, informações que apresentam interdependência ao longo do tempo. Esse tipo de rede é projetado para lidar com situações em que a ordem e a correlação entre os dados são essenciais para a tarefa de previsão ou classificação (IBM, 2024a).

A principal característica que diferencia uma rede neural recorrente de uma rede neural tradicional reside em sua estrutura interna, a qual incorpora o conceito de memória. Enquanto nas redes neurais tradicionais as entradas e saídas de cada camada são tratadas de forma independente, nas RNNs, cada etapa do processamento é influenciada por informações provenientes das etapas anteriores. Isso ocorre porque, a cada novo elemento da sequência processada, a RNN utiliza não apenas a entrada atual, mas também o estado oculto gerado na etapa anterior, estabelecendo, assim, uma dependência temporal entre os dados (IBM, 2024a).

Figura 2.7 – Exemplo de uma rede neural recorrente em comparação a uma rede neural tradicional



Fonte: IBM (2024a).

Essa capacidade de capturar dependências temporais permite que as RNNs sejam aplicadas em diversas tarefas, como o processamento de linguagem natural, reconhecimento de fala e análise de séries temporais. Além disso, mesmo imagens podem ser utilizadas como entrada, desde que sejam tratadas e convertidas em sequências apropriadas para o modelo (SCHMIDT, 2019).

O treinamento das redes neurais recorrentes requer adaptações específicas na fase de *backpropagation*, devido à sua natureza sequencial. Diferentemente das redes tradicionais, em que o erro é propagado camada por camada, nas RNNs o erro é acumulado ao longo do tempo,

etapa por etapa, no processo conhecido como *Backpropagation Through Time* (BPTT). Essa abordagem permite identificar quais estados ocultos são responsáveis por erros significativos, tornando possível realizar ajustes mais precisos nos pesos da rede (IBM, 2024a).

Apesar de suas vantagens, as RNNs enfrentam um desafio importante durante o treinamento: o problema do desaparecimento ou explosão do gradiente. Esse fenômeno ocorre quando os gradientes calculados durante *backpropagation* tornam-se muito pequenos ou muito grandes, dificultando a atualização eficaz dos pesos e, consequentemente, prejudicando o aprendizado da rede (SCHMIDT, 2019).

Dessa forma, embora as RNNs se mostrem altamente eficazes no processamento de dados sequenciais, é fundamental considerar suas limitações estruturais e os cuidados necessários durante o treinamento para garantir a estabilidade e o desempenho do modelo. Portanto, *Long Short Term Memory Neural Network* e *Extended Long Short Term Memory Neural Network*, tornam-se alternativas de melhorias para tais redes.

2.3.6.1 *Long Short Term Memory Neural Network* e *Extended Long Short Term Memory Neural Network*

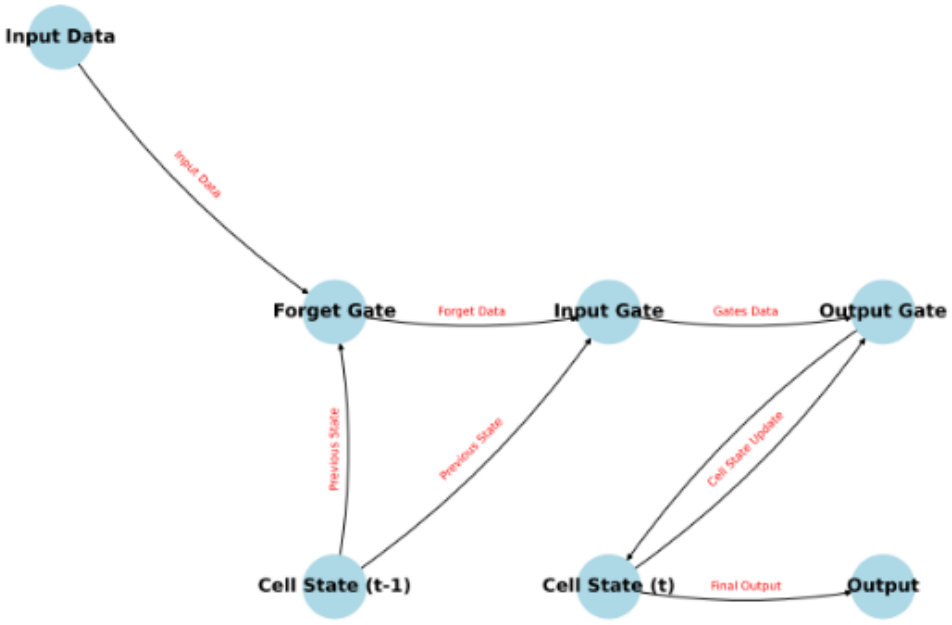
A arquitetura LSTM foi desenvolvida com o propósito de solucionar um problema recorrente nas RNNs, denominado desaparecimento do gradiente (da Silva; MENESES, 2023). As LSTMs distinguem-se por sua capacidade de preservar informações relevantes ao longo do tempo, superando a limitação das RNNs, que armazenam dados apenas em um estado oculto. Para isso, a LSTM introduz uma célula de memória capaz de reter informações por períodos prolongados (da Silva; MENESES, 2023).

Nesse contexto, as células de memória desempenham papel fundamental no controle do fluxo de informações da rede, por meio de um sistema de portões que regula o armazenamento e a exclusão de dados. Essa característica possibilita que o gradiente flua de maneira mais eficiente, com o principal objetivo de mitigar o problema que compromete o desempenho das RNNs (MALASHIN et al., 2024). Estruturalmente, uma rede LSTM é composta por uma unidade de memória e três portões principais: *input*, *output* e *forget* (da Silva; MENESES, 2023).

O portão *forget* é responsável por determinar quais informações devem ser descartadas da célula de memória. Para isso, aplica-se uma função *sigmoid* ao estado oculto anterior e ao estado atual de entrada, produzindo um vetor cujos valores variam entre 0 e 1. Esses valores definem a proporção de informação que será mantida ou eliminada (da Silva; MENESES, 2023).

O portão *input*, por sua vez, regula quais informações serão armazenadas na célula. Inicialmente, é utilizada uma função *sigmoid* para avaliar a relevância da informação, de forma semelhante ao portão anterior. Em seguida, aplica-se uma função *tanh* para gerar um vetor candidato, que será integrado à célula de memória. A combinação dessas operações permite atualizar o estado da célula de forma eficiente (da Silva; MENESES, 2023).

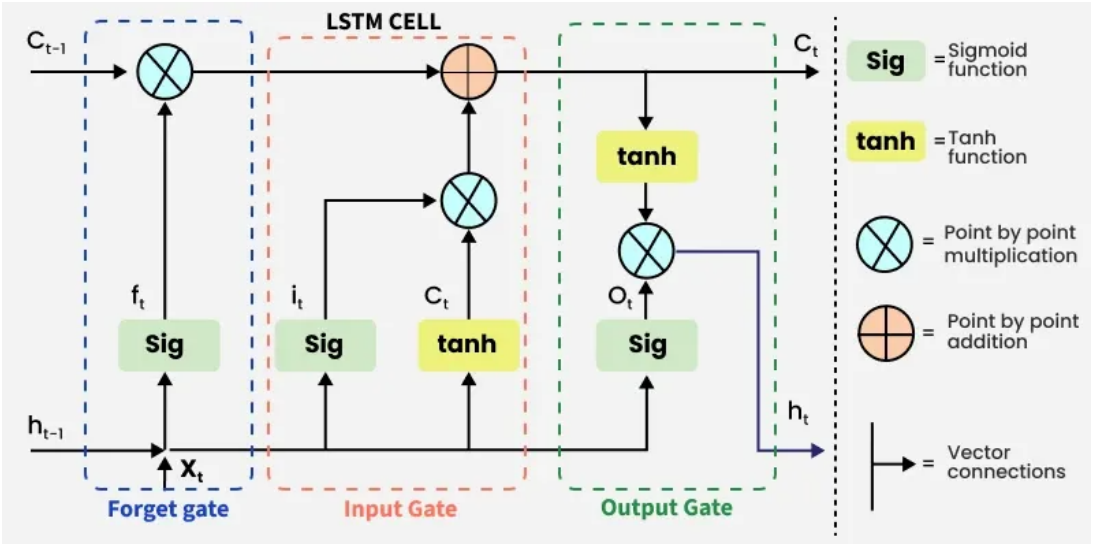
Figura 2.8 – Exemplo de uma rede neural *Long Short Term Memory*



Fonte: Malashin et al. (2024).

Por fim, o portão *output* define quais informações serão utilizadas como saída da rede. Para isso, o estado anterior e atual são processados por uma função *sigmoid*, enquanto a célula de memória passa por uma função *tanh*. A multiplicação dos resultados obtidos determina os valores que comporão a saída da rede (da Silva; MENESES, 2023).

Figura 2.9 – Arquitetura dos portões de uma rede LSTM.



Fonte: GeeksForGeeks (2025).

Em síntese, as redes LSTMs representam um avanço substancial em relação às RNNs convencionais, uma vez que introduzem mecanismos capazes de lidar com dependências de longo prazo, assegurando maior estabilidade no aprendizado e precisão em tarefas que envolvem

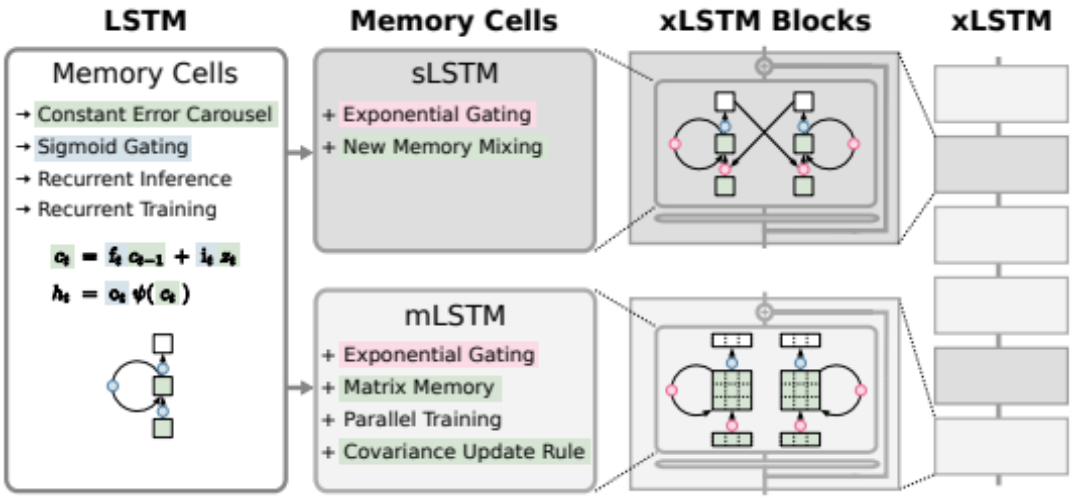
sequências extensas(HOCHREITER; SCHMIDHUBER, 1997).

Não obstante, embora mitiguem os problemas de explosão e desaparecimento do gradiente, as LSTMs apresentam limitações relevantes, tais como: a impossibilidade de revisar decisões anteriores de armazenamento, a restrição na capacidade de memória e a dificuldade de paralelização (BECK et al., 2024). Para superar esses entraves, foi proposta a arquitetura *Extended Long Short Term Memory Neural Network* (xLSTM), que aprimora a estrutura original.

A xLSTM incorpora dois novos módulos: o *sLSTM* e o *mLSTM*. O primeiro introduz uma função de ativação exponencial nos portões *input* e *forget*, conferindo maior estabilidade à rede em cenários de alto fluxo de informações. Além disso, adiciona um estado de normalização obtido pela soma do produto entre o portão *input* e os portões *forget* futuros (BECK et al., 2024). Outra característica importante do sLSTM é o suporte a múltiplas células de memória, o que, aliado ao uso de ativações exponenciais, contribui para a robustez da arquitetura.

O *mLSTM*, por sua vez, amplia a capacidade de armazenamento mediante a utilização de células de memória matriciais. Nesse caso, o normalizador é definido como a soma ponderada dos valores da matriz, determinada pelos portões *input* e *forget* futuros. Essa abordagem mantém um registro detalhado da intensidade dos vetores, indicando se foram atenuados ou preservados ao longo do processamento (BECK et al., 2024).

Figura 2.10 – Arquitetura xLSTM.



Fonte: Beck et al. (2024).

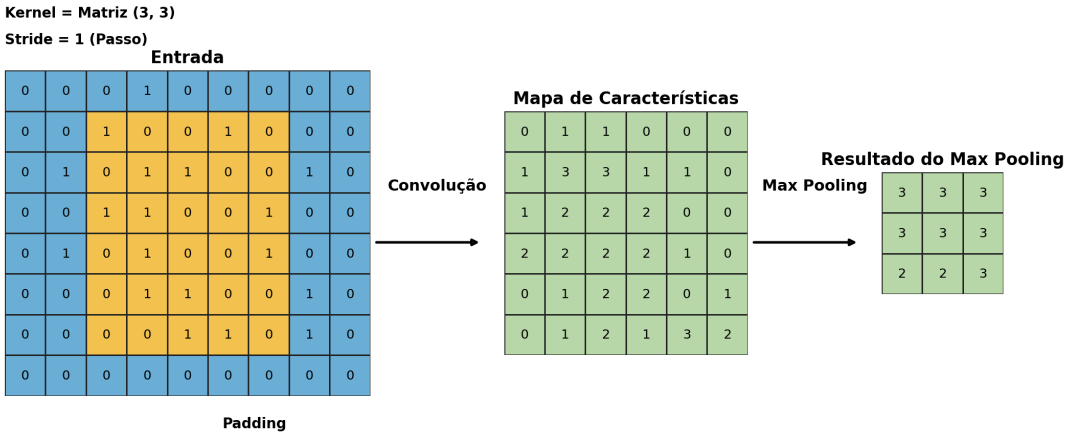
Enquanto as arquiteturas recorrentes focam na persistência temporal, outra classe de modelos destaca-se pela extração de características espaciais e padrões locais: *Convolutional Neural Networks*.

2.3.7 *Convolutional Neural Networks*

As *Convolutional Neural Networks* (CNNs) consolidaram-se como uma das arquiteturas mais relevantes no âmbito do *Deep Learning* (LI et al., 2021). Essas redes baseiam-se em três

princípios fundamentais que asseguram sua eficiência: o uso de conexões locais, o compartilhamento de pesos e a redução de dimensionalidade por meio de camadas de *pooling*. Tais características possibilitam uma diminuição significativa do número de parâmetros do modelo, contribuindo para uma convergência mais rápida durante o treinamento e para a mitigação do problema de *overfitting*. Do ponto de vista estrutural, as CNNs operam a partir da aplicação de *kernels* convolucionais, responsáveis pela extração de mapas de características, fazendo uso de técnicas como *padding*, para o ajuste das dimensões da entrada, e *stride*, para o controle do deslocamento e da densidade das operações de convolução (LI et al., 2021).

Figura 2.11 – Figura que demonstra o processo de uma convolução e uma aplicação de *pooling* em uma CNN



Fonte: Próprio Autor.

Na camada convolucional, *kernels* ou filtros aprendíveis deslizam sobre os dados de entrada para gerar mapas de ativação que identificam características específicas, sendo controlados por hiperparâmetros como a profundidade do volume de saída, o *stride* (passo do deslocamento) e o *padding*, que ajusta as bordas para manter o controle da dimensionalidade. A eficiência desse processo é ampliada pelo compartilhamento de parâmetros, onde o mesmo filtro é aplicado em diferentes regiões sob a premissa de que uma característica útil em um ponto da imagem provavelmente será relevante em outro. Após a extração, camadas de pooling, como o *max-pooling*, reduzem a ativação e a contagem de parâmetros enquanto preservam as informações essenciais(O'SHEA; NASH, 2015).

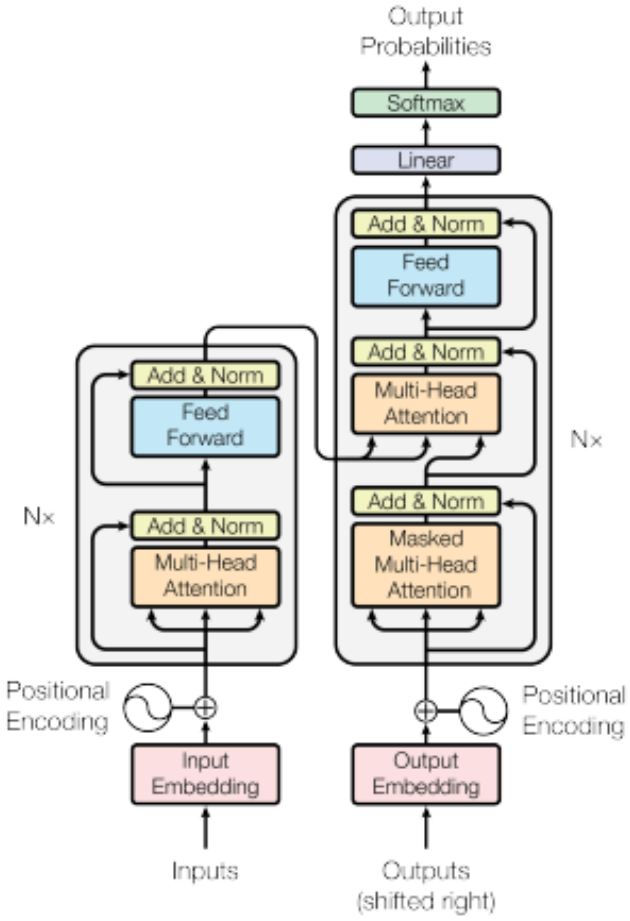
Já para funções de ativação não lineares, sendo a *ReLU* a mais proeminente por mitigar problemas de desaparecimento do gradiente, garantem que a rede consiga aprender padrões complexos e profundos, com o surgimento da *AlexNet*(LI et al., 2021).

2.3.8 Transformers

A arquitetura de redes neurais denominada *Transformer* destaca-se por empregar mecanismos de autoatenção capazes de identificar relações entre diferentes partes da entrada fornecida ao modelo (IBM, 2025). Tais mecanismos de atenção permitem capturar dependências entre palavras,

independentemente da distância entre elas, o que contrasta com as RNNs, cujos estados dependem diretamente da saída do estado anterior. Essa dependência sequencial dificulta o paralelismo no processamento de sequências. Em contrapartida, a arquitetura *Transformer*, fundamentada em mecanismos de atenção, viabiliza tanto a modelagem de dependências globais entre entradas e saídas quanto a execução paralela de operações, otimizando a eficiência computacional do treinamento (VASWANI et al., 2023).

Figura 2.12 – Arquitetura *Transformers*.



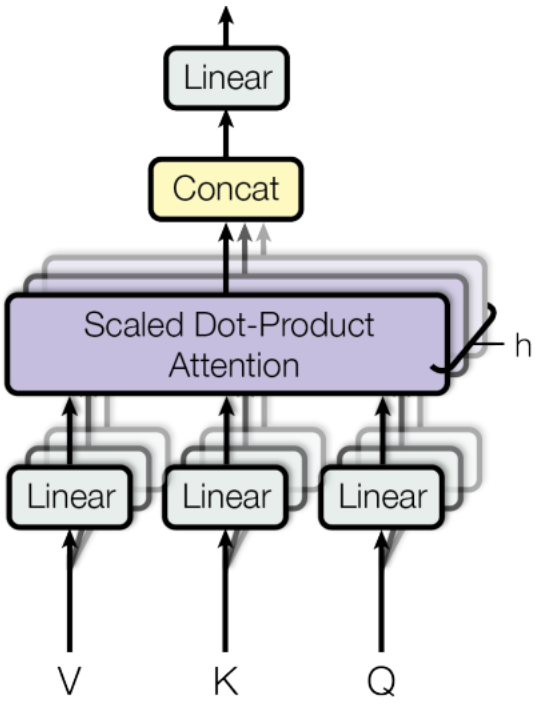
Fonte: Vaswani et al. (2023).

Essa arquitetura inicia seu processamento transformando os dados de entrada em unidades denominadas *tokens*, aos quais são atribuídos identificadores numéricos. Esses identificadores permitem que o modelo navegue pelos vetores representacionais gerados com o propósito de reter informações semânticas e contextuais sobre cada *token*, além de calcular relações de influência entre eles por meio dos mecanismos de atenção (IBM, 2025). Como a arquitetura não possui componentes recorrentes, como nas RNNs, faz-se necessário incorporar informações posicionais para que o modelo compreenda a ordem sequencial dos *tokens* (VASWANI et al., 2023). Isso é realizado por meio de vetores posicionais, os quais indicam as posições relativas e absolutas dos elementos na sequência, permitindo ao modelo atribuir maior atenção a elementos próximos

(IBM, 2025). Em seguida, cada vetor de entrada é projetado linearmente em três representações distintas, consulta, chave e valor, por meio de multiplicações com matrizes de pesos aprendíveis durante o pré-treinamento do modelo (IBM, 2025).

Após essas etapas, o processamento avança para as camadas de autoatenção, consideradas um dos principais avanços em arquiteturas de modelos para tarefas sequenciais (XIAO; ZHU, 2023). A autoatenção é um mecanismo que permite ao modelo concentrar-se em partes específicas da entrada com base no contexto atual do processamento. Isso é feito comparando vetores de consulta com vetores de chave e utilizando essas comparações para ponderar os vetores de valor na formação da saída (VASWANI et al., 2023). Nesse contexto, a técnica de *Multi-Head Attention* se destaca: em vez de aplicar uma única autoatenção, realiza-se múltiplas atenções paralelas em diferentes projeções lineares dos vetores, permitindo ao modelo capturar múltiplas relações contextuais simultaneamente (VASWANI et al., 2023). Ao final, os resultados dessas múltiplas cabeças de atenção são concatenados antes de serem processados pelas próximas camadas.

Figura 2.13 – Estrutura *Multi-Head Attention*.



Fonte: Vaswani et al. (2023).

Após o processamento pelas camadas de *Multi-Head Attention* e pelas camadas subsequentes compostas por redes neurais totalmente conectadas, aplica-se a normalização das ativações. Esse processo, conhecido como *layer normalization*, tem como objetivo padronizar as saídas intermediárias, promovendo maior estabilidade no processo de treinamento e melhorando a convergência do modelo (VASWANI et al., 2023).

Essa mudança de paradigma possibilitada pelos *Transformers* serviu de alicerce para o desenvolvimento de modelos de larga escala, como o BERT e as modernas LLMs, que redefiniram

o estado da arte em compreensão textual.

2.3.8.1 *Bidirectional Encoder Representations from Transformers*

A arquitetura *Transformer* constitui a base de diversos modelos de linguagem de grande escala, entre os quais se destaca o *Bidirectional Encoder Representations from Transformers* (BERT) (KOROTEEV, 2021). O BERT foi projetado com ênfase em tarefas de compreensão textual, fazendo uso exclusivo do componente codificador da arquitetura *Transformer*. Seu principal diferencial reside na adoção do mecanismo de atenção bidirecional, que possibilita ao modelo aprender representações contextuais considerando simultaneamente o contexto à esquerda e à direita de cada palavra. Essa característica permitiu avanços significativos em tarefas como classificação de sentenças, resposta automática a perguntas e análise de sentimentos. Ao empregar atenção bidirecional, o BERT superou abordagens baseadas em representações estáticas e independentes de contexto, como *Word2Vec*(MA; ZHANG, 2015) e *GloVe*(DEY; DESAI, 2022), ao introduzir *embeddings* contextuais, que são representações numéricas do texto capazes de capturar o significado específico das palavras de acordo com o contexto em que são utilizadas (GARDAZI et al., 2025).

Do ponto de vista estrutural, o BERT é composto por múltiplas camadas de codificadores *Transformer*, que utilizam mecanismos de autoatenção para o processamento eficiente de sequências de entrada. Com o objetivo de contornar limitações relacionadas ao processamento de textos extensos, pesquisas posteriores propuseram estratégias como a segmentação das entradas e a adição de camadas, ampliando as capacidades originais do modelo. Ademais, estudos sobre seus estados ocultos indicam que o processamento interno do BERT apresenta correspondência com as etapas tradicionais de um pipeline de PLN, o que reforça sua eficácia na captura de relações contextuais profundas e semanticamente relevantes (GARDAZI et al., 2025).

2.3.8.2 *Large Language Model*

O progresso acelerado dos modelos de *Large Language Models* (LLMs) transformou o campo de PLN (CHEN; VAROQUAUX, 2024). Tais modelos demonstram uma excelente performance em diversas tarefas, mas isso normalmente implica um enorme custo computacional e consumo de energia, uma vez que esses modelos possuem uma grande quantidade de parâmetros (CHEN; VAROQUAUX, 2024). Esses modelos são principalmente baseados em redes com a arquitetura *Transformer* (VASWANI et al., 2023), como é o caso do modelo *Llama*(TOUVRON et al., 2023), já que este tipo de modelo é treinado com uma quantidade enorme de dados (MINAEE et al., 2024).

O conjunto de modelos da família *Llama* é composto por modelos de código aberto disponibilizados pela empresa *Meta* (MINAEE et al., 2024). Apesar da utilização da arquitetura de *Transformers*, esse modelo apresenta algumas modificações que diferem da arquitetura comumente observada. Essas alterações incluem a normalização de cada entrada da subcamada

do *Transformer* para a estabilidade do treinamento (TOUVRON et al., 2023), a substituição da função de ativação *ReLU* (HE et al., 2020) por *SwiGLU* (AWASTHI et al., 2025) e, por fim, a substituição dos vetores posicionais da arquitetura tradicional por *Rotary Position Embedding* (RoPE) (SU et al., 2024), que é uma técnica avançada para codificar a posição de *tokens* em uma sequência.

Embora haja essas mudanças no modelo *Llama* (TOUVRON et al., 2023), a construção de um LLM eficaz envolve um pipeline rigoroso, que começa com a preparação de dados, incluindo filtragem de qualidade, remoção de ruído e deduplicação, processos essenciais para a capacidade de generalização do modelo (MINAEE et al., 2024). Além disso, ter um processo de tokenização que possa suportar o grande vocabulário presente em modelos desta magnitude torna-se importante para lidar com palavras em diferentes línguas e também palavras que não são vistas nos dados de treinamento. Portanto, métodos como *BytePairEncoding* (KOZMA; VODERHOLZER, 2024), que é um algoritmo de tokenização e compressão que transforma textos em subpalavras menores e mais frequentes, combinando iterativamente os pares de caracteres mais comuns, baseados em sub-palavras, podem ser combinados para gerar um grande numero de palavras e, assim, aumentar o repertório de vocabulário do modelo.

2.3.8.3 *Pooling*

Tendo em vista as representações de *tokens* em modelos baseados em arquitetura *Transformers*, camadas de *pooling* tem um papel extremamente importante em termos de agregação no processo de linguagem natural, já que mecanismos de atenção geram representações em termos de *tokens* e tais itens devem ser agregados para formar sentenças. O papel consiste em condensar tais *tokens*, de maneira que formem sentenças unificadas(XING et al., 2024).

Tendo em vista a existência de diversas técnicas que envolvem *pooling*, de acordo com (XING et al., 2024) pode-se citar:

- *Mean Pooling*: Esta técnica fornece uma representação balanceada dos *tokens*, utilizando a média dos valores, tendo assim uma perspectiva uniforme sobre todos os *tokens*.
- *Max Pooling*: Esta técnica consiste em um foco nas características mais representativas na sequência dos *tokens*, selecionando os valores máximos sobre todas as dimensões dos *tokens*, dando ênfase em características distintas.

Embora as técnicas de *pooling* permitam uma condensação eficiente das representações textuais, a adaptação de modelos de larga escala para tarefas específicas exige eficiência não apenas na representação dos dados, mas também no ajuste dos parâmetros, o que justifica o emprego de métodos como o LoRA.

2.3.8.4 *Low-Rank Adaptation* (LoRA)

É notório o rápido avanço de modelos LLM, no qual traz desafios em relação a adaptações de tais arquiteturas, como custo computacional e a eficiência de parâmetros(LIANG et al., 2025). Tal adaptação demanda recursos e com isso, técnicas envolvendo o uso de LoRA, buscam reduzir o número de parâmetros treináveis, sem sacrificar a performance do modelo(LIANG et al., 2025). Ao reduzir a quantidade de parâmetros no processo de ajuste do modelo, resulta-se em um significante incremento da eficiência computacional e de armazenamento(HU et al., 2022).

O processo de tal técnica consiste em aplicar estratégias de inicialização nas matrizes de atualização de pesos do modelo para fornecer estabilidade e eficiência para o treinamento. Este método de redução de parâmetros corresponde ao uso de matrizes de baixo *rank*, ou seja, diminui a espessura das matrizes do modelo original(YANG et al., 2024) enquanto mantém as matrizes originais congeladas.

Tal aplicação de método, pode ser adaptada a partir da definição dos seguintes parâmetros(YANG et al., 2024):

- Rank: O valor deste parâmetro controla diretamente o número de parâmetros treináveis do modelo que precisam ser aprendidos durante o processo de ajuste do modelo.
- Alpha: É um fator escalar, que juntamente com o parâmetro *rank*, controla a magnitude de atualização da matriz adaptativa final. Ou seja, influencia diretamente nas matrizes alteradas para funcionarem de maneira mais leve com o modelo.

À vista disso, este recurso constitui vantagens importantes para aplicação em modelos de larga escala com ajuste, de acordo com (YANG et al., 2024), como:

- Eficiência de parâmetros e treinamento: Dispõe um número mínimo de parâmetros treináveis, focando nas matrizes com a definição de baixo *rank*. Enquanto em métodos convencionais, que haveria a atualização de todos os parâmetros, o método foca nas matrizes de baixo *rank*.
- Não introduz latência extra na inferência: Isso ocorre devido a fácil incorporação, que pode ocorrer, da matriz de atualização as matrizes originais do modelo que foram congeladas, mantendo a eficiência operacional.
- Retenção de conhecimento: LoRA ao preservar os pesos do modelo pré-treinado, evita que tenha uma perda de conhecimento ao modelo. Portanto, a técnica mantém o conhecimento base do modelo, a medida que consegue ajustar a nova situação.

Em suma, percebe-se o benefício e a adaptabilidade do método aos modelos LLM, podendo ser uma alternativa para diminuição de custo computacional e adaptação a uma tarefa, permitindo uma grande flexibilidade de uso.

2.4 Balanceamento de dados

O balanceamento de dados configura-se como uma etapa essencial no pré-processamento de conjuntos de dados, especialmente quando há um desbalanceamento significativo entre as classes. Essa etapa é de suma importância, uma vez que pode contribuir de maneira significativa para a melhoria do desempenho dos modelos de aprendizado de máquina, elevando métricas como acurácia e reduzindo o viés que o modelo pode apresentar em relação à classe majoritária (MDUMA, 2023). Para ilustrar a relevância dessa etapa, considere um cenário em que um conjunto de dados possui 1.000 amostras, das quais 999 estão rotuladas com o sentimento "feliz" e apenas uma com o sentimento "triste". Ao treinar um modelo com esse conjunto de dados, é provável que ele apresente uma elevada taxa de acerto. Entretanto, essa performance ocorre em decorrência de um viés acentuado em favor da classe majoritária. Isso significa que, ao ser aplicado em um contexto real, no qual a distribuição de sentimentos pode ser mais equilibrada ou até inversa, o modelo terá dificuldades em identificar corretamente exemplos da classe minoritária, apresentando um desempenho insatisfatório.

Com o objetivo de mitigar esse tipo de viés, diversas técnicas de balanceamento são utilizadas. Dentre elas, destaca-se o método *Synthetic Minority Over-sampling Technique* (SMOTE) (PRADIPTA et al., 2021), que consiste na geração de novas amostras sintéticas da classe minoritária por meio da interpolação entre instâncias existentes dessa classe. Outra técnica amplamente utilizada é o *Random OverSampling* (WONGVORACHAN; HE; BULUT, 2023), que realiza a duplicação aleatória de exemplos da classe minoritária no conjunto de dados.

Além disso, o uso de *Easy Data Augmentation* (EDA) (WEI; ZOU, 2019) é outro exemplo que consiste em uma técnica de quatro modificações que podem ser realizadas: a substituição de palavras por sinônimos, a inserção aleatória de uma palavra, a troca aleatória de uma palavra do texto e a remoção de uma palavra aleatória. Por fim, outra estratégia é a de utilizar *Generative Artificial Intelligence* (GenAI) para gerar dados sintéticos que compõem o conjunto de dados e podem oferecer a generalização necessária, além de balancear os dados de maneira eficiente.

Por fim, após garantir que o modelo foi treinado em um conjunto de dados equilibrado e com a arquitetura adequada, é valido utilizar métricas de avaliação rigorosas para validar sua real eficácia e capacidade preditiva.

2.5 Métricas de avaliação

O uso de métricas de avaliação é uma alternativa no contexto de experimentações em aprendizado de máquina, uma vez que essas métricas permitem mensurar com precisão o desempenho de um modelo diante de uma tarefa específica. Por meio dessas ferramentas de qualificação, é possível compreender o quão eficaz um modelo é ao realizar previsões, possibilitando ajustes e comparações entre diferentes abordagens(OROZCO-ARIAS et al., 2020).

Considerando essa importância, (VUJOVIć, 2021) apresenta algumas métricas essenciais para a avaliação de modelos de classificação, como descrito a seguir:

- *Verdadeiro Positivo* (VP): número de amostras positivas corretamente classificadas.
- *Verdadeiro Negativo* (VN): número de amostras negativas corretamente classificadas.
- *Falso Positivo* (FP): número de amostras negativas incorretamente classificadas como positivas.
- *Falso Negativo* (FN): número de amostras positivas incorretamente classificadas como negativas

- Número de classificações corretas: Representa o total de instâncias corretamente classificadas, ou seja, a soma dos verdadeiros positivos e verdadeiros negativos. Ela é definida por:

$$NCC = VP + VN. \tag{2.8}$$

- Número de classificações incorretas: Corresponde à soma das classificações incorretas, englobando tanto os falsos positivos (*False Positives* - FP) quanto os falsos negativos (*False Negatives* - FN). Ela é definida por:

$$NCI = FP + FN. \tag{2.9}$$

- Acurácia: Mede a proporção de classificações corretas (TP + TN) em relação ao total de instâncias analisadas, sendo uma métrica amplamente utilizada para indicar o desempenho geral de um modelo. Ela é definida matematicamente por:

$$ACC = \frac{VP + VN}{VP + VN + FP + FN}. \tag{2.10}$$

Além dessas métricas básicas, (CHRISTEN; HAND; KIRIELLE, 2023) destaca outras medidas que fornecem uma visão mais aprofundada sobre o comportamento do modelo de classificação:

- *Precision*: Indica a proporção de classificações positivas que foram corretamente previstas. Ou seja, dentre todas as previsões positivas feitas pelo modelo, quantas de fato pertenciam à classe positiva. Essa métrica é especialmente útil quando o custo de falsos positivos é elevado. Ela é definida por:

$$PRE = \frac{VP}{VP + FP}. \tag{2.11}$$

- *Recall*: Mede a capacidade do modelo em identificar corretamente as instâncias positivas. Em outras palavras, dentre todas as instâncias realmente positivas, quantas foram corretamente identificadas como tal. Esta métrica é relevante em contextos onde os falsos negativos são mais críticos. Ela é definida por:

$$REC = \frac{VP}{VP + FN}. \tag{2.12}$$

- *F1-score*: Corresponde à média harmônica entre *Precision* e *Recall*, proporcionando um equilíbrio entre essas duas métricas. É especialmente útil quando se busca um compromisso entre minimizar falsos positivos e falsos negativos. Ela é definida por:

$$F1 = 2 * \frac{PRE * REC}{PRE + REC}. \tag{2.13}$$

3 Trabalhos Relacionados

Atualmente, observa-se que as redes sociais vêm sendo amplamente utilizadas como meio de expressão de emoções, sejam elas positivas ou negativas. No contexto da análise de sentimentos, essas plataformas se configuram como uma fonte extremamente ampla e útil de dados. Diversos pesquisadores têm explorado as redes sociais como campo de investigação, considerando que esses ambientes favorecem a livre manifestação de sentimentos por parte dos usuários (VILLAVICENCIO et al., 2021). Ainda que entrevistas e conversas transcritas também possam ser utilizadas como corpus para esse tipo de estudo, as redes sociais apresentam vantagens práticas e metodológicas relevantes. Com isso, a área científica de análise de sentimentos expande-se não apenas dentro da Psicologia, mas também para o campo de PLN. A partir de postagens em mídias sociais e de transcrições, é possível aplicar técnicas de PLN com o intuito de realizar classificações automáticas, possibilitando a compreensão das emoções e opiniões manifestadas diante de determinadas situações, contribuindo, assim, para tomadas de decisão mais embasadas (SONAWANE; SHINDE, 2025).

Na literatura recente, é possível identificar uma diversidade de trabalhos que abordam a temática da análise de sentimentos utilizando modelos de aprendizado de máquina aplicados ao PLN. Um desses estudos é o de Edara et al. (2023), que propôs um modelo baseado em LSTM para a classificação de sentimentos de pacientes com câncer. O autor realizou a coleta de dados em mídias sociais, como o *Twitter*, utilizando a API oficial da plataforma para dois dos três conjuntos de dados utilizados na pesquisa. Tal abordagem evidencia a facilidade de utilizar as redes sociais como fonte para análise de sentimentos, uma vez que os usuários frequentemente expressam suas emoções de forma espontânea (EDARA et al., 2023). O modelo alcançou uma acurácia de 97,84% no primeiro conjunto de dados, composto por *tweets*, utilizando a plataforma *Apache Spark* para processar o grande volume de dados, e a biblioteca *ML Lib Spark* para aplicar o modelo de aprendizado. Ainda que os outros conjuntos tenham apresentado resultados inferiores, com acurácias de 88,37% e 84,1%, os valores obtidos indicam um potencial promissor de melhorias em futuras pesquisas.

Outro estudo que também utilizou mídias sociais como fonte de dados foi desenvolvido por Balakrishnan, Idicula e Jones (2021), o qual analisou uma comunidade de saúde *online* dedicada à partilha de relatos de pacientes oncológicos. Este ambiente foi explorado como objeto de estudo para a tarefa de classificação de sentimentos. A autora empregou duas técnicas distintas de representação de palavras e aplicou um modelo *Bidirectional Long Short Term Memory Neural Network* (BiLSTM), que, diferentemente do modelo proposto por (EDARA et al., 2023), combina duas redes LSTM. O modelo BiLSTM foi comparado com outras arquiteturas, incluindo uma *Convolutional Neural Networks* (CNN) e uma LSTM convencional. A abordagem BiLSTM destacou-se, obtendo um *F1-Score* de 91,9% utilizando a técnica de *Sentiment Embedding*,

a qual visa capturar não apenas a semântica das palavras, mas também o sentimento nelas contido. Com a técnica de *Word Embedding*, os resultados foram de 90,3%. Os modelos CNN e LSTM, por sua vez, alcançaram 87,7% e 85,9%, bem como 90,2% e 89,8%, respectivamente. Os resultados demonstram que diferentes abordagens técnicas e fontes de dados, como comunidades de pacientes, podem contribuir significativamente para a eficácia da classificação de sentimentos (BALAKRISHNAN; IDICULA; JONES, 2021).

Seguindo essa linha de investigação, Chatzimina et al. (2024) apresentaram um estudo comparativo entre os modelos BERT, RoBERTa, GPT-2 e XLNet, aplicados à análise de sentimentos em transcrições de interações clínicas entre pacientes oncológicos e profissionais de saúde. A base de dados foi composta por 185 horas de gravações de discussões entre médicos e pacientes, com os enunciados rotulados em categorias positivas, negativas e neutras. Os dados foram pré-processados e divididos em 80% para treinamento, 10% para teste e 10% para validação. Durante o treinamento, foi utilizada a técnica de *Early Stopping* para evitar o *overfitting*, ou seja, a perda da capacidade de generalização do modelo. As métricas adotadas para avaliação incluíram acurácia, *F1-Score*, *Precision*, *Recall* e especificidade. O modelo BERT apresentou os melhores resultados, com uma acurácia de 95,48% e um *F1-Score* de 95,47%. O modelo RoBERTa, por sua vez, apresentou desempenho ligeiramente inferior, com 91,43% de acurácia e 91,30% de *F1-Score*. Já os modelos GPT-2 e XLNet apresentaram acurácias de 78,72% e 74,90%, e *F1-Scores* de 66,10% e 86,15%, respectivamente. Os resultados indicam a superioridade dos modelos BERT e RoBERTa na análise de sentimentos em contextos clínicos complexos.

Com uma proposta aplicada ao âmbito comercial, (MIRDAN; BUYRUKOğLU; BAKER, 2025) desenvolveu um modelo híbrido composto por BiLSTM, CNN e mecanismos de atenção, voltado à análise de sentimentos em avaliações de produtos. A coleta de dados foi realizada a partir de avaliações publicadas no *Twitter*, por meio de sua API, tal como em (EDARA et al., 2023). No entanto, tal procedimento pode gerar um problema recorrente: o desbalanceamento de classes, um dos principais desafios enfrentados na área de Inteligência Artificial. Isso ocorre quando há predominância de dados em uma classe específica, gerando viés no modelo. Para mitigar esse efeito, foi aplicada a técnica *Synthetic Minority Over-sampling Technique* (SMOTE), a qual cria amostras sintéticas para a classe minoritária. Com essa abordagem, o modelo híbrido alcançou 96% de acurácia e um *F1-Score* de 94,5%, demonstrando um desempenho elevado. De forma semelhante, (SINGH; BARVE; DWIVEDI, 2025) propôs um modelo híbrido voltado ao domínio da saúde, utilizando BiLSTM e *Gated Recurrent Unit* (GRU), combinados com *Embeddings* do modelo BERT enriquecidos com *Named Entity Recognition* (NER), técnica que visa extrair entidades nomeadas associadas à saúde. Também foi incorporado um mecanismo de atenção para identificar as palavras mais relevantes em cada texto, com o objetivo de realizar uma *Aspect-Based Sentiment Analysis* (ABSA), abordagem que permite a análise de sentimentos relacionados a aspectos específicos do texto. A base de dados utilizada foi o *Yelp Dataset*, e foram selecionadas avaliações relacionadas ao contexto hospitalar. O modelo proposto superou variações da própria arquitetura, atingindo uma acurácia de 82%.

Com o intuito de lidar com diferentes domínios e vocabulários técnicos variados, (YUE; LI, 2025) propôs um modelo baseado em *Multi Task Learning* (MTL), técnica que busca lidar simultaneamente com múltiplas tarefas. A proposta baseou-se no *Domain Attention Model* (DAM) (YUAN et al., 2018), substituindo as redes LSTM por xLSTM. Os experimentos foram realizados com três conjuntos de dados distintos, representando diferentes domínios. O modelo central, denominado *Dynamic Domain Information Modulation Algorithm* (DAMA), é responsável por identificar o domínio de origem do dado de entrada, etapa crucial para a acurácia final na classificação do sentimento. Em síntese, o modelo proposto superou os demais analisados na maioria dos domínios, destacando-se no domínio de Saúde, com acurácia de 91,5%, e no domínio de Revistas, com 90,8%, embora a etapa de identificação de domínio ainda represente um ponto de melhoria (YUE; LI, 2025).

Dessa forma, observa-se que este trabalho guarda similaridades com os estudos discutidos, ao empregar técnicas baseadas em *Deep Learning* e explorar fontes de dados provenientes de redes sociais, avaliações e interações clínicas. Apesar dos resultados promissores apresentados, os modelos analisados ainda possuem margem para aprimoramentos, os quais podem contribuir significativamente para a tomada de decisão no contexto da saúde, especialmente no que diz respeito à compreensão dos sentimentos expressos por pacientes.

4 Metodologia proposta

Neste capítulo, é apresentado o conjunto de dados utilizado neste trabalho na seção 4.1, enquanto a metodologia utilizada para treinar e processar os dados é apresentada na seção 4.2.

4.1 Conjunto de dados

Como mencionado anteriormente, o *dataset*[1] de ORCHI et al. corresponde a um conjunto de dados disponibilizado na plataforma *Kaggle*, configurando-se como um recurso relevante para a análise qualitativa das experiências relacionadas à saúde mental de pacientes oncológicos e seus respectivos cuidadores. Esse *dataset* foi criteriosamente criado com o propósito de possibilitar investigações mais aprofundadas acerca dos aspectos emocionais envolvidos na trajetória do câncer, por meio da aplicação de técnicas de análise de linguagem natural.

A estrutura do conjunto de dados contempla um total de 10.087 postagens textuais, extraídas de diversas plataformas online voltadas ao suporte e à discussão entre indivíduos afetados direta ou indiretamente pela doença. Essas plataformas funcionam como espaços de compartilhamento de vivências e relatos, tanto por parte dos pacientes quanto dos cuidadores. As postagens estão organizadas de acordo com cinco tipos específicos de câncer — cerebral, de cólon, hepático, leucemia e pulmonar — permitindo, assim, uma análise comparativa e segmentada das nuances emocionais associadas a cada diagnóstico.

No que se refere à anotação emocional, cada postagem foi atribuída a uma pontuação dentro de uma escala que varia de -2 a 1. Esta escala busca quantificar o conteúdo emocional expresso, sendo que os valores negativos (-1 e -2) denotam a presença de emoções como sofrimento, luto ou angústia; o valor positivo (1) representa manifestações de emoções positivas, como felicidade, alívio ou sensação de superação; e o valor neutro (0) refere-se a postagens nas quais não se identificam expressões emocionais evidentes. Como pode-se observar na Tabela 4.1:

A seguir, apresenta-se o gráfico com a proporção de cada classe emocional identificada no conjunto de dados:

4.2 Metodologia proposta

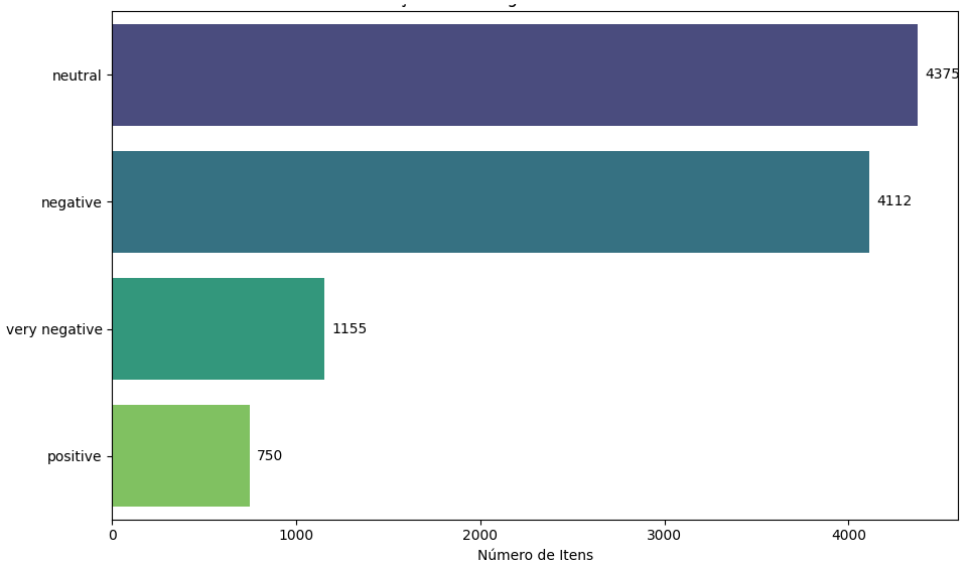
Esta seção descreve a metodologia adotada para o desenvolvimento deste trabalho. As etapas que compõem o processo metodológico incluem: o pré-processamento dos dados, a divisão

[1] Mental Health Insights: Vulnerable Cancer Patients. Disponível em: <https://www.kaggle.com/datasets/irinhoque/mental-health-insights-vulnerable-cancer-patients>. Acessado em 3 de março de 2026.

Texto	Classe	Intensidade
I know as parent of child with down syndrome that you have all hear that our child are at a high risk...	negative	-1
but in my heart I know this is the future promise article regardless http ottawa ctvnew can ottawa r...	neutral	0
I have mylefibrosis which turn to leukemia they want to do a stem cell transplant stc on I but want...	negative	-1
from one of my health group subject wayne dyer leukemia in case anyone here is not aware wayne have...	neutral	0

Tabela 4.1 – Exemplos de dados presente no *dataset* utilizado

Figura 4.1 – Quantidade de textos presentes em cada classe do conjunto de dados.



Fonte: Próprio Autor

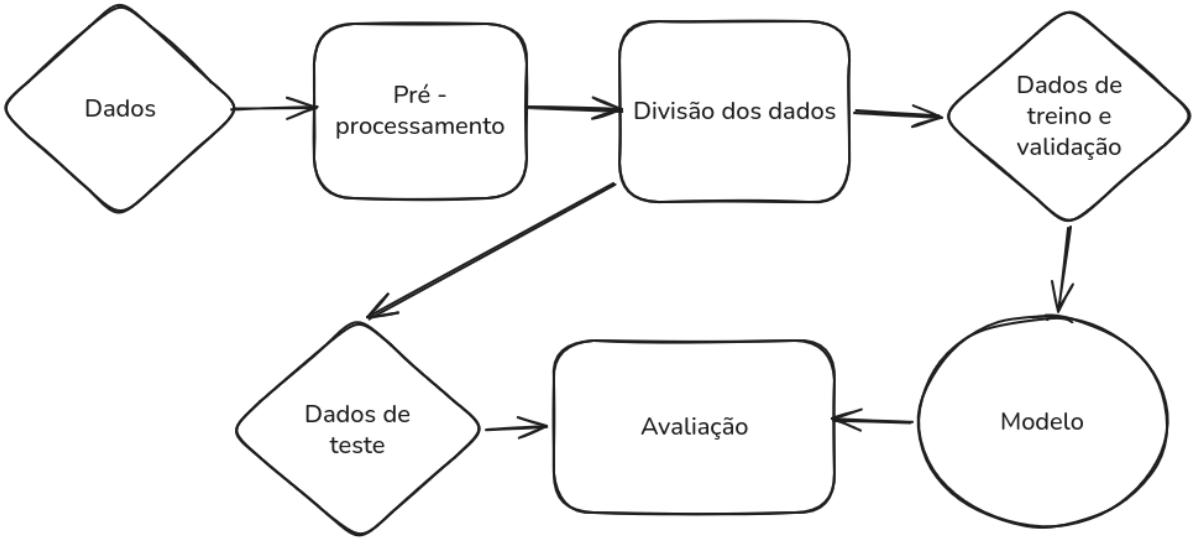
do conjunto em subconjuntos de treino, validação e teste, o treinamento dos modelos e, por fim, a avaliação do desempenho dos mesmos.

Inicialmente, realiza-se a extração das informações contidas no conjunto de dados (ORCHI et al., 2023) e a separação do conjunto de dados em três subconjuntos: treino, validação e teste. Em seguida, aplica-se o processo de pré-processamento, que consiste em duas etapas principais: o balanceamento das classes e a limpeza textual. Para mitigar o desbalanceamento das classes, adotaram-se três técnicas: *Easy Data Augmentation* (EDA), com o intuito de utilizar a estratégia de troca de palavras aleatoriamente, para geração de dados sintéticos nas duas classes minoritárias(positiva e muito negativa). Para utilizar um método de parafraseamento nas frases dessas classes, utilizou-se *Generative Artificial Intelligence* (GenAI) e por fim o uso de *Class Weights*. É válido ressaltar que, para balancear a quantidade de criação de novos dados para as classes minoritárias, adotou-se o uso de *Class Weights* em ambos balanceamentos, ou seja, foi usado em conjunto com EDA e também com os dados gerados por GenAI. Quanto ao tratamento textual, cada entrada é processada para a remoção de caracteres especiais e dígitos, além da

conversão de todo o conteúdo textual para letras minúsculas, a fim de garantir uniformidade e compatibilidade com o processo de tokenização.

O treinamento dos modelos é realizado com os dados de treino e validação, permitindo o ajuste de seus parâmetros e a mitigação do *overfitting*. Durante o treino, cada etapa de treinamento é avaliada, e o modelo que tiver a melhor acurácia nos dados de validação será utilizado na etapa de teste. Por fim, os modelos treinados são avaliados com base no conjunto de teste, possibilitando a verificação de sua capacidade de generalização e desempenho na tarefa proposta.

Figura 4.2 – Fluxograma da metodologia proposta.



Fonte: Próprio Autor

4.2.1 Pré-processamento

Os dados do *dataset* utilizados são inicialmente submetidos a uma etapa de balanceamento antes do início do processamento textual voltado ao treinamento do modelo. Essa etapa tem como objetivo mitigar o desbalanceamento entre as classes, promovendo uma distribuição mais equitativa das amostras no conjunto de treinamento.

Para esse fim, são empregadas técnicas de EDA, aplicadas exclusivamente às instâncias pertencentes às classes minoritárias e aos dados de treinamento. Especificamente, utiliza-se apenas a operação de substituição aleatória de palavras, limitada a 3% do total de palavras de cada texto, de modo a preservar o contexto semântico original e evitar alterações excessivas no significado das amostras.

Adicionalmente, é adotada a estratégia de *Class Weights*, na qual são calculados pesos proporcionais à frequência de cada classe no conjunto de treinamento. Para isso, considera-se o número total de amostras e a quantidade de instâncias pertencentes a cada classe, atribuindo pesos maiores às classes com menor representatividade.

Complementarmente, aplica-se uma estratégia de geração de dados sintéticos por meio do uso de GenAI, com o objetivo de ampliar a diversidade de exemplos disponíveis para as classes minoritárias. Para isso, os dados de treino obtidos na divisão do conjunto de dados foram enviados à plataforma *ChatGPT* com o seguinte *prompt*:

Prompt de Entrada

```
Enhance the dataset contained in this CSV file by paraphrasing and
slightly diversifying the texts in the 'posts' column, focusing only
on rows where the 'predicted' column has the classes 'positive'. The
goal is to generate additional variations that help balance the dataset
across classes and improve model generalization during training. When
paraphrasing, preserve the original semantic meaning but introduce
linguistic diversity through changes in structure, vocabulary, and tone.
After generating these new samples, append them to the original dataset
and output a new CSV file containing both the original and enriched
data. Keep the structure of the file, just add new data. Be careful with
overfitting; the model needs better generalization, so use a high variety
of words but maintain the meaning of phrases. For each phrase in these
examples, generate only one paraphrasing data point.
```

Tal *prompt* visa enriquecer o conjunto de dados, mantendo a estrutura semântica e o significado do texto, de forma que utilize uma variedade de palavras, mas que não perca o sentido dos textos para os quais esta sendo utilizado o enriquecimento dos dados.

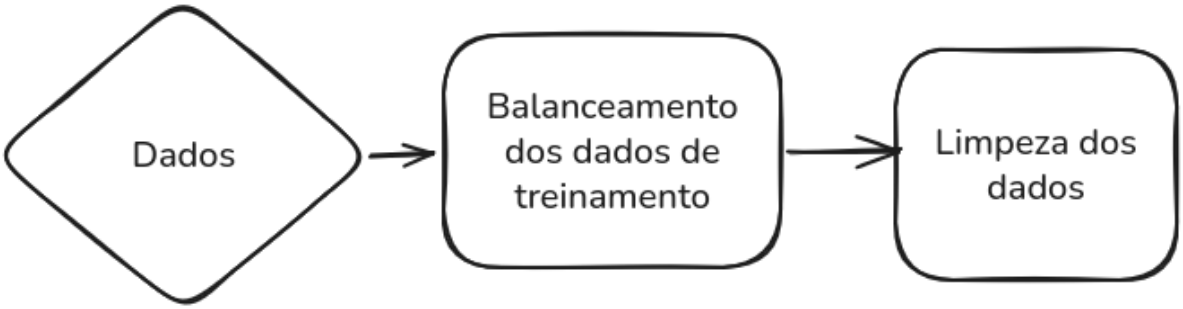
Em conjunto, essas abordagens visam favorecer o processo de aprendizado do modelo, reduzindo a tendência de viés em direção às classes majoritárias durante o treinamento. Ao mesmo tempo, busca-se evitar o fenômeno de *overfitting*, uma vez que a adição de dados sintéticos é realizada de forma controlada e combinada com a técnica de *Class Weights*, equilibrando robustez e capacidade de generalização do modelo.

Após o balanceamento, os textos passam por uma etapa de limpeza, cujo intuito é padronizar e normalizar os dados textuais. Nesta fase, são removidos caracteres especiais e dígitos, além de converter todos os caracteres para letras minúsculas. Essa padronização evita que o modelo diferencie palavras apenas por variações de capitalização, promovendo maior consistência no reconhecimento dos padrões linguísticos.

4.2.2 Divisão dos dados

A divisão dos dados foi realizada de forma estratificada, com o objetivo de garantir a representatividade das classes em todas as partições. O conjunto original foi segmentado em 80%

Figura 4.3 – Fluxograma de todo o pipeline de pré-processamento dos dados.



Fonte: Próprio Autor

para o treinamento do modelo, 10% para a validação durante o processo de treinamento, e os 10% restantes foram reservados para a avaliação final do modelo após o término do treinamento.

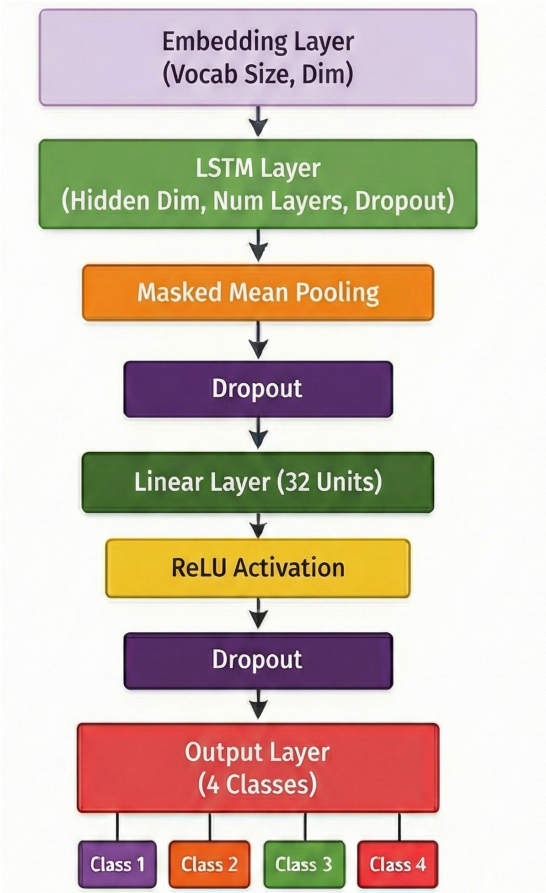
4.2.3 Treinamento do modelo

Esta seção apresentará cinco estratégias de treinamento diferentes, cada uma relacionada a um modelo distinto, sendo eles: LSTM, xLSTM, BERT, um modelo híbrido que utiliza a arquitetura BERT e CNN, e, por fim, LLM.

4.2.3.1 Arquitetura baseada em *Long Short Term Memory Neural Network*

A arquitetura do modelo de classificação baseia-se em uma rede LSTM estruturada em fluxo sequencial. O processamento inicia-se com uma camada de *Embedding* de 256 dimensões, aplicada a um vocabulário de 1.000 termos, utilizando máscaras de atenção para normalizar o comprimento das entradas. O núcleo de processamento é composto por duas camadas LSTM sobrepostas, cada uma com 256 unidades. A saída é consolidada por uma camada de *Mean Pooling* e direcionada a uma cabeça de classificação que inclui: uma camada de *Dropout* a uma taxa de 30%, uma camada densa intermediária de 32 neurônios com uma função de ativação *ReLU*, um segundo estágio de *Dropout* com a mesma taxa anterior e uma camada de saída com 4 neurônios para a classificação dos sentimentos. 4 neurônios, correspondentes às classes de sentimento.

Figura 4.4 – Arquitetura da estratégia ao utilizar *Long Short Term Memory Neural Network*.



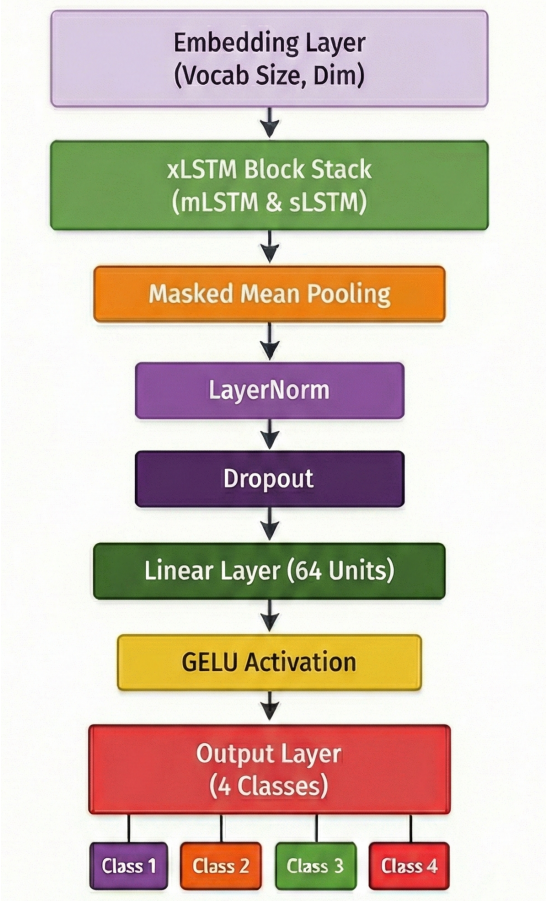
Fonte: Próprio Autor

A arquitetura, portanto, combina uma representação vetorial eficiente de palavras com a capacidade de memória de uma LSTM. As camadas de *Dropout* garantem a robustez do modelo, e as camadas lineares finais traduzem os padrões aprendidos em uma previsão de sentimento.

4.2.3.2 Arquitetura baseada em *Extended Long Short Term Memory Neural Network* (xLSTM)

A estratégia de classificação baseia-se na arquitetura xLSTM, iniciando com um vocabulário de 5.500 termos, oferecendo maior conhecimento do contexto geral do conjunto de dados, o que a torna uma estratégia diferente do modelo LSTM, obtendo um desempenho melhor com mais vocabulário disponível. O modelo utiliza uma camada de *Embedding* de 256 dimensões que alimenta blocos empilhados de *mLSTM* e *sLSTM*. A rede integra uma função de ativação GELU e normalização das camadas para estabilidade. A saída dos blocos é consolidada via *Mean Pooling* mascarado, seguida de normalização e *Dropout* com uma taxa de 10%. O estágio final consiste em uma camada densa de 64 unidades que projeta os dados para os 4 neurônios de saída correspondentes às classes de sentimento.

Figura 4.5 – Arquitetura do modelo xLSTM.

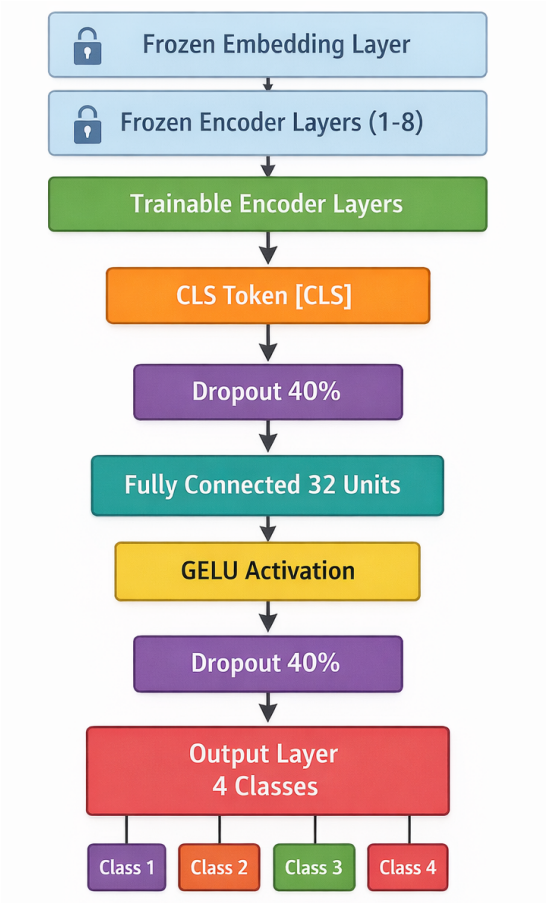


Fonte: Próprio Autor

4.2.3.3 Arquitetura baseada em *Bidirectional Encoder Representations from Transformers*

O modelo de *transfer learning* utiliza uma arquitetura de transformadores configurada para um ajuste seletivo, mantendo os pesos das camadas de *embeddings* e dos oito primeiros blocos do codificador congelados. A atualização dos gradientes concentra-se exclusivamente nas quatro camadas finais da rede para especialização na tarefa. A representação contextual é extraída do estado oculto do primeiro *token* da sequência, pois contém informação resumida sobre toda a sequência, sendo, portanto, menos custosa de ser utilizada na classificação, e é submetida a um *Dropout* de 20%. O vetor resultante passa por uma camada linear de 32 dimensões com ativação GELU e um segundo estágio de regularização, finalizando em uma camada de saída com quatro neurônios para classificação dos sentimentos.

Figura 4.6 – Arquitetura da estratégia ao utilizar BERT.



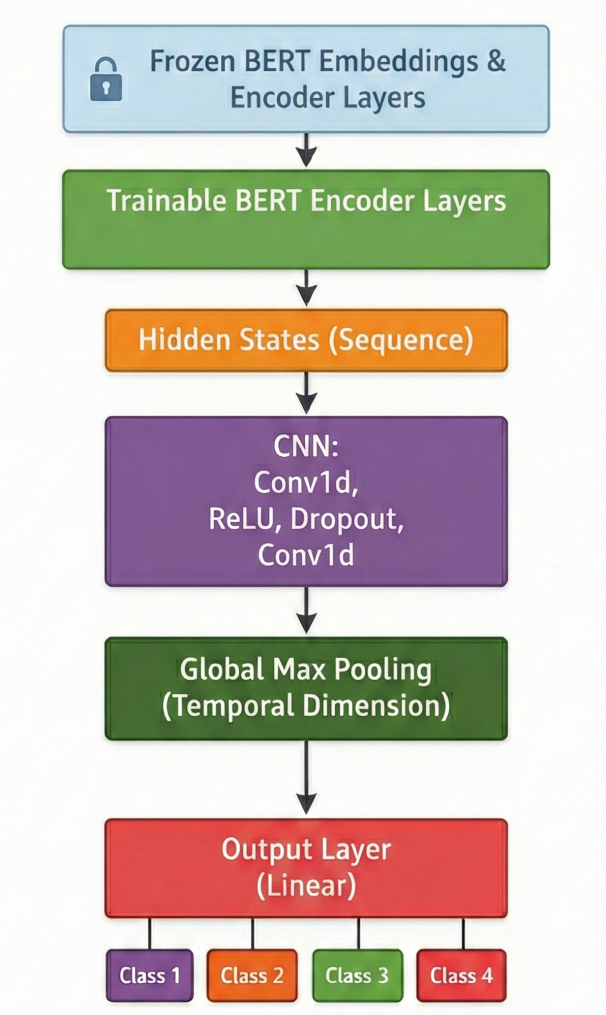
Fonte: Próprio Autor

4.2.3.4 Arquitetura do modelo híbrido *Bidirectional Encoder Representations from Transformers* e *Convolutional Neural Networks*

A estratégia de modelagem utiliza uma arquitetura híbrida que integra *Transformers* para contexto global e CNN para padrões locais. O modelo base aplica um ajuste fino, mantendo as camadas de *embeddings* e os oito primeiros blocos do codificador congelados, concentrando o aprendizado exclusivamente nas camadas superiores.

Os estados ocultos da última camada do codificador são transpostos para um bloco convolucional unidimensional em dois estágios. O primeiro estágio utiliza convolução com 128 canais, *kernel* de tamanho 5, função de ativação *ReLU* e *Dropout* de 30%. O segundo estágio refina a representação para 32 canais, utilizando *padding* para preservar a dimensão sequencial. A consolidação das características é realizada por uma operação de *Max Pooling*, seguida por uma camada linear de saída que projeta os dados para os quatro neurônios das classes de sentimento.

Figura 4.7 – Arquitetura da estratégia ao utilizar BERT em conjunto com a arquitetura CNN.



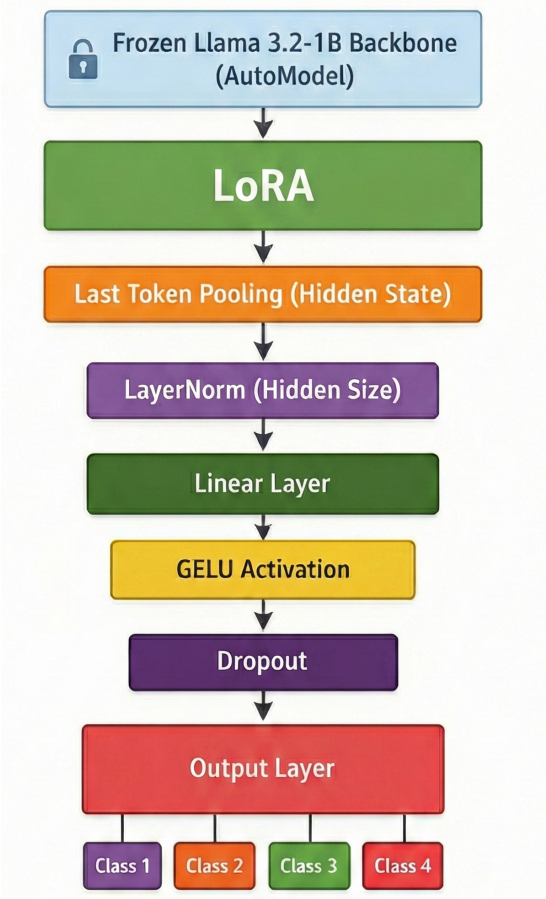
Fonte: Próprio Autor

4.2.3.5 Arquitetura baseada em *Large Language Model* (LLM)

A estratégia de modelagem proposta fundamenta-se na utilização do modelo LLM *Llama* 3.2, baseado em uma arquitetura *Transformer*. O processamento inicial dos dados é realizado por um tokenizador especializado que padroniza as sequências de texto para um comprimento fixo, utilizando máscaras de atenção. Para viabilizar o treinamento e a adaptação do *Llama*, o modelo integra uma estrutura de *Low-Rank Adaptation* (LoRA) aplicada aos módulos de projeção de atenção e camadas finais do modelo, permitindo a especialização do conhecimento linguístico para a tarefa de classificação com o parâmetro *rank* com valor 16 e o *alpha* com valor 32.

A extração de características da rede ocorre por meio de uma técnica de *pooling* no estado oculto do último *token* válido de cada sequência, já que carrega todo um contexto semântico importante para utilizar na classificação, consolidando a representação contextual final gerada pelo *Llama*. Este vetor é então processado por uma cabeça de classificação sequencial, composta por uma camada de normalização, que estabiliza as ativações provenientes do núcleo do modelo. Na sequência, a dimensionalidade é reduzida por uma camada linear para 16 unidades, seguida pela função de ativação GELU e uma camada de *Dropout* com taxa de 40%. O estágio final consiste em uma camada linear de saída que mapeia as representações para as quatro classes de sentimento alvo.

Figura 4.8 – Arquitetura do modelo LLM.



Fonte: Próprio Autor

4.2.4 Avaliação do modelo

Após o treinamento dos modelos, foram aplicadas métricas de avaliação com o objetivo de mensurar o desempenho obtido frente à tarefa proposta. As métricas utilizadas foram: acurácia, precisão, revocação (ou sensibilidade) e *F1-Score*. Tais métricas permitem uma análise abrangente da capacidade do modelo em classificar corretamente os dados, especialmente em cenários com desequilíbrio entre as classes.

As respectivas fórmulas utilizadas para o cálculo dessas métricas estão descritas nas Equações (2.10), (2.11), (2.12) e (2.13).

Para a consolidação das métricas em um cenário multiclasse, optou-se pela utilização da **média macro**. Diferente da média ponderada, que prioriza o volume total de acertos, a abordagem macro avalia o desempenho do modelo sob uma perspectiva de equidade entre as categorias.

Nesse método, a precisão, a revocação e o *F1-score* são calculados isoladamente para cada classe, e o resultado final é obtido por meio da média aritmética simples desses valores. Essa escolha metodológica é estratégica para o presente trabalho, pois assegura que cada estado emocional tenha o mesmo peso na avaliação final, independentemente de sua representatividade numérica no *dataset*. Dessa forma, o modelo é penalizado caso negligencie as classes minoritárias para favorecer a classe majoritária, garantindo um sistema de classificação mais justo e sensível às nuances emocionais menos frequentes, mas clinicamente relevantes.

5 Experimentos e Resultados

Este capítulo apresenta os resultados obtidos com os métodos e estratégias apresentados no Capítulo 4 dos modelos BERT, BERT em conjunto com CNN, LSTM, xLSTM e o uso do LLM *Llama*. Na seção 5.1 apresentam-se as especificações da máquina utilizada para treinamento, bem como as configurações e hiperparâmetros utilizados. Por fim, na seção 5.2, mostra-se uma análise dos resultados de cada método de balanceamento utilizado e, ao final, uma análise geral dos modelos.

5.1 *Setup* de experimentos

Para a execução dos algoritmos, utilizou-se o ambiente do *CSILab* com uma máquina *Intel* i9-10900 com 10 núcleos físicos (20 *threads*) de 2,80 GHz e 128 GB de RAM DDR4, que possui uma GPU RTX 3090, com 24 GB de RAM GDDR6X e mais de 10 mil cuda cores. A implementação de cada algoritmo dos modelos classificadores foi realizada utilizando a linguagem *Python* juntamente com o *framework Pytorch*. Por fim, utilizou-se a biblioteca *Scikit Learn* para avaliação das métricas dos modelos.

5.1.1 Configuração dos parâmetros de treinamento

O treinamento dos modelos foi realizado utilizando configurações distintas de taxa de aprendizado e tamanho de lote, definidas de acordo com a arquitetura avaliada. Para o modelo baseado em LLM, foram utilizadas quatro épocas de treinamento, com *learning rate* de $1{,}1 \times 10^{-4}$ e tamanho de lote igual a oito. Já para os modelos baseados em arquiteturas recorrentes do tipo LSTM e xLSTM, o treinamento foi conduzido com oito épocas, adotando-se um *learning rate* de 2×10^{-4} e tamanho de lote de 64, visando o melhor exploração temporal das sequências textuais. Por fim, os modelos baseados em BERT e BERT com CNN foram treinados por 4 épocas, com *learning rate* de $1{,}1 \times 10^{-4}$ e tamanho de lote de 32.

Em todos os experimentos, a função de perda utilizada foi a entropia cruzada categórica, adequada para tarefas de classificação multi-classe.

A otimização dos pesos dos modelos foi realizada por meio de um otimizador adaptativo com regularização *AdamW*, permitindo ajustes dinâmicos das taxas de atualização dos parâmetros ao longo do treinamento. Adicionalmente, foi adotada uma estratégia de agendamento da taxa de aprendizado, com redução progressiva linear ao longo das épocas, incluindo uma fase inicial de aquecimento, de modo a favorecer a convergência e evitar oscilações abruptas no início do processo de treinamento a uma taxa de 10%.

5.2 Resultados

Nesta seção é apresentado a análise dos resultados de cada método de balanceamento aplicado no conjunto de dados de treino, assim como o desempenho do conjunto de dados base.

5.2.1 Estudo da utilização do conjunto de dados Base

A Tabela 5.1 evidencia uma distinção consistente de desempenho em favor das arquiteturas fundamentadas em mecanismos de atenção e do modelo híbrido, em detrimento das redes recorrentes puras. O modelo BERT destacou-se como a abordagem mais robusta do experimento, alcançando a maior acurácia (0,8106) e o melhor *F1-Score* (0,7653), o que atesta sua superior capacidade de capturar dependências contextuais e de generalizar adequadamente sobre os dados. Em contraste, o LLM, embora tenha apresentado a maior precisão do conjunto (0,7925), exibiu um comprometimento relevante da revocação, caracterizando um *trade-off* que sugere um comportamento mais conservador na identificação de instâncias positivas. Por sua vez, o modelo híbrido demonstrou um perfil estatístico mais equilibrado, com valores de precisão e revocação praticamente equivalentes.

Tabela 5.1 – Comparativo das métricas de desempenho dos modelos testados.

Modelo	Acurácia	Precisão	Revogação	F1-Score
BERT	0,810577	0,777172	0,756040	0,765267
BERT-CNN	0,800962	0,754933	0,750563	0,748570
LLM Llama	0,807692	0,792534	0,721529	0,750418
LSTM	0,713462	0,519894	0,558484	0,538134
xLSTM	0,750962	0,701424	0,641129	0,658908

Fonte: Próprio autor.

No contexto das redes recorrentes, observa-se que o xLSTM apresentou avanços significativos em relação à arquitetura LSTM tradicional, elevando a acurácia para 0,7510 e reduzindo parcialmente as limitações associadas à modelagem de dependências de longo prazo. Ainda assim, a LSTM convencional obteve os piores resultados em todos os indicadores avaliados, com um *F1-Score* de apenas 0,5381, evidenciando sua dificuldade em lidar com a complexidade dos dados quando comparada às arquiteturas de estado da arte.

Sob outra perspectiva, ao confrontar esses resultados com os cenários de treinamento e validação, observa-se, conforme ilustrado na curva de treinamento da Figura 5.1, que os modelos baseados em *Transformers* (BERT, Híbrido e LLM Llama) atingem rapidamente seu pico de desempenho no conjunto de validação, por volta da quarta época, o que corrobora os valores de acurácia superiores a 0,80 reportados na Tabela 5.1. Entretanto, o gráfico da Figura 5.1 revela uma discrepância no comportamento da LLM Llama, cuja acurácia no conjunto de treinamento cresce abruptamente até valores próximos de 0,95, enquanto a acurácia de validação permanece praticamente estável. Tal divergência indica uma memorização excessiva

dos dados de treino, explicando sua elevada precisão, mas também sugerindo a ocorrência de *overfitting*. Em contrapartida, as redes recorrentes exibem um processo de aprendizagem mais gradual e aproximadamente linear, demandando até oito épocas para estabilização. A superioridade do xLSTM em relação ao LSTM, observada nas métricas finais, é corroborada pela inclinação mais acentuada de sua curva de aprendizado, embora ambas permaneçam aquém da capacidade de modelagem demonstrada pelas arquiteturas baseadas em mecanismos de atenção.

5.2.2 Estudo do impacto da utilização do método de *Class Weights*

A Tabela 5.2 apresenta os resultados obtidos com a aplicação do método de *class weights*, evidenciando a superioridade do modelo LLM Llama como a arquitetura de melhor desempenho neste experimento; todavia, não superou os resultados de acurácia dos experimentos sem o método. Esse modelo atingiu a acurácia máxima observada no experimento (0,8000), além de registrar o maior *F1-Score* (0,7590), superando de forma marginal o BERT, cuja acurácia foi de 0,7952. Esses resultados indicam que a ponderação das classes contribuiu para uma melhor adaptação dos modelos baseados em mecanismos de atenção ao desbalanceamento do conjunto de dados.

Tabela 5.2 – Comparativo das métricas de desempenho dos modelos testados utilizando *Class Weights*

Modelo	**Acurácia**	**Precisão**	**Revocação**	**F1-Score**
BERT	0,7952	0,7228	0,7815	0,7462
BERT-CNN	0,7817	0,7098	0,7901	0,7374
LLM Llama	0,8000	0,7545	0,7638	0,7590
LSTM	0,5923	0,5439	0,6355	0,5437
xLSTM	0,6962	0,6288	0,7157	0,6550

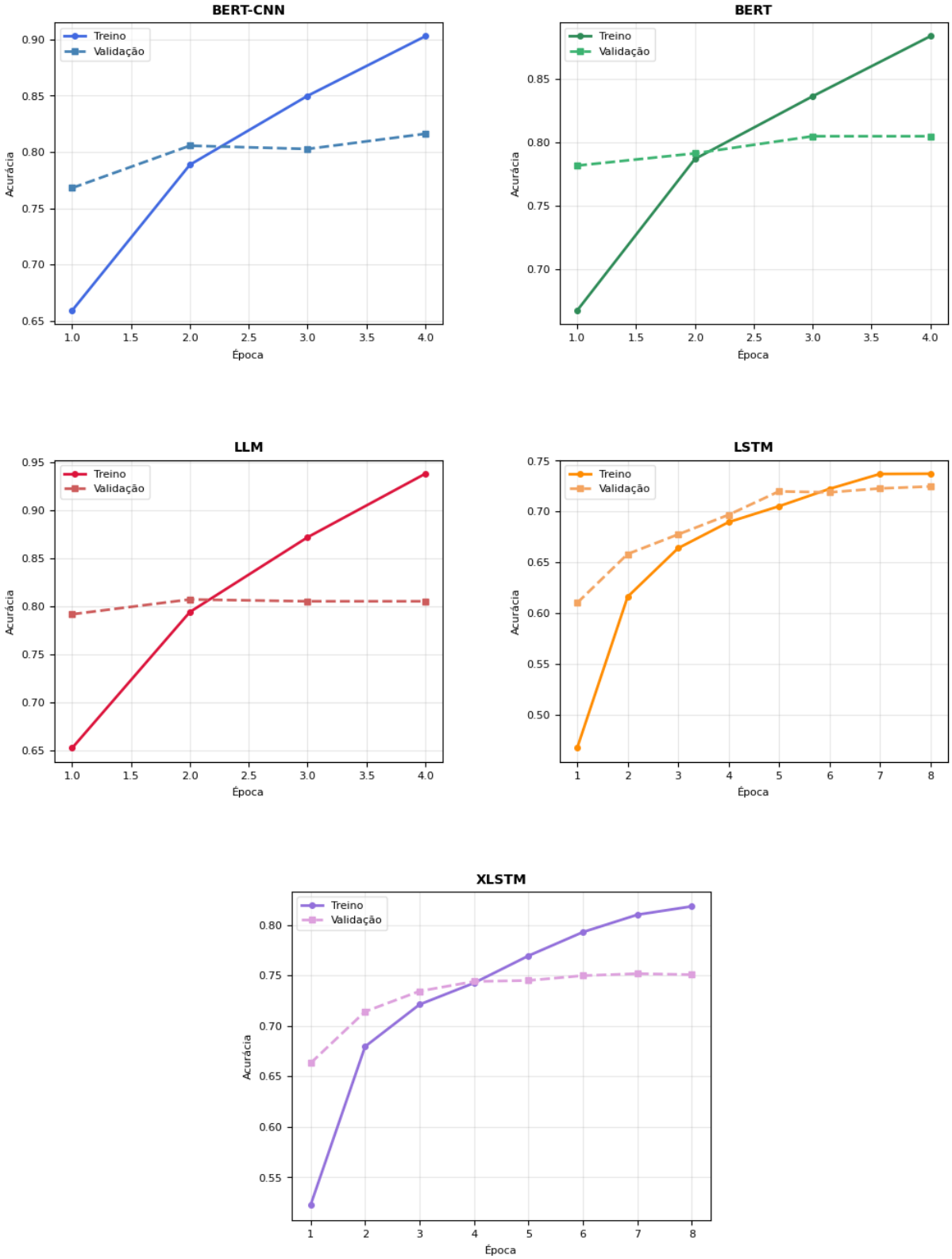
Fonte: Próprio autor.

Apesar do desempenho global inferior, o modelo híbrido destacou-se pela sua capacidade de identificar instâncias da classe positiva, alcançando a maior revocação entre todos os modelos avaliados (0,7901). Tal comportamento sugere que essa arquitetura é particularmente adequada para cenários em que a minimização de falsos negativos é prioritária, ainda que esse ganho ocorra em detrimento da precisão, que se manteve em um patamar inferior (0,7098).Para um melhor entendimento, pode-se observar a Figura 5.2.

Na Figura 5.2 mostra-se um alto índice de acerto em duas classes (negativo e neutro), o que pode ser observado também na Figura 5.3, com um maior índice de acerto.

Porém, apesar de obter uma acurácia maior, o modelo híbrido proporciona um equilíbrio maior ao acertar as outras classes, mesmo sendo inferior em acurácia, demonstrando sua alta revocação neste cenário.

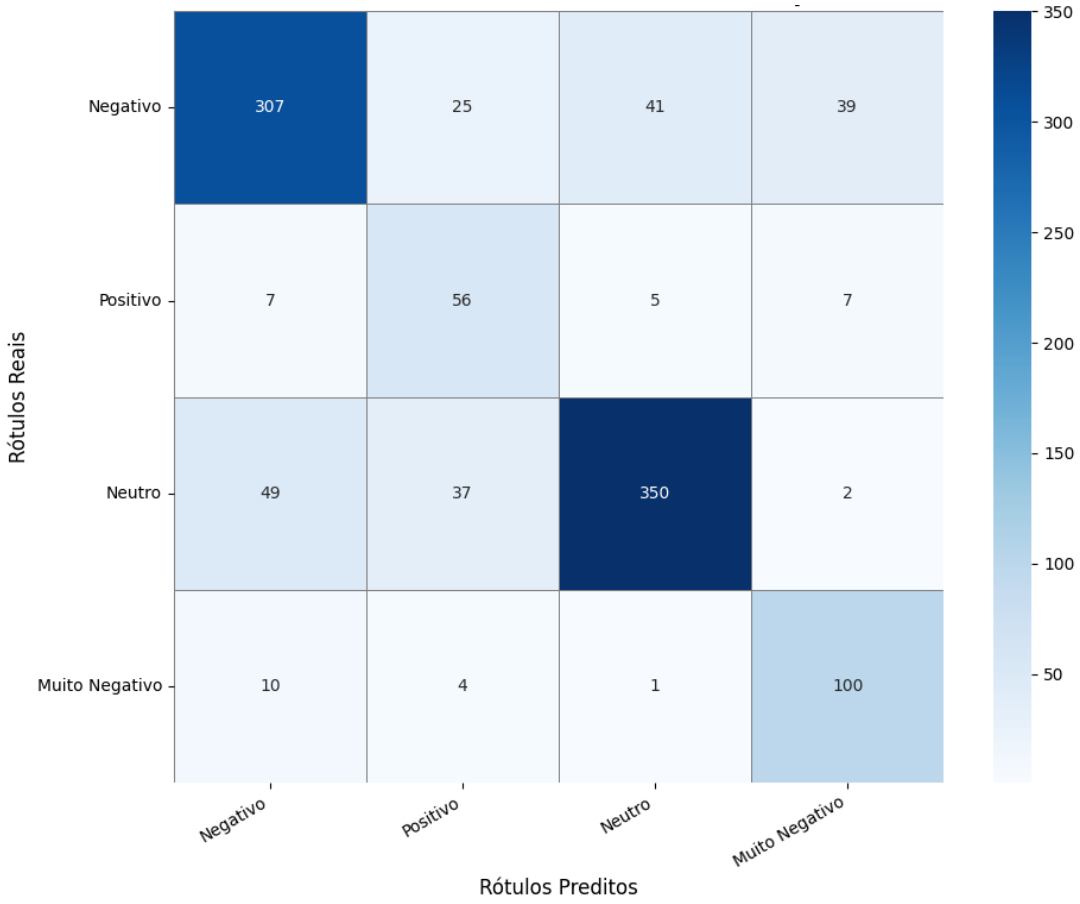
Figura 5.1 – Curvas de acurácia durante o treinamento referente ao conjunto de treino e ao conjunto de validação dos modelos avaliados.



Fonte: Próprio autor.

No que se refere às redes recorrentes, observa-se uma desvantagem clara em relação às arquiteturas baseadas em *Transformers*. Embora o xLSTM tenha apresentado um avanço expres-

Figura 5.2 – Matriz de confusão do modelo BERT-CNN.



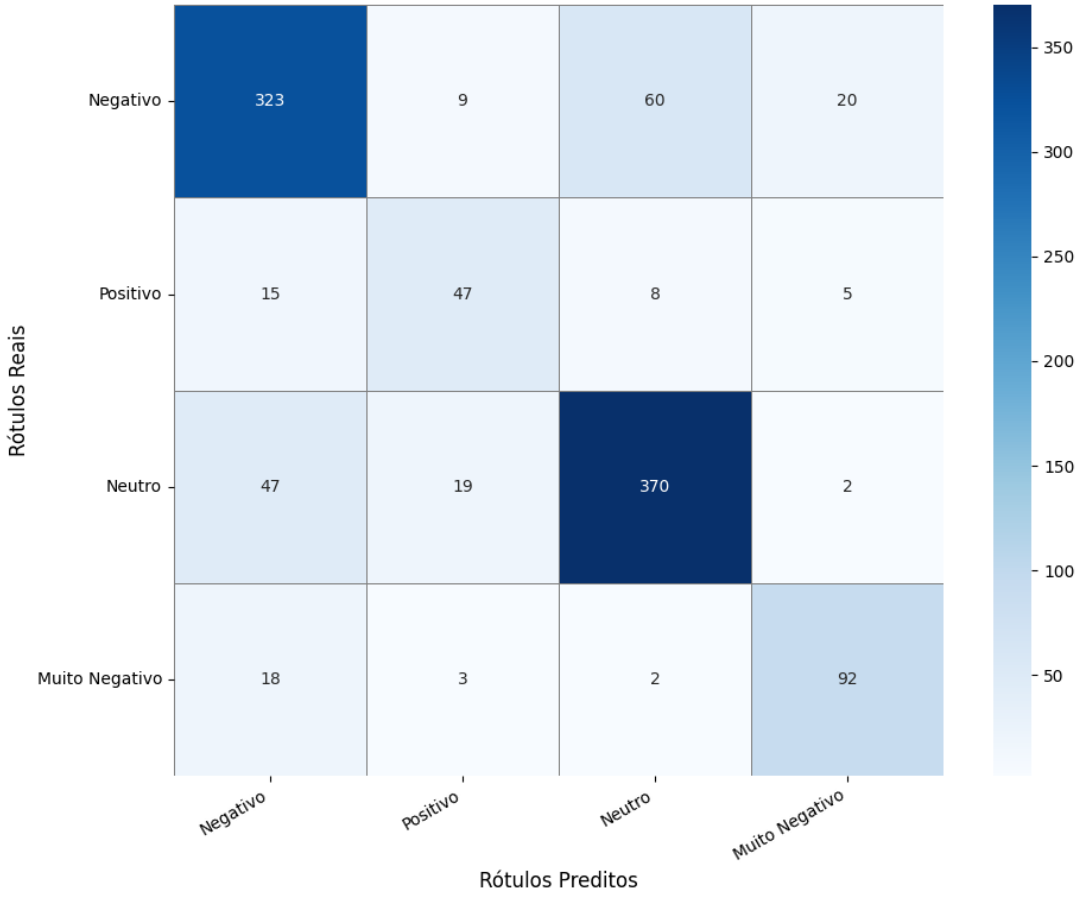
Fonte: Próprio autor.

sivo em acurácia quando comparado à LSTM tradicional, ambas as arquiteturas permaneceram abaixo do desempenho alcançado pelos modelos de atenção, indicando limitações estruturais para lidar com a complexidade do problema proposto.

A análise das curvas de treinamento e validação, ilustradas na Figura 5.4, reforça essas conclusões. Os modelos LLM e BERT demonstram uma convergência acelerada, atingindo níveis elevados de acurácia já por volta da quarta época. No entanto, o gráfico também evidencia um comportamento característico de *overfitting*, uma vez que a acurácia no conjunto de treinamento continua a crescer de forma acentuada, ultrapassando 0,90, enquanto a acurácia de validação se estabiliza em torno de 0,80.

Em contraste, as redes recorrentes apresentam um processo de aprendizado mais lento e aproximadamente linear, estendendo-se até a oitava época. Esse padrão visual confirma que, mesmo com maior tempo de treinamento, tais arquiteturas não dispõem de capacidade paramétrica suficiente para competir com os mecanismos de atenção frente à complexidade deste *corpus* específico. Observa-se, adicionalmente, certa instabilidade durante o treinamento da rede LSTM, embora, ao final do processo, a acurácia de validação convirja para valores compatíveis com aqueles observados no treinamento, ainda que em um patamar inferior aos demais modelos.

Figura 5.3 – Matriz de confusão do modelo LLM.



Fonte: Próprio autor.

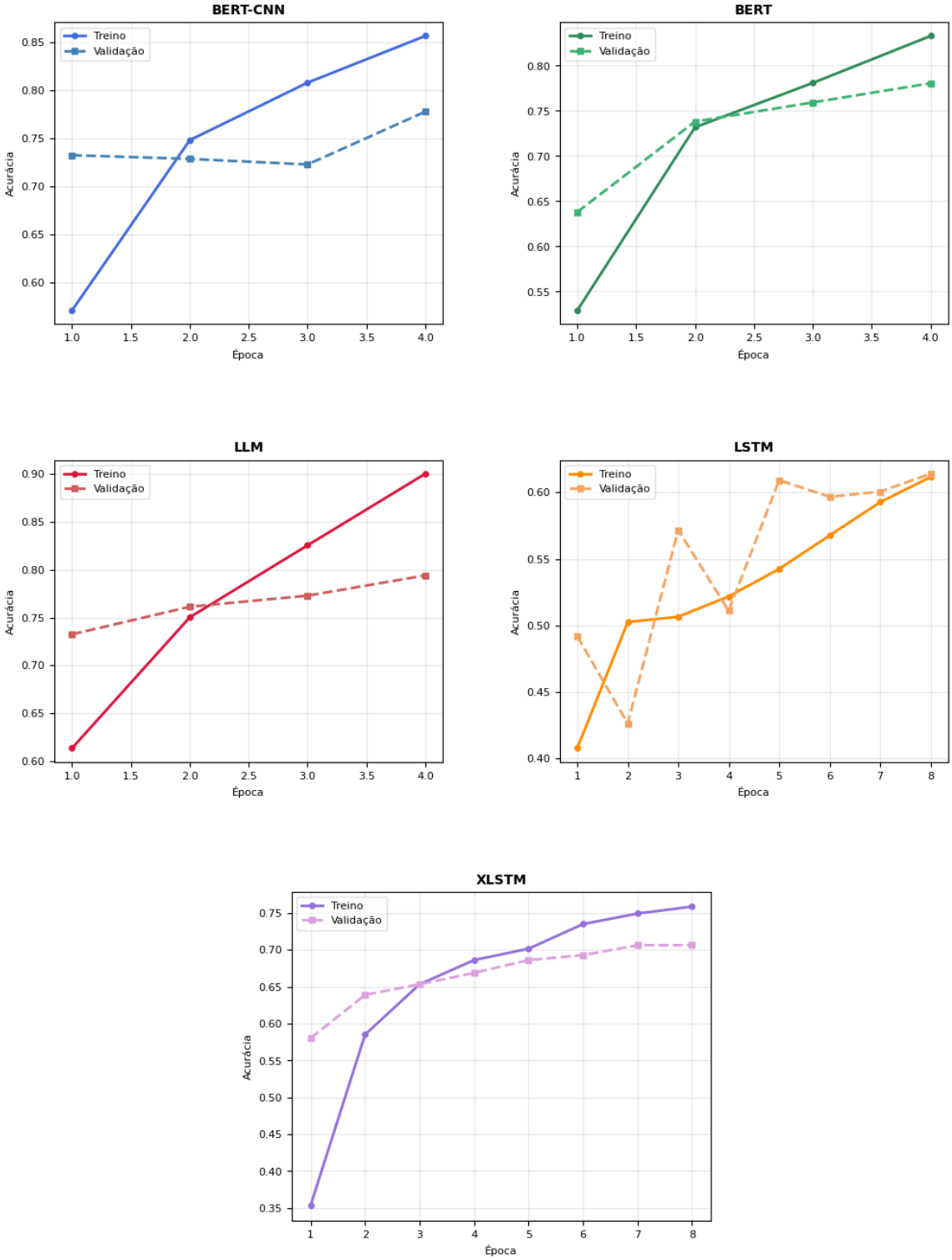
5.2.3 Estudo do impacto da utilização EDA em conjunto com a técnica *Class Weights*

A Tabela 5.3 apresenta os resultados obtidos com a aplicação combinada de EDA e do método de *Class Weights*. Observa-se que o modelo LLM alcançou o melhor desempenho global, registrando uma acurácia de 82% e o maior *F1-Score* (0,7738). Esse resultado indica uma capacidade superior de equilibrar precisão e revocação, sugerindo que a estratégia conjunta de aumento de dados e ponderação das classes contribuiu de forma efetiva para a melhoria da generalização, mesmo em um contexto de potencial desbalanceamento do conjunto de dados.

O modelo híbrido também apresentou um desempenho competitivo, com destaque para a revocação (0,7675), valor superior ao obtido pelo BERT. Esse comportamento sugere que a arquitetura híbrida para o contexto do *corpus* avaliado, quando treinada com dados aumentados, torna-se mais sensível à identificação correta de instâncias que estavam mais desbalanceadas, característica particularmente relevante em aplicações nas quais o custo associado a falsos negativos é elevado. O BERT, por sua vez, embora mantenha um desempenho robusto e consistente, apresentou resultados ligeiramente inferiores aos do LLM nesse cenário experimental.

Em contraste, o modelo LSTM apresentou o pior desempenho em todas as métricas

Figura 5.4 – Curvas de acurácia durante o treinamento utilizando *Class Weights* referente ao conjunto de treino e ao conjunto de validação dos modelos avaliados.



Fonte: Próprio autor.

avaliadas. Mesmo com a incorporação de técnicas de aumento de dados e da utilização de pesos na função de perda, as limitações estruturais inerentes às arquiteturas sequenciais tradicionais tornam-se evidentes quando comparadas aos modelos baseados em mecanismos de atenção. O

Tabela 5.3 – Comparativo das métricas de desempenho dos modelos testados utilizando EDA e *Class Weights*

Modelo	Acurácia	Precisão	Revocação	F1-Score
BERT	0,8000	0,7583	0,7439	0,7505
BERT-CNN	0,7971	0,7428	0,7675	0,7540
LLM Llama	0,8200	0,7764	0,7713	0,7738
LSTM	0,6702	0,5511	0,5819	0,5555
xLSTM	0,7288	0,6558	0,7175	0,6782

Fonte: Próprio autor.

xLSTM, apesar de demonstrar uma melhora significativa em relação à LSTM convencional, ainda permanece aquém do desempenho alcançado pelos modelos baseados em *Transformers*, indicando que os ganhos obtidos são incrementais, porém insuficientes para competir em cenários de maior complexidade.

A análise das curvas de aprendizado apresentadas na Figura 5.5 revela que, apesar da melhoria nos resultados globais, os modelos ainda exibem indícios de *overfitting*. Observa-se uma discrepância entre as curvas de treinamento e de validação, mesmo após a aplicação do aumento de dados por meio do EDA. Esse comportamento é mais pronunciado nos modelos baseados em *Transformers*, os quais, embora tenham alcançado os melhores resultados, ainda apresentam potencial para ajustes adicionais.

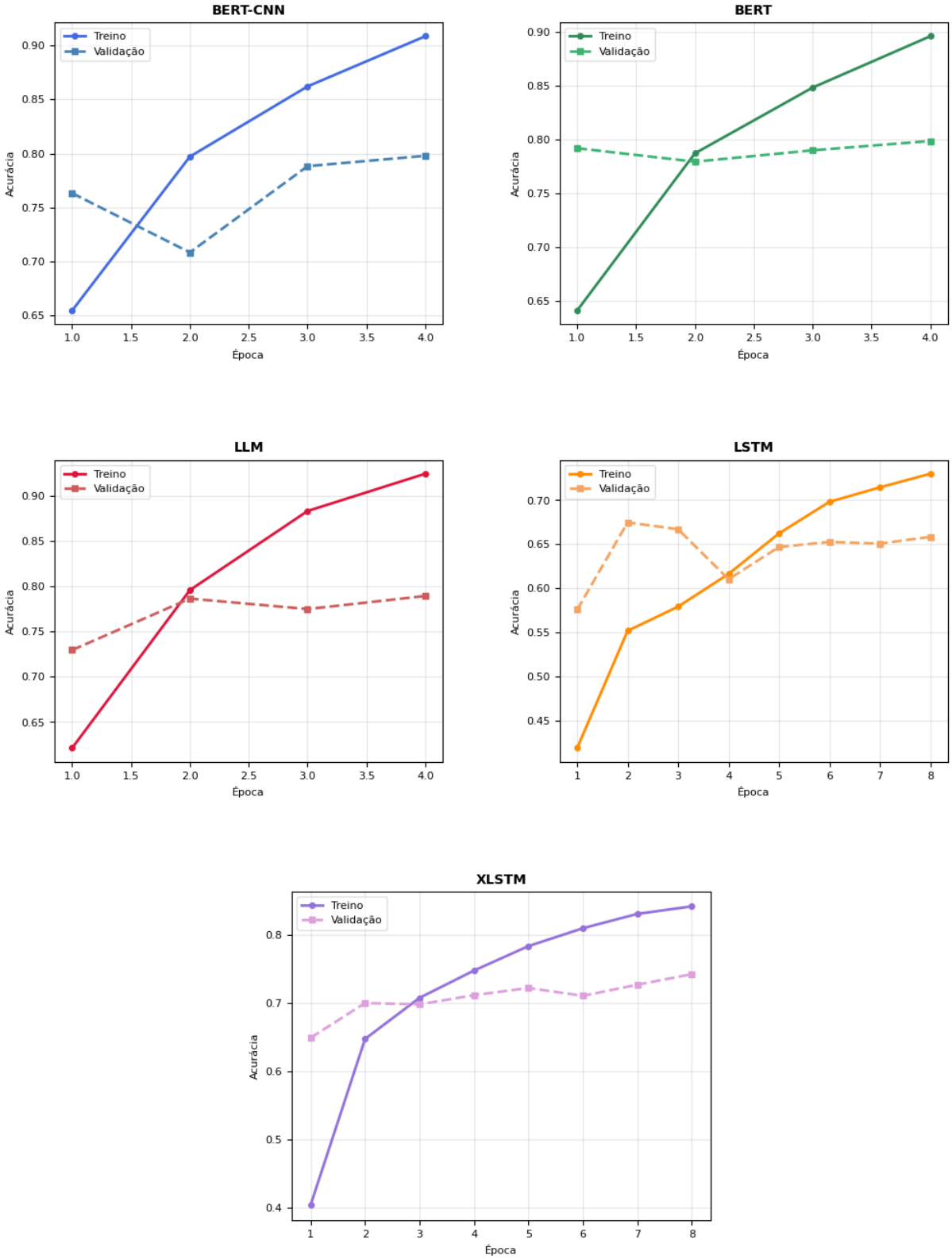
5.2.4 Estudo do impacto da utilização da GenAI para gerar dados sintéticos com *Class Weights*

A Tabela 5.4 apresenta os resultados obtidos com a combinação de dados sintéticos gerados por modelos generativos e a aplicação do método de *Class Weights*. Observa-se que o modelo BERT alcançou o melhor desempenho global, registrando uma acurácia de 81,83% e o maior *F1-Score* (0,7709). Esses resultados indicam que o modelo se beneficia de forma significativa do aumento da diversidade introduzido pelos dados sintéticos, mantendo um equilíbrio adequado entre precisão e revocação. Ademais, a utilização de pesos na função de perda contribuiu para a estabilização do processo de aprendizado, mitigando o viés em favor das classes majoritárias.

O modelo híbrido destacou-se pelo maior valor de revocação (0,7698), o que sugere uma maior sensibilidade na identificação correta das instâncias da classe-alvo. Esse comportamento é coerente com cenários em que a ampliação sintética dos dados favorece uma cobertura mais ampla do espaço semântico da classe minoritária. Contudo, observa-se uma leve redução na precisão, evidenciando um *trade-off* entre cobertura e seletividade.

O modelo LLM Llama apresentou desempenho consistente, porém inferior ao BERT nessa configuração experimental. Embora tenha mantido valores relativamente equilibrados de precisão e revocação, os resultados sugerem que essa arquitetura pode ser mais sensível a

Figura 5.5 – Curvas de acurácia durante o treinamento utilizando *Class Weights* e EDA referente ao conjunto de treino e ao conjunto de validação dos modelos avaliados.



Fonte: Próprio autor.

redundâncias semânticas introduzidas por técnicas de parafraseamento, o que pode impactar negativamente sua capacidade de generalização.

Tabela 5.4 – Comparativo das métricas de desempenho dos modelos testados utilizando dados sintéticos parafraseados e *Class Weights*.

Modelo	Acurácia	Precisão	Revocação	F1-Score
BERT	0,8183	0,7927	0,7629	0,7709
BERT-CNN	0,7942	0,7389	0,7698	0,7510
LLM Llama	0,7962	0,7588	0,7346	0,7456
LSTM	0,6635	0,6010	0,6573	0,6119
xLSTM	0,7202	0,6524	0,7185	0,6765

Fonte: Próprio autor.

As arquiteturas recorrentes, representadas pelos modelos LSTM e xLSTM, apresentaram desempenho inferior quando comparadas aos modelos baseados em *Transformers*. Apesar de o xLSTM demonstrar uma melhora relevante em relação ao LSTM convencional, os resultados indicam que redes recorrentes possuem capacidade limitada para explorar plenamente a diversidade linguística introduzida por dados sintéticos gerados por modelos generativos, mesmo quando combinadas com estratégias de ponderação da função de perda.

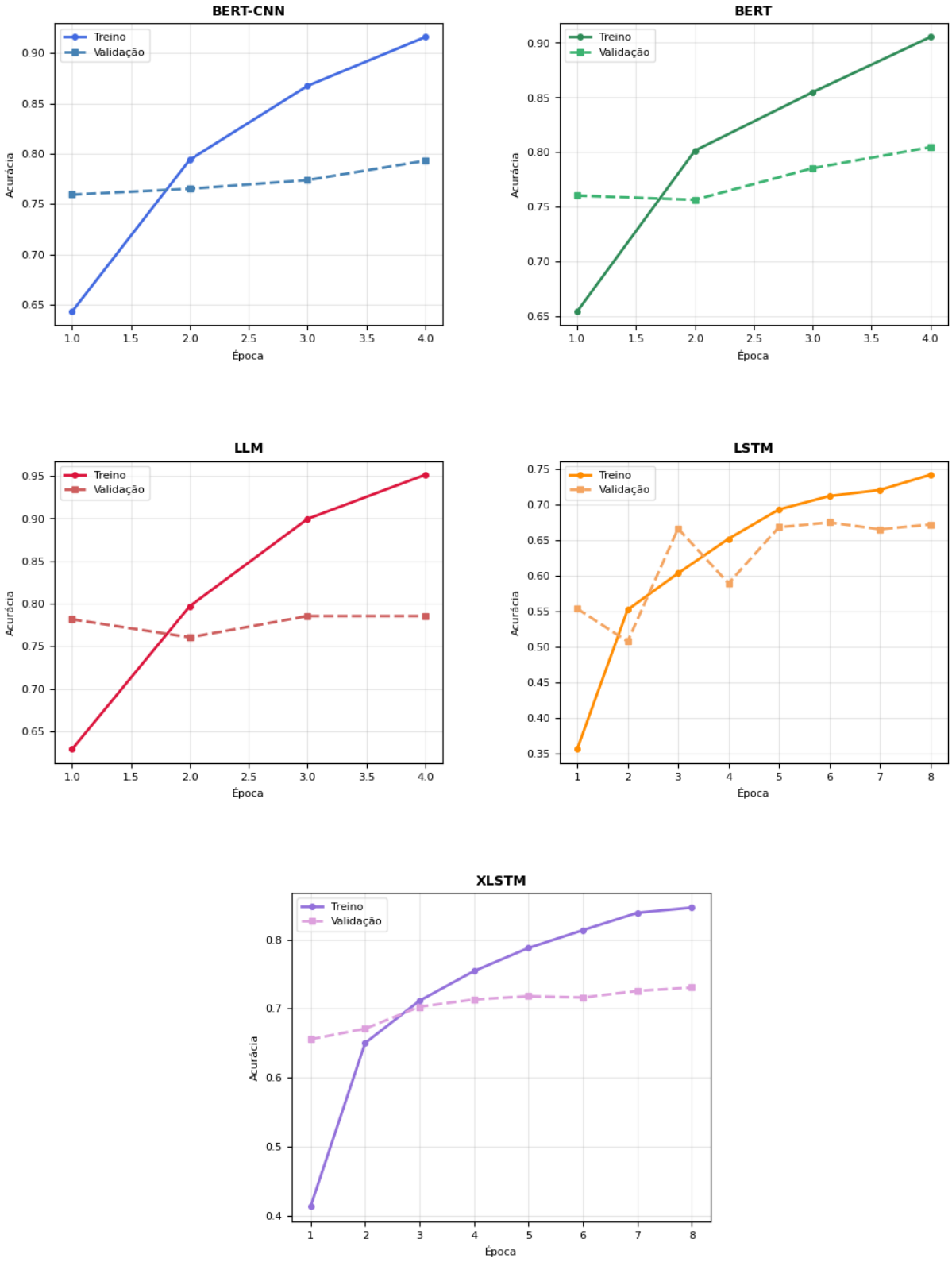
A análise das curvas de aprendizado apresentadas na Figura 5.6 revela que os modelos ainda exibem indícios de *overfitting*, sendo esse comportamento mais pronunciado nos modelos baseados em *Transformers* e no xLSTM. Tal padrão evidencia desafios adicionais de generalização diante do desequilíbrio de classes, mesmo após a tentativa de mitigação por meio da inserção de dados sintéticos. Esses resultados sugerem que, embora o aumento de dados contribua para ampliar a diversidade do conjunto de treinamento, ajustes adicionais no processo de regularização e na estratégia de geração de dados podem ser necessários para que os modelos alcancem um regime de aprendizado mais estável e generalizável.

5.3 Discussão dos resultados

As análises individuais dos experimentos evidenciam de forma consistente que o desempenho dos modelos é fortemente influenciado tanto pela arquitetura adotada quanto pelas estratégias empregadas para mitigar o desbalanceamento dos dados. Em todos os cenários avaliados, observa-se uma superioridade sistemática dos modelos baseados em mecanismos de atenção quando comparados às arquiteturas recorrentes tradicionais.

No cenário base, no qual não foram aplicadas técnicas explícitas de balanceamento, os modelos baseados em *Transformers* já demonstram elevada capacidade de generalização, alcançando rapidamente níveis de acurácia superiores a 80% no conjunto de teste. Contudo, a análise das curvas de treinamento e validação revela que, particularmente no caso do LLM, esse alto desempenho é acompanhado por um comportamento característico de *overfitting*. Tal fenômeno manifesta-se pela divergência progressiva entre as curvas, com a acurácia no conjunto de treinamento aproximando-se de 0,95, enquanto o desempenho no conjunto de validação

Figura 5.6 – Curvas de acurácia durante o treinamento utilizando *Class Weights* e dados sintéticos gerados por GenAI referente ao conjunto de treino e ao conjunto de validação dos modelos avaliados.



Fonte: Próprio autor.

permanece estagnado.

A introdução do método de *Class Weights* promoveu uma redistribuição mais equilibrada do processo de aprendizado, resultando em ganhos consistentes de revocação, especialmente no modelo híbrido. Na aplicação dos outros métodos de balanceamento, tal modelo seguiu-se se destacando na métrica de revocação, o que sugere um certo equilíbrio no acerto das classes. Essa estratégia mostrou-se eficaz na redução do viés em favor das classes majoritárias. No entanto, apesar desses avanços, a aplicação de pesos na função de perda não foi suficiente para eliminar completamente os indícios de *overfitting* observados nos modelos baseados em atenção.

No experimento que combinou EDA com o uso de *Class Weights*, verificou-se um ganho adicional no desempenho global, com destaque para o LLM Llama. Ainda assim, a análise das curvas de treinamento e validação indica que o aumento de dados, embora benéfico em termos de métricas finais, pode ter contribuído para a intensificação do *overfitting*, sobretudo nas arquiteturas baseadas em *Transformers*.

De forma complementar, a utilização de dados sintéticos parafraseados gerados por GenAI, em conjunto com o método de *Class Weights*, resultou em melhorias relevantes, especialmente para o BERT, que apresentou o melhor equilíbrio geral entre precisão, revocação e *F1-Score*. Entretanto, novamente observa-se que os ganhos quantitativos são acompanhados por limitações de generalização, evidenciadas pela persistente discrepância entre as curvas de treinamento e teste.

Por fim, conclui-se que os modelos baseados em *Transformers* superam de forma consistente aqueles derivados de redes recorrentes, com destaque para os modelos BERT e LLM Llama. Além disso, embora as estratégias de balanceamento de classes tenham contribuído para a melhoria de métricas específicas em determinados cenários, tais abordagens não foram suficientes para mitigar o fenômeno de *overfitting*, evidenciando desafios recorrentes de generalização no processo de aprendizado dos modelos avaliados.

6 Considerações Finais

Este capitulo apresenta as conclusões obtidas acerca da realização do trabalho, destacando o cumprimento dos objetivos e os resultados que foram obtidos. Além disso, são apresentadas as possibilidades futuras que podem ser utilizadas para a continuação do trabalho.

6.1 Conclusão

Este trabalho teve como objetivo investigar e comparar o desempenho de diferentes arquiteturas de redes neurais na tarefa de classificação de sentimentos em textos produzidos por pacientes oncológicos. O foco central da pesquisa foi a identificação de padrões emocionais capazes de subsidiar a oferta de cuidados paliativos psicológicos mais eficazes. Para esse propósito, foram avaliados modelos baseados em redes neurais recorrentes, como o LSTM e o xLSTM, bem como arquiteturas fundamentadas em mecanismos de atenção, incluindo o BERT, uma abordagem híbrida BERT+CNN e um LLM da família *Llama*, especificamente a versão 3.2. Adicionalmente, investigou-se o impacto de diferentes estratégias de mitigação do desbalanceamento de dados, tais como o uso de *Class Weights*, técnicas de aumento de dados por meio de EDA e a geração de dados sintéticos parafraseados com o auxílio de GenAI.

Os resultados obtidos demonstram, de forma consistente, a superioridade das arquiteturas baseadas em *Transformers* em relação às redes neurais recorrentes tradicionais. Em todos os cenários avaliados, modelos como BERT, LLM e a arquitetura híbrida apresentaram valores superiores de acurácia e *F1-Score*, evidenciando maior capacidade de capturar dependências semânticas complexas e de generalizar adequadamente em textos livres provenientes de redes sociais.

A análise dos diferentes cenários de balanceamento revelou que o tratamento do desbalanceamento de classes resultou em ganhos métricos moderados. O uso de *Class Weights* contribuiu para o aumento da revocação e para a redução do viés em favor das classes majoritárias. Por sua vez, as estratégias de aumento de dados, tanto por meio de EDA quanto pela geração de dados sintéticos com GenAI, proporcionaram ganhos adicionais nas métricas de desempenho. Entretanto, a análise das curvas de treinamento e validação indicou que essas técnicas também intensificaram o fenômeno de *overfitting*, especialmente nos modelos baseados em *Transformers*. Esse comportamento sugere que o aumento de dados, quando não cuidadosamente controlado, pode introduzir redundâncias semânticas capazes de comprometer a capacidade de generalização dos modelos.

De maneira geral, o modelo BERT destacou-se como a solução mais equilibrada ao longo dos experimentos, apresentando desempenho consistente em termos de precisão, revocação

e *F1-Score*, particularmente nos cenários que combinaram balanceamento de classes e dados sintéticos. O LLM LLama, embora tenha alcançado os maiores valores absolutos em alguns experimentos, demonstrou maior suscetibilidade ao *overfitting*, evidenciando a necessidade de estratégias adicionais de regularização para garantir um aprendizado mais estável e generalizável.

6.2 Trabalhos Futuros

Como proposta para trabalhos futuros, sugere-se a investigação de técnicas de quantização pós-treinamento e de poda de neurônios (*pruning*), com o objetivo de reduzir o custo computacional dos modelos, sem comprometer as características fundamentais de suas arquiteturas robustas. Essas abordagens podem viabilizar a aplicação dos modelos em ambientes com restrições de recursos, mantendo níveis adequados de desempenho.

De forma complementar, para mitigar o fenômeno de *overfitting* associado às estratégias de balanceamento de classes, recomenda-se a exploração de funções de perda sensíveis ao custo, como a *Focal Loss*. Além disso, a adoção de técnicas alternativas de aumento de dados sintéticos, como *back-translation*, pode contribuir para uma ampliação mais controlada da diversidade semântica do conjunto de treinamento, reduzindo a introdução de redundâncias.

Essas abordagens podem favorecer um equilíbrio mais refinado entre a complexidade estrutural dos modelos e sua capacidade de generalização, especialmente em cenários caracterizados por escassez ou desproporcionalidade de dados. Ademais, sugere-se o refinamento das técnicas já empregadas, como a exploração de diversas variações do LoRA(YANG et al., 2024), bem como a incorporação de dados provenientes de bases adicionais, com o objetivo de enriquecer o conjunto de dados. A ampliação e diversificação das fontes de dados tendem a aumentar o potencial de aprendizado dos modelos e a promover uma generalização mais robusta dos resultados obtidos.

Referências

AMINI, A. MIT Introduction to Deep Learning. 2023. <https://www.youtube.com/watch?v=QDX-1M5Nj7s&t=1203s>. Acessado em 8 de junho de 2025.

AMISSE, C.; JIJÓN-PALMA, M. E.; CENTENO, J. A. S. Fine-tuning deep learning models for pedestrian detection. Boletim de Ciências Geodésicas, SciELO Brasil, v. 27, n. 02, p. e2021013, 2021.

ASHENDEN, S. K.; BARTOSIK, A.; AGAPOW, P.-M.; SEMENOVA, E. Chapter 2 - introduction to artificial intelligence and machine learning. In: ASHENDEN, S. K. (Ed.). The Era of Artificial Intelligence, Machine Learning, and Data Science in the Pharmaceutical Industry. Academic Press, 2021. p. 15–26. ISBN 978-0-12-820045-2. Disponível em: <https://www.sciencedirect.com/science/article/pii/B9780128200452000039>.

AWASTHI, V.; TIWARI, M.; YADAV, A.; THAKUR, G.; PANDA, M. M.; KUMAR, H.; TRIPATHI, S. Optimizing brain tumor detection in mri scans through inceptionresnetv2 and deep stacked autoencoders with swiglu activation and sparsity regularization. MethodsX, Elsevier, v. 14, p. 103255, 2025.

BALAKRISHNAN, A.; IDICULA, S. M.; JONES, J. Deep learning based analysis of sentiment dynamics in online cancer community forums: An experience. SAGE, v. 27, p. 18, 2021.

BANERJEE, K.; C, V. P.; GUPTA, R. R.; VYAS, K.; H, A.; MISHRA, B. Exploring Alternatives to Softmax Function. 2020. Disponível em: <https://arxiv.org/abs/2011.11538>.

BANSAL, R. A survey on bias and fairness in natural language processing. arXiv, 2022.

BECK, M.; PöPPEL, K.; SPANRING, M.; AUER, A.; PRUDNIKOVA, O.; KOPP, M.; KLAMBAUER, G.; BRANDSTETTER, J.; HOCHREITER, S. xLSTM: Extended Long Short-Term Memory. 2024. Disponível em: <https://arxiv.org/abs/2405.04517>.

BELSIC, I.; STRYKER, C. O que é aprendizado supervisionado? 2024. <https://www.ibm.com/br-pt/think/topics/supervised-learning>. Accessed: 2026-03-02.

BERGMANN, C. S. D. What is backpropagation? [S.l.]: IBM.

BERGMANN, D. O que é aprendizado semissupervisionado? [S.l.]: IBM.

BERGMANN, D. What is machine learniing? [S.l.]: IBM.

BOCHENEK, B.; USTRNUL, Z. Machine learning in weather prediction and climate analyses—applications and perspectives. Atmosphere, v. 16, p. 16, 2022.

BOTTINO, S. M. B.; FRáGUAS, R.; GATTAZ, W. F. Depressão e câncer. Scielo Brasil, v. 36, p. 7, 2009.

BíBLIA SAGRADA. Barueri: Sociedade Bíblica do Brasil, 2011. Almeida Revista e Atualizada.

CHATZIMINA, M. E.; PAPADAKI, H. A.; PONTIKOGLOU, C.; TSIKNAKIS, M. A comparative sentiment analysis of greek clinical conversations using bert, roberta, gpt-2, and xlnet. bioengeneering, v. 11, p. 12, 2024.

CHEN, L.; VAROQUAUX, G. What is the role of small models in the llm era: A survey. arXiv preprint arXiv:2409.06857, 2024.

CHOPRA, A.; PRASHAR, A.; SAIN, C. Natural language processing. INTERNATIONAL JOURNAL OF TECHNOLOGY ENHANCEMENTS AND EMERGING ENGINEERING RESEARCH, v. 1, 2013.

CHRISTEN, P.; HAND, D. J.; KIRIELLE, N. A review of the f-measure: Its history, properties, criticism, and alternatives. ACM Comput. Surv., Association for Computing Machinery, New York, NY, USA, v. 56, n. 3, 2023. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3606367>.

da Silva, D. G.; MENESES, A. A. de M. Comparing long short-term memory (lstm) and bidirectional lstm deep neural networks for power consumption prediction. Energy Reports, v. 10, p. 3315–3334, 2023. ISSN 2352-4847. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352484723014208>.

DEY, J.; DESAI, D. Nlp based approach for classification of mental health issues using lstm and glove embeddings. International Journal of Advanced Research in Science, Communication and Technology, v. 10, p. 347–354, 2022.

DONG, J.; HE, F.; GUO, Y.; ZHANG, H. A commodity review sentiment analysis based on bert-cnn model. In: IEEE. 2020 5th International conference on computer and communication systems (ICCCS). [S.l.], 2020. p. 143–147.

DUBEY, S. R.; SINGH, S. K.; CHAUDHURI, B. B. Activation functions in deep learning: A comprehensive survey and benchmark. ScienceDirect, v. 503, p. 7, 2022.

EDARA, D. C.; VANUKURI, L. P.; SISTLA, V.; KOLLI, V. K. K. Sentiment analysis and text categorization of cancer medical records with lstm. Journal of Ambient Intelligence and Humanized Computing, v. 14, p. 17, 2023.

ELHARROUSSA, O.; MAHMOODA, Y.; BECHQITOA, Y.; SERHANIB, M. A.; BADIDIA, E.; RIFFIC, J.; TAIRIC, H. Loss functions in deep learning: A comprehensive review. arXiv, p. 36, 2025.

GARDAZI, N. M.; DAUD, A.; MALIK, M. K.; BUKHARI, A.; ALSAHFI, T.; ALSHEMAIMRI, B. Bert applications in natural language processing: a review. Artificial Intelligence Review, Springer, v. 58, n. 6, p. 1–49, 2025.

GEEKSFORGEEKS. What is LSTM. 2025. <https://www.geeksforgeeks.org/deep-learning/deep-learning-introduction-to-long-short-term-memory/>. Accessed: 2025-07-28.

GHOSH, K.; BELLINGER, C.; CORIZZO, R. The class imbalance problem in deep learning. Mach Learn, v. 113, 2022.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. [S.l.]: MIT Press, 2018. v. 19. 3 p.

GUPTA GAURAV KHATRI, C. S. G. Artificial Neural Networks. [S.l.]: Pratibodh, 2025. 5 p.

GUSTINELI, M. A survey on recently proposed activation functions for Deep Learning. 2022. Disponível em: <https://arxiv.org/abs/2204.02921>.

HAN, S.-H.; KIM, K. W.; KIM, S.; YOUN, Y. C. Artificial neural network: Understanding the basic concepts without mathematics. DND, p. 7, 2019.

HE, J.; LI, L.; XU, J.; ZHENG, C. Relu deep neural networks and linear finite elements. Journal of Computational Mathematics, JSTOR, v. 38, n. 3, p. 502–527, 2020.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. Neural Computation, v. 9, n. 8, p. 1735–1780, 1997.

HU, E. J.; SHEN, Y.; WALLIS, P.; ALLEN-ZHU, Z.; LI, Y.; WANG, S.; WANG, L.; CHEN, W. et al. Lora: Low-rank adaptation of large language models. Iclr, v. 1, n. 2, p. 3, 2022.

HUSSAIN, M.; NASEER, M. Comparative analysis of logistic regression, lstm, and bi-lstm models for sentiment analysis on imdb movie reviews. Journal of Artificial Intelligenceand Computing(JAIC), v. 2, 2023.

IBM. O que é uma rede neural recorrente (RNN)? 2024. <https://www.ibm.com/br-pt/think/topics/recurrent-neural-networks>. Accessed: 2025-07-14.

IBM. What is NLP? 2024. <https://www.ibm.com/think/topics/natural-language-processinghttps://www.ibm.com/think/topics/natural-language-processing>. Accessed: 2025-07-27.

IBM. What is a transformer model? 2025. <https://www.ibm.com/think/topics/transformer-model#Why+are+transformer+models+important%3F>. Accessed: 2025-08-02.

INCA, I. N. do C. O que é câncer? 2022. <https://www.gov.br/inca/pt-br/assuntos/cancer/o-que-e-cancer>. Acessado em 23 de junho de 2025.

JABBAR, H.; KHAN, R. Z. Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). Computer science, communication and instrumentation devices, Res. Publ Singapore, v. 70, n. 10.3850, p. 978–981, 2015.

JOSHI, A.; DABRE, R.; KANOJIA, D.; LI, Z.; ZHAN, H.; HAFFARI, G.; DIPPOLD, D. Natural language processing for dialects of a language: A survey. ACM, v. 57, 2025.

KHAN, I. U.; AFZAL, S.; LEE, J. W. Human activity recognition via hybrid deep learning based model. Sensors, v. 22, p. 16, 2022.

KIM, H. E.; COSA-LINAN, A.; SANTHANAM, N.; JANNESARI, M.; MAROS, M. E.; GANSLANDT, T. Transfer learning for medical image classification: a literature review. BMC medical imaging, Springer, v. 22, n. 1, p. 69, 2022.

KOROTEEV, M. V. BERT: A Review of Applications in Natural Language Processing and Understanding. 2021. Disponível em: <https://arxiv.org/abs/2103.11943>.

KOZMA, L.; VODERHOLZER, J. Theoretical analysis of byte-pair encoding. arXiv preprint arXiv:2411.08671, 2024.

LECUN YOSHUA BENGIO, G. H. Y. Deep Learning. [S.l.]: Nature, 2015. 10 p.

LEE, M. Gelu activation function in deep learning: a comprehensive mathematical analysis and performance. arXiv preprint arXiv:2305.12073, 2023.

LI, Z.; LIU, F.; YANG, W.; PENG, S.; ZHOU, J. A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, IEEE, v. 33, n. 12, p. 6999–7019, 2021.

LIANG, C. X.; BI, Z.; WANG, T.; LIU, M.; SONG, X.; ZHANG, Y.; SONG, J.; NIU, Q.; PENG, B.; CHEN, K. et al. Low-rank adaptation for scalable large language models: A comprehensive survey. Authorea Preprints, Authorea, 2025.

MA, L.; ZHANG, Y. Using word2vec to process big text data. In: IEEE. 2015 IEEE international conference on big data (Big Data). [S.l.], 2015. p. 2895–2897.

MALASHIN, I.; TYNCHENKO, V.; GANTIMUROV, A.; NELYUB, V.; BORODULIN, A. Applications of long short-term memory (lstm) networks in polymeric sciences: A review. Energy Reports, v. 18, 2024.

MDUMA, N. Data balancing techniques for predicting student dropout using machine learning. Data, v. 8, n. 3, 2023. ISSN 2306-5729. Disponível em: <https://www.mdpi.com/2306-5729/8/3/49>.

MINAEE, S.; MIKOLOV, T.; NIKZAD, N.; CHENAGHLU, M.; SOCHER, R.; AMATRIAIN, X.; GAO, J. Large language models: A survey. arXiv preprint arXiv:2402.06196, 2024.

MIRDAN, A. S.; BUYRUKOğLU, S.; BAKER, M. R. Advanced deep learning techniques for sentiment analysis: combining bi- lstm, cnn, and attention layers. International Journal of Advances in Intelligent Informatics, v. 11, p. 18, 2025.

NIELSEN, M. Neural Networks and Deep Learning. [S.l.: s.n.], 2015.

OMS. Palliative Care. 2020. <https://www-who-int.translate.goog/news-room/fact-sheets/detail/palliative-care?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt&_x_tr_pto=tc>. Accessed: 2025-05-29.

OPAS, O. P.-A. da S. Câncer. 2025. <https://www.paho.org/pt/topicos/cancer>. Acessado em 23 de junho de 2025.

ORCHI, I. H.; TABASSUM, N.; HOSSAIN, J.; ALAM, I.; TAJRIN, S. Mental Health Insights: Vulnerable Cancer Patients. Kaggle, 2023. Disponível em: <https://www.kaggle.com/dsv/7284561>.

OROZCO-ARIAS, S.; PIÑA, J. S.; TABARES-SOTO, R.; CASTILLO-OSSA, L. F.; GUYOT, R.; ISAZA, G. Measuring performance metrics of machine learning algorithms for detecting and classifying transposable elements. Processes, MDPI, v. 8, n. 6, p. 638, 2020.

O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458, 2015.

PRADIPTA, G. A.; WARDOYO, R.; MUSDHOLIFAH, A.; SANJAYA, I. N. H.; ISMAIL, M. Smote for handling imbalanced data problem : A review. In: 2021 Sixth International Conference on Informatics and Computing (ICIC). [S.l.: s.n.], 2021. p. 1–8.

RAMACHANDRAN, P.; ZOPH, B.; LE, Q. V. Searching for activation functions. arXiv preprint arXiv:1710.05941, 2017.

SCHMIDT, R. M. Recurrent neural networks (rnns): A gentle introduction and overview. CoRR, abs/1912.05911, 2019.

SINGH, D.; BARVE, S. S.; DWIVEDI, A. K. Optiasar: Optimized aspect-based sentiment analysis of reviews with bilstm-gru and ner-bert in healthcare decision-making. IEEE, v. 13, p. 16, 2025.

SONAWANE, A.; SHINDE, S. Sentiment analysis for social media: Using natural language processing to understand public opinion. IJSRSET, v. 12, p. 4, 2025.

SU, J.; AHMED, M.; LU, Y.; PAN, S.; BO, W.; LIU, Y. Roformer: Enhanced transformer with rotary position embedding. Neurocomputing, Elsevier, v. 568, p. 127063, 2024.

TABASSUM, A.; PATIL, R. R. A survey on text pre-processing & feature extraction techniques in natural language processing. International Research Journal of Engineering and Technology (IRJET), v. 7, n. 06, p. 4864–4867, 2020.

TAN, C.; SUN, F.; KONG, T.; ZHANG, W.; YANG, C.; LIU, C. A survey on deep transfer learning. In: SPRINGER. International conference on artificial neural networks. [S.l.], 2018. p. 270–279.

TERVEN1, J.; CORDOVA-ESPARZA, D.-M.; ROMERO-GONZáLEZ, J.-A.; RAMíREZ-PEDRAZA, A.; CHáVEZ-URBIOLA, E. A. A comprehensive survey of loss functions and metrics in deep learning. Artificial Intelligence Review, v. 58, p. 172, 2025.

THANAKI, J. Python Natural Language Processing. [S.l.: s.n.], 2017.

TOUVRON, H.; LAVRIL, T.; IZACARD, G.; MARTINET, X.; LACHAUX, M.-A.; LACROIX, T.; ROZIÈRE, B.; GOYAL, N.; HAMBRO, E.; AZHAR, F. et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; POLOSUKHIN, I. Attention Is All You Need. 2023. Disponível em: <https://arxiv.org/abs/1706.03762>.

VILLAVICENCIO, C.; MACROHON; JERISON, J.; INBARAJ; ALPHONSE, X.; JENG; JYH-HORNG; HSIEH; JER-GUANG. Twitter sentiment analysis towards covid-19 vaccines in the philippines using naïve bayes. MDPI, v. 12, p. 16, 2021.

VUJOVIć Željko. Classification model evaluation metrics. (IJACSA)International Journal of Advanced Computer Science and Applications, v. 12, 2021.

WEI, J.; ZOU, K. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. 2019. Disponível em: <https://arxiv.org/abs/1901.11196>.

WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D. A survey of transfer learning. Journal of Big data, Springer, v. 3, n. 1, p. 9, 2016.

WONGVORACHAN, T.; HE, S.; BULUT, O. A comparison of undersampling, oversampling, and smote methods for dealing with imbalanced classification in educational data mining. Information, v. 14, n. 1, 2023. ISSN 2078-2489. Disponível em: <https://www.mdpi.com/2078-2489/14/1/54>.

WU, Y.; JIN, Z.; SHI, C.; LIANG, P.; ZHAN, T. Research on the Application of Deep Learning-based BERT Model in Sentiment Analysis. 2024. Disponível em: <https://arxiv.org/abs/2403.08217>.

XIAO, T.; ZHU, J. Introduction to Transformers: an NLP Perspective. 2023. Disponível em: <https://arxiv.org/abs/2311.17633>.

XING, J.; LUO, D.; XUE, C.; XING, R. Comparative analysis of pooling mechanisms in llms: A sentiment analysis perspective. arXiv preprint arXiv:2411.14654, 2024.

YACIM, J.; BOSHOFF, D. Impact of artificial neural networks training algorithms on accurate prediction of property values. Journal of Real Estate Research, v. 40, p. 375–418, 11 2018.

YANG, M.; CHEN, J.; TAO, J.; ZHANG, Y.; LIU, J.; ZHANG, J.; MA, Q.; VERMA, H.; ZHANG, R.; ZHOU, M. et al. Low-rank adaptation for foundation models: A comprehensive review. arXiv preprint arXiv:2501.00365, 2024.

YAQOOB, A.; AZIZ, R. M.; VERMA, N. K. Applications and techniques of machine learning in cancer classification: A systematic review. Human-Centric Intelligent Systems, v. 3, p. 28, 2023.

YAZDANI, A.; SHAMLOO, M.; KHAKI, M.; NAHVIJOU4, A. Use of sentiment analysis for capturing hospitalized cancer patients' experience from free-text comments in the persian language. BMC, v. 23, p. 14, 2023.

YING, X. An Overview of Overfitting and its Solutions. [S.l.]: Joournal of Physics, 2019. 7 p.

YUAN, Z.; WU, S.; WU, F.; LIU, J.; HUANG, Y. Domain attention model for multi-domain sentiment classification. ScienceDirect, v. 15, p. 10, 2018.

YUE, C.; LI, A. Dynamic domain information modulation algorithm for multi-domain sentiment analysis. arXiv, p. 17, 2025.

ZERKOUK, M.; MIHOUBI, M.; CHIKHAOUI, B. Contextual attention-based multimodal fusion of llm and cnn for sentiment analysis. arXiv preprint arXiv:2508.13196, 2025.

Declaração

Durante a preparação deste documento, eu, **Pedro Morais Fernandes**, estudante de **graduação** do **curso de Ciência da Computação**, declaro o uso de IAGen **Gemini e ChatGPT**, versão **Pro e Plus respectivamente**, para **aumento de dados de treino, revisão de texto e criação de figuras**.

Após o uso desta ferramenta/modelo/serviço, revisei e editei o conteúdo em conformidade com os princípios éticos para uso de IAGen (Resolução CONPEP 144) e com os acordos estabelecidos com a pessoa orientadora da pesquisa. Dessa forma, assumo total responsabilidade pelo conteúdo da publicação.

A IAGen foi utilizada nas seguintes etapas:

- Etapa 1: [Aumento de dados de treino]. [Uso de *data augmentation* no conjunto de treino para tentativa de aumento da generalização do modelo]. O modelo utilizado foi o **ChatGPT 5.1**. O *prompt* utilizado foi:

  > *Enhance the dataset contained in this CSV file by paraphrasing and slightly diversifying the texts in the 'posts' column, focusing only on rows where the 'predicted' column has the classes 'positive'. The goal is to generate additional variations that help balance the dataset across classes and improve model generalization during training. When paraphrasing, preserve the original semantic meaning but introduce linguistic diversity through changes in structure, vocabulary, and tone. After generating these new samples, append them to the original dataset and output a new CSV file containing both the original and enriched data. Keep the structure of the file, just add new data. Be careful with overfitting; the model needs better generalization, so use a high variety of words but maintain the meaning of phrases. For each phrase in these examples, generate only one paraphrasing data point.*

- Etapa 2: [Revisão de texto]. [Revisão de texto visando erros de português e correção de linguagem informal usada no texto]. O modelo utilizado foi o **Gemini Pro**. O *prompt* utilizado foi:

  > *Revise o texto, corrigindo erros de português e qualquer uso de linguagem informal que não se enquadre com um trabalho científico*

- Etapa 3: [Criação de figuras dos modelos]. [Criação de figuras a partir de uma figura ou código estabelecido]. O modelo utilizado foi o **Gemini Pro**. O *prompt* utilizado foi:

  > *Crie uma figura que demonstre a arquitetura do modelo utilizado neste código*

- Etapa 4: [Criação de figuras dos modelos]. [Criação de figuras a partir de uma figura ou código estabelecido]. O modelo utilizado foi o **Gemini Pro**. O *prompt* utilizado foi:

  Adapte esta figura, traduzindo o texto dela para o português.