UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DeB3RTa: Um modelo de linguagem pré-treinado para análise de textos financeiros em português usando domínios mistos

Higo Felipe Silva Pires

São Luís — MA
2025

Higo Felipe Silva Pires

DeB3RTa: Um modelo de linguagem pré-treinado para análise de textos financeiros em português usando domínios mistos

Tese de Doutorado submetida à Coordenação do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Maranhão (UFMA) como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica na área de concentração de Ciência da Computação.

Orientador: Prof. Dr. Vicente Leonardo Paucar Casas

Programa de Pós-Graduação em Engenharia Elétrica

Universidade Federal do Maranhão

Orientador: Prof. Dr. Vicente Leonardo Paucar Casas

Coorientador: Prof. Dr. João Paulo Baptista de Carvalho

São Luís — MA

2025

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Pires, Higo Felipe Silva.
DeB3RTa : um modelo de linguagem pré-treinado para análise de textos financeiros em português usando domínios mistos / Higo Felipe Silva Pires. - 2025.
165 f.

Coorientador(a) 1: João Paulo Baptista de Carvalho.
Orientador(a): Vicente Leonardo Paucar Casas.
Tese (Doutorado) - Programa de Pós-graduação em Engenharia Elétrica/ccet, Universidade Federal do Maranhão, São Luís, 2025.

1. Processamento de Linguagem Natural Em Português. 2. Processamento de Linguagem Financeira. 3. Transformers. 4. Pré-treinamento de Domínio Misto. 5. Arquitetura Deberta. I. Carvalho, João Paulo Baptista de. II. Casas, Vicente Leonardo Paucar. III. Título.

Higo Felipe Silva Pires

DeB3RTa: Um modelo de linguagem pré-treinado para análise de textos financeiros em português usando domínios mistos

Trabalho de Tese. São Luís — MA, 16 de Abril de 2025.

Prof. Dr. Vicente Leonardo Paucar Casas
Orientador
Universidade Federal do Maranhão

Prof. Dr. João Paulo Baptista de Carvalho
Coorientador
Instituto Superior Técnico, Universidade de Lisboa (Portugal)

Prof. Dr. Roberto Célio Limão de Oliveira, UFPA
Membro da Banca Examinadora

Prof. Dr. Marcos César da Rocha Seruffo, UFPA
Membro da Banca Examinadora

Prof. Dr. João Viana da Fonseca Neto, UFMA
Membro da Banca Examinadora

Prof. Dr. Denivaldo Cícero Pavão Lopes, UFMA
Membro da Banca Examinadora

Ad Iesum per Mariam.

Agradecimentos

A Deus, princípio e fim de toda a Sabedoria e toda a Ciência.

Aos meus pais, Rosa e Simião, por terem investido suas vidas na minha educação formal.

À minha esposa, Aline, pelo enorme apoio e pela paciência maior ainda durante esta etapa de nossas vidas.

Ao João Francisco, minha maior motivação para finalizar este trabalho.

Ao meu orientador, Prof. Dr. Vicente Leonardo Paucar Casas, pela orientação, dedicação e ajuda nos mais necessários momentos. A este mestre, minha eterna gratidão.

Ao meu coorientador, Prof. Dr. João Paulo Baptista de Carvalho, pela coorientação e apoio dado antes, durante e depois da minha estadia no INESC-ID, sem os quais este trabalho teria sido impossível.

Ao Instituto Federal de Educação, Ciência e Tecnologia do Maranhão (IFMA) e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

"Os limites da minha linguagem são os limites do meu mundo."

Ludwig Wittgenstein (1889-1951)

Resumo

A terminologia complexa e especializada da linguagem financeira nos mercados de língua portuguesa cria desafios significativos para as aplicações de processamento de linguagem natural (PLN), que devem capturar informações linguísticas e contextuais diferenciadas para apoiar análises e tomadas de decisão precisas. Este trabalho apresenta o DeB3RTa, um modelo baseado na arquitetura *Transformer* desenvolvido especificamente por meio de uma estratégia de pré-treinamento de domínio misto que combina *corpora* extensos de finanças, política, administração de negócios e contabilidade para permitir uma compreensão diferenciada da linguagem financeira. O DeB3RTa foi avaliado em comparação com modelos proeminentes — incluindo BERTimbau, XLM-RoBERTa, SEC-BERT, BusinessBERT e variantes baseadas em GPT — e obteve consistentemente ganhos significativos nos principais benchmarks de PLN financeiro. Para maximizar a adaptabilidade e a precisão, o DeB3RTa integra técnicas avançadas de *fine-tuning*, como reinicialização de camadas, regularização de *Mixout*, média estocástica de pesos e decaimento da taxa de aprendizado por camada, que, juntas, melhoram seu desempenho em tarefas de PLN variadas e de grande importância. Essas descobertas ressaltam a eficácia do pré-treinamento de domínio misto na criação de modelos de linguagem de alto desempenho para aplicações especializadas. Com seu desempenho robusto em tarefas analíticas e de classificação complexas, o DeB3RTa oferece uma ferramenta poderosa para o avanço da PLN no setor financeiro e para atender às necessidades de processamento de linguagem diferenciada em contextos de língua portuguesa.

Palavras-chave: Processamento de Linguagem Natural em Português; Processamento de Linguagem Financeira; Transformers; Pré-treinamento de Domínio Misto; Decaimento da Taxa de Aprendizado por Camada; Regularização de Mixout; Reinicialização de Camadas; Arquitetura DeBERTa; Classificação de Textos Financeiros.

Abstract

The complex and specialized terminology of financial language in Portuguese-speaking markets create significant challenges for natural language processing (NLP) applications, which must capture nuanced linguistic and contextual information to support accurate analysis and decision-making. This paper presents DeB3RTa, a transformer-based model specifically developed through a mixed-domain pretraining strategy that combines extensive corpora from finance, politics, business management, and accounting to enable a nuanced understanding of financial language. DeB3RTa was evaluated against prominent models—including BERTimbau, XLM-RoBERTa, SEC-BERT, BusinessBERT, and GPT-based variants — and consistently achieved significant gains across key financial NLP benchmarks. To maximize adaptability and accuracy, DeB3RTa integrates advanced fine-tuning techniques such as layer reinitialization, Mixout regularization, stochastic weight averaging, and layer-wise learning rate decay, which together enhance its performance across varied and high-stakes NLP tasks. These findings underscore the efficacy of mixed-domain pretraining in building high-performance language models for specialized applications. With its robust performance in complex analytical and classification tasks, DeB3RTa offers a powerful tool for advancing NLP in the financial sector and supporting nuanced language processing needs in Portuguese-speaking contexts.

Keywords: Portuguese Natural Language Processing; Financial Language Processing; Transformers; Mixed Domain Pre-training; Layer-wise Learning Rate Decay; Mixout Regularization; Layer Resetting; DeBERTa Architecture; Financial Text Classification.

Lista de ilustrações

*

Lista de tabelas

*

Lista de abreviaturas e siglas

ALBERT	*A Lite BERT*
API	*Application Programming Interface*
ASSIN	*Avaliação de Similaridade Semântica e INferência textual*
BBRC	*Brazilian Banking Regulation Corpora*
BERT	*Bidirectional Encoder Representations from Transformers*
BI-RADS	*Breast Imaging-Reporting and Data System*
BoW	*Bag-of-Words*
BPE	*Byte Pair Encoding*
BrWaC	*Brazilian Web as Corpus*
CBOW	*Continuous Bag-of-Words*
CNN	*Convolutional Neural Network*
DBN	*Deep Belief Network*
DeBERTa	*Decoding-enhanced BERT with Disentangled Attention*
FP16	*Formato de ponto flutuante de 16 bits*
GAN	*Generative Adversarial Network*
GLUE	*General Language Understanding Evaluation*
GPT	*Generative Pre-trained Transformer*
GPS	*General Problem Solver*
GPU	*Graphics Processing Unit*
GRU	*Gated Recurrent Unit*
IA	*Inteligência Artificial*
LLM	*Large Language Model*
LLRD	*Layer-wise Learning Rate Decay*
LSTM	*Long Short-Term Memory*

MADGRAD	*Momentumized, Adaptive, Dual Averaged GRADient*
MAR	*Modelo Autoregressivo*
mBERT	*Multilingual BERT*
MLM	*Masked Language Modeling*
MLQA	*Multilingual Question Answering*
MNAR	*Modelo Não Autoregressivo*
MNLI	*Multi-Genre Natural Language Inference*
NER	*Named Entity Recognition*
NLLLoss	*Negative Log-Likelihood Loss*
NLU	*Natural Language Understanding*
NSP	*Next Sentence Prediction*
OOV	*Out-of-Vocabulary*
PLM	*Pretrained Language Model*
PLN	*Processamento de Linguagem Natural*
PMI	*Pointwise Mutual Information*
PR-AUC	*Area Under Precision-Recall Curve*
RACE	*ReAding Comprehension dataset from Examinations*
RBM	*Restricted Boltzmann Machine*
RLHF	*Reinforcement Learning from Human Feedback*
RNA	*Rede Neural Artificial*
RNN	*Recurrent Neural Network*
RoBERTa	*Robustly Optimized BERT Approach*
RTD	*Replaced Token Detection*
SEC	*Securities and Exchange Commission*
SQuAD	*Stanford Question Answering Dataset*
SVM	*Support Vector Machine*

SWA	*Stochastic Weight Averaging*
SWM	*Subword Word Masking*
TF-IDF	*Term Frequency-Inverse Document Frequency*
TPU	*Tensor Processing Unit*
UFMA	*Universidade Federal do Maranhão*
V2X	*Vehicle-to-Everything*
VAE	*Variational Autoencoder*
WWM	*Whole Word Masking*
XBRL	*eXtensible Business Reporting Language*
XLM-R	*XLM-RoBERTa*
XLM-RoBERTa	*Cross-lingual Language Model RoBERTa*
XNLI	*Cross-lingual Natural Language Inference*

Sumário

*

1 Introdução

1.1 Contextualização e Problemática

Nos últimos anos, o surgimento de modelos de inteligência artificial mudou significativamente o campo do PLN, permitindo técnicas avançadas de aprendizado de máquina que podem modelar a linguagem com escala e profundidade notáveis. Essa evolução não apenas aprimorou nossa capacidade de processar e analisar dados de linguagem natural, mas também abriu caminho para aplicativos especializados em vários domínios. Um exemplo notável é o *Bidirectional Encoder Representations from Transformers* (BERT) [1], que serve como um componente fundamental para o desenvolvimento de modelos de PLN específicos de tarefas.

A arquitetura do BERT, que permite a consideração do processamento de contexto bidirecional, revolucionou a forma como as máquinas entendem a linguagem. Ao utilizar a modelagem de linguagem mascarada e as tarefas de previsão da próxima frase durante o treinamento, o BERT captura relações complexas entre palavras e frases, levando a um melhor desempenho em vários benchmarks de PLN [2]. Esse recurso fez do BERT uma referência em tarefas como análise de sentimentos, reconhecimento de entidades nomeadas (*named entity recognition* — NER) e resposta a perguntas (*question answering)*, demonstrando sua versatilidade e eficácia em aplicações do mundo real [3].

À medida que o cenário do PLN continua a evoluir, os modelos autorregressivos causais, especialmente os da família *Generative Pretrained Transformer* (GPT), fizeram avanços substanciais ao se concentrarem na geração de texto e em tarefas preditivas. Modelos como GPT-2 [4], GPT-3 [5] e GPT-4 [6] exemplificam essa abordagem, utilizando o contexto de *tokens* gerados anteriormente para prever palavras subsequentes em uma sequência. Essa metodologia autorregressiva causal contrasta com os modelos bidirecionais, como o BERT, que utilizam *tokens* passados e futuros para criar representações contextuais. O sucesso dos modelos GPT em diversas aplicações, inclusive em solicitações de disparo zero e de poucos disparos, ressalta sua eficácia na geração de textos coerentes e contextualmente relevantes.

O sucesso impressionante dos modelos baseados em *Transformers* levou ao desenvolvimento de modelos especializados adaptados a setores específicos. Na esteira deste desenvolvimento, pesquisadores desenvolveram vários modelos específicos de domínio para tarefas relacionadas ao domínio financeiro [7], [8], [9], [10], [11], [12], [13]. Expandir a disponibilidade de modelos baseados em *Transformers* em vários idiomas é fundamental para maximizar sua eficácia e acessibilidade. No entanto, o desenvolvimento de modelos

de linguagem financeira adaptados para outros idiomas que não o inglês apresenta desafios significativos, especialmente devido à escassez de *datasets* de alta qualidade e em grande escala no setor financeiro, o que cria barreiras para alcançar o desempenho ideal em contextos financeiros que não sejam em inglês.

1.2 Motivação

Nos últimos anos, o setor financeiro tem experimentado um aumento significativo no volume e na complexidade das informações digitais, incluindo documentos, relatórios e artigos de notícias que são essenciais para a tomada de decisões em investimentos, gerenciamento de riscos e formulação de políticas. Apesar desse crescimento, os modelos atuais de processamento de linguagem natural enfrentam desafios para interpretar com precisão as nuances da linguagem financeira, principalmente em idiomas pouco representados, como o português. Essa dificuldade é exacerbada pelo jargão específico do domínio, pela terminologia mista de campos de estudo relacionados e pela falta de *datasets* anotados em grande escala. Para aprimorar a classificação de textos financeiros e possibilitar a análise em tempo real, é fundamental desenvolver modelos de linguagem específicos do domínio que melhorem o desempenho em contextos de tomada de decisões financeiras, abordando de forma eficaz esses desafios [14], [15], [16].

Quando um domínio especializado não está disponível para um domínio específico, os *pretrained language models* (modelos de linguagem pré-treinados — PLM) de domínio geral que suportam o português, incluindo modelos multilíngues como XLM-RoBERTa [17] ou modelos de domínio geral português como BERTimbau [18], são comumente usados como soluções alternativas. Há também a possibilidade de usar a transferência entre idiomas (*cross-lingual transfer*), que é uma técnica que permite que modelos treinados em um idioma sejam aplicados a outro idioma, geralmente sem treinamento adicional. No entanto, essas alternativas podem não capturar totalmente as nuances de domínios especializados, como o financeiro. Em resposta a essa limitação, desenvolvemos um modelo especializado de linguagem financeira adaptado para o português, projetado para se destacar em tarefas de processamento de linguagem financeira, oferecendo uma solução precisa e eficaz.

A criação desses PLMs especializados é essencial não apenas para aprimorar o desempenho, mas também para garantir que essas tecnologias sejam inclusivas e representativas de diversas comunidades linguísticas. Ao se concentrar nas características exclusivas de diferentes domínios de conhecimento, os pesquisadores podem desenvolver ferramentas mais eficazes que atendam às necessidades específicas dos usuários. Por exemplo, as instituições financeiras que operam em países de língua portuguesa podem se beneficiar de modelos personalizados que entendam as terminologias e as estruturas regulatórias locais.

1.3 Hipótese e Objetivos

A crescente complexidade e especialização da linguagem financeira em mercados de língua portuguesa apresenta desafios únicos para o desenvolvimento de modelos de processamento de linguagem natural que possam compreender nuances linguísticas e contextuais, essenciais para apoiar análises e tomadas de decisão em cenários críticos. Partindo dessa premissa, este trabalho propõe investigar a seguinte hipótese: *O uso de estratégias de pré-treinamento de domínio misto, exemplificado pelo modelo DeB3RTa, é particularmente adequado para capturar a terminologia altamente especializada e os contextos linguísticos diferenciados da linguagem financeira em português. Além disso, o emprego dessas estratégias, combinado com técnicas avançadas de fine-tuning, favorece o desenvolvimento de soluções de PLN que oferecem desempenho computacional superior, maior precisão analítica, adaptabilidade a diferentes cenários de aplicação, escalabilidade eficiente e a capacidade de lidar com tarefas variadas e complexas em tempo hábil. Essas características tornam modelos como o DeB3RTa ferramentas promissoras para avançar o estado da arte em PLN aplicado ao setor financeiro, proporcionando melhorias mensuráveis em benchmarks de desempenho e atendendo às necessidades específicas do setor em termos de processamento de linguagem especializada.*

De maneira a comprovar a veracidade da hipótese levantada, este trabalho de tese estabelece objetivos gerais e específicos que orientam sua condução e fundamentam suas contribuições científicas.

O objetivo geral desta pesquisa é propor um modelo baseado na arquitetura *Transformer* para o processamento de linguagem natural no contexto financeiro de língua portuguesa, buscando alcançar a melhor adequação possível às características e demandas linguísticas específicas do setor financeiro. Tal modelo visa capturar com precisão a terminologia especializada e os contextos semânticos únicos desse domínio, promovendo soluções otimizadas que aliem desempenho computacional robusto, adaptabilidade a cenários diversos e escalabilidade eficiente.

Para viabilizar a realização do objetivo geral desta tese, foram definidos os seguintes objetivos específicos:

- Projetar, desenvolver e implementar um modelo baseado em *Transformers*, mais especificamente no modelo DeBERTa, voltado para o processamento de linguagem natural no contexto financeiro de língua portuguesa, considerando as especificidades terminológicas e contextuais desse domínio;
- Criar um ambiente computacional que possibilite a análise do desempenho do modelo, através de *benchmarks* obtidos através de tarefas de processamento de linguagem natural executadas em bases de dados relacionadas ao domínio do modelo;

- Realizar experimentos comparativos que analisem o desempenho do DeB3RTa em relação a outros modelos relevantes, como BERTimbau, XLM-RoBERTa e variantes de modelos GPT, utilizando a métrica F1-score em diferentes tarefas de PLN;
- Examinar e validar as técnicas avançadas de *fine-tuning* aplicadas ao DeB3RTa, incluindo reinicialização de camadas, regularização *Mixout* e decaimento da taxa de aprendizado por camadas, avaliando seu impacto na adaptabilidade, escalabilidade e desempenho geral em tarefas analíticas e classificatórias complexas no setor financeiro.

1.4 Metodologia de Pesquisa

Para êxito dos objetivos do trabalho, este trabalho seguirá a seguinte metodologia:

1. Revisão bibliográfica, feita mediante levantamento de diversas fontes, como sites Web, artigos de jornais e conferências e pré-publicações;
2. Definição do escopo do problema, de maneira a traçar limites claros a respeito do que deve ser empreendido durante o trabalho;
3. Levantamento dos requisitos necessários para elaboração do modelo proposto;
4. Definição e criação do *corpus* a ser utilizado no treinamento do modelo proposto;
5. Definição dos modelos do estado-da-arte (*baseline*) com os quais o modelo proposto será comparado;
6. Definição dos datasets a serem utilizados nas tarefas de treinamento do modelo proposto e dos modelos da *baseline*;
7. Definição dos requisitos relativos ao *benchmarking* do modelo proposto e dos modelos da *baseline*;
8. Execução do treinamento do modelo proposto e dos modelos da *baseline* com os datasets propostos;
9. Coleta e posterior avaliação dos resultados do treinamento.

1.5 Organização do Trabalho

Este trabalho está estruturado da seguinte forma:

- O Capítulo 2 apresenta um contexto tecnológico que aborda os fundamentos teóricos necessários para a compreensão deste trabalho. Nele, são discutidos os

conceitos relacionados a Inteligência Artificial, Processamento de Linguagem Natural, Aprendizagem Profunda, arquitetura *Transformer*, aprendizado por transferência, processamento de linguagem natural aplicado ao domínio financeiro, modelos de linguagem pré-treinados, Inteligência Artificial Generativa, modelos de linguagem autoregressivos, Grandes Modelos de Linguagem (LLMs) e técnicas de otimização e regularização de modelos de linguagem. Por fim, é apresentada uma síntese do capítulo;

- O Capítulo 3 apresenta uma análise abrangente do estado da arte, contemplando uma revisão crítica e aprofundada das contribuições científicas mais significativas e contemporâneas relacionadas aos conceitos previamente abordados no Capítulo 2. Esta análise é complementada por um estudo comparativo entre as obras consideradas mais relevantes para o presente trabalho, culminando em uma síntese que consolida os principais aspectos discutidos;

- O Capítulo 4 apresenta a contribuição central desta tese: um modelo fundamentado na arquitetura *Transformer* desenvolvido especificamente para o processamento de linguagem natural no domínio financeiro em português. O capítulo detalha sistematicamente a estruturação arquitetural do modelo proposto, sua fundamentação metodológica e implementação prototípica, incluindo uma demonstração abrangente do modelo em um cenário operacional. A exposição culmina com uma síntese que consolida os elementos fundamentais da proposta apresentada;

- O Capítulo 5 apresenta uma análise sistemática dos resultados obtidos na avaliação do modelo proposto. O Capítulo contempla uma avaliação comparativa abrangente entre o modelo desenvolvido e os trabalhos correlatos identificados na literatura, seguida por uma discussão aprofundada das implicações dos resultados. A exposição é concluída com uma síntese analítica dos testes realizados, consolidando as principais descobertas e contribuições observadas;

- O Capítulo 6 apresenta as conclusões desta tese, oferecendo uma análise crítica dos objetivos propostos e seus respectivos níveis de consecução. O capítulo contempla uma discussão pormenorizada das limitações identificadas no modelo desenvolvido, bem como uma compilação das publicações acadêmicas derivadas desta pesquisa. Adicionalmente, são delineadas perspectivas para investigações futuras, visando o aprimoramento e a expansão do trabalho realizado.

2 Contexto Tecnológico

Este capítulo apresenta os conceitos explorados para o desenvolvimento do estudo, tais como conceitos referentes a Inteligência Artificial, Aprendizagem Profunda, Processamento de Linguagem Natural, arquitetura *Transformer*, aprendizado por transferência, processamento de linguagem natural aplicado ao domínio financeiro, modelos de linguagem pré-treinados, Inteligência Artificial Generativa, modelos de linguagem autoregressivos, Grandes Modelos de Linguagem (LLMs) e técnicas de otimização e regularização de modelos de linguagem.

2.1 Inteligência Artificial

A IA é um domínio multifacetado que busca criar sistemas capazes de realizar tarefas que normalmente exigem inteligência humana. Essas tarefas incluem raciocínio, aprendizado, solução de problemas, percepção e compreensão da linguagem. O termo "inteligência artificial" foi cunhado pela primeira vez em 1956, durante a Conferência de *Dartmouth*, que é frequentemente considerada como o nascimento da IA como um campo de estudo. Desde sua criação, a IA passou por transformações significativas, evoluindo de sistemas simples baseados em regras para algoritmos complexos de aprendizado de máquina capazes de aprendizado profundo e tarefas generativas [19], [20].

2.1.1 Contexto Histórico e Origens

As origens da IA remontam à metade do século XX, com o trabalho fundamental de pioneiros como Alan Turing, que propôs o Teste de Turing como uma medida de inteligência de máquina. O trabalho de Turing lançou as bases para futuros desenvolvimentos em teoria computacional e redes neurais artificiais [19], [20]. Os primeiros anos da Inteligência Artificial foram marcados pelo desenvolvimento da IA simbólica, em que os pesquisadores se concentraram na criação de sistemas capazes de manipular símbolos para resolver problemas. Essa primeira fase envolveu sistemas baseados em regras, que dependiam de regras predefinidas para tomar decisões. Esses sistemas eram limitados em sua capacidade de se adaptar a novas informações ou aprender com a experiência [19], [21]. Essa abordagem foi exemplificada por programas pioneiros, como o *Logic Theorist* [22] e o *General Problem Solver* (GPS) [23].

O trabalho de Newell e Simon no *Logic Theorist* e no GPS lançou as bases para o início da IA, introduzindo ferramentas fundamentais como processamento de listas e sistemas de produção. O *Logic Theorist* foi um dos primeiros programas a simular

as habilidades humanas de resolução de problemas, influenciando significativamente o campo da psicologia cognitiva e do processamento de informações e causando um impacto duradouro na psicologia cognitiva, especialmente nas áreas de representação mental e resolução de problemas [24], [25].

À medida que o campo progredia, as limitações da IA simbólica se tornaram aparentes, levando a um declínio no interesse durante as décadas de 1970 e 1980, um período frequentemente chamado de "inverno da IA". No entanto, o ressurgimento do interesse na década de 1990 foi impulsionado pelos avanços no poder computacional, pela disponibilidade de grandes *datasets* (*datasets*) e pelo desenvolvimento de algoritmos mais sofisticados, especialmente em aprendizado de máquina e redes neurais. Essa fase viu o surgimento do aprendizado de máquina, em que os algoritmos podiam aprender com os dados e melhorar seu desempenho ao longo do tempo. Essa mudança foi marcada pela introdução de técnicas de aprendizado supervisionadas e não supervisionadas, que permitiram que os sistemas identificassem padrões e fizessem previsões com base nos dados de entrada [19], [21].

O advento da aprendizagem profunda na metade da década dos anos 1990 marcou um ponto de virada significativo no desenvolvimento da IA. A aprendizagem profunda, um subconjunto da aprendizagem de máquina, utiliza redes neurais artificiais com várias camadas para processar grandes quantidades de dados. Essa abordagem levou a avanços em várias aplicações, incluindo reconhecimento de imagens, processamento de linguagem natural e sistemas autônomos. O sucesso da aprendizagem profunda foi atribuído à disponibilidade de grandes *datasets*, avanços em hardware e algoritmos aprimorados, permitindo que os sistemas de IA alcancem desempenho de nível humano em tarefas específicas [19], [21].

2.1.2 Àreas de Aplicação

A IA permeou vários domínios, revolucionando os setores e aprimorando os recursos em vários campos. As principais áreas de aplicação incluem:

2.1.2.1 Visão Computacional

A visão computacional é uma área proeminente da IA que se concentra em permitir que as máquinas interpretem e compreendam as informações visuais do mundo. Essa tecnologia encontrou aplicações em reconhecimento facial, veículos autônomos e imagens médicas. Por exemplo, os algoritmos de IA podem analisar imagens médicas para detectar anomalias, ajudando os profissionais de saúde a diagnosticar doenças com maior precisão. A integração da IA na visão computacional melhorou significativamente a eficiência e a eficácia da análise de imagens, levando a avanços em campos como radiologia e patologia [26], [27].

2.1.2.2 Robótica

A integração da IA na robótica tem o potencial de transformar vários setores, impulsionando a inovação e a eficiência. A robótica com IA está revolucionando os setores ao aumentar a produtividade, a precisão e a segurança [28], [29]. Na área da saúde, os sistemas robóticos auxiliam nas cirurgias e no atendimento aos pacientes, reduzindo os erros humanos e melhorando os resultados [30], [31], [32]. Na logística, a IA e a robótica agilizam os processos, reduzem os custos, aprimoram o gerenciamento da cadeia de suprimentos e permitem que os robôs habilitados para IA possam lidar com tarefas como transporte e despacho de materiais, melhorando a eficiência operacional geral [33].

2.1.2.3 Saúde

A inteligência artificial está revolucionando a área da saúde ao aprimorar o diagnóstico, a medicina personalizada e o atendimento ao paciente. Os algoritmos de IA podem analisar grandes quantidades de dados de pacientes, identificar padrões e prever resultados, permitindo que os prestadores de serviços de saúde tomem decisões informadas.

A IA acelera a descoberta de medicamentos ao analisar a literatura biomédica, os dados genômicos e os resultados de ensaios clínicos, identificando possíveis alvos de medicamentos, prevendo a toxicidade dos medicamentos e prevendo o surto de doenças como gripe, zika, ebola, tuberculose e COVID-19, auxiliando no planejamento e na resposta da saúde pública [34], [35].

A IA pode aumentar a eficiência clínica, reduzindo o tempo de trabalho e melhorando a jornada do paciente nas clínicas, demonstrando custo-benefício em comparação com as práticas padrão baseadas em humanos e potencial para transformar a prestação de serviços de saúde e o envolvimento do paciente [36].

A IA possibilita a medicina de precisão por meio da análise de dados genéticos, fenotípicos e clínicos para adaptar os tratamentos a cada paciente, melhorando os resultados e reduzindo os efeitos adversos e facilitando regimes de tratamento personalizados para doenças como o câncer [37], [38], [39].

A IA está causando um impacto profundo na área da saúde, melhorando o diagnóstico de doenças, possibilitando a medicina personalizada, aprimorando a tomada de decisões clínicas e acelerando a descoberta de medicamentos. Esses avanços prometem revolucionar o atendimento ao paciente, tornando-o mais preciso, eficiente e eficaz. No entanto, os desafios relacionados à privacidade dos dados, à ética e à integração da IA na prática clínica devem ser abordados para que seu potencial seja plenamente aproveitado.

2.1.2.4 Finanças

A Inteligência Artificial está revolucionando o setor financeiro ao aprimorar o gerenciamento de riscos, fornecendo avaliações de risco mais precisas e sofisticadas, reduzindo o tempo necessário para identificar e mitigar riscos, detectar fraudes, analisar grandes quantidades de dados em tempo real e identificar padrões e anomalias [40] e tomar decisões de investimento, ajudando as instituições financeiras a tomar decisões de investimento mais precisas por meio do reconhecimento de padrões [41], [42].

A implementação da IA em instituições financeiras oferece inúmeros benefícios, mas também apresenta desafios significativos relacionados à privacidade de dados, segurança e considerações éticas [43], [44]. Para garantir o uso responsável e sustentável da IA, as instituições financeiras devem adotar a governança e os controles adequados, incluindo medidas de proteção de dados [45], conformidade ética [46], adesão regulatória [47] e transparência nos processos de tomada de decisões sobre IA [45]. Equilibrar a inovação com a proteção do consumidor e considerações éticas é crucial para a integração bem-sucedida da IA no setor financeiro.

2.1.2.5 Educação

A Inteligência Artificial está sendo cada vez mais integrada aos contextos educacionais, aprimorando as experiências de aprendizagem personalizadas e a eficiência administrativa. Os sistemas inteligentes de tutoria podem se adaptar aos estilos individuais de aprendizagem, fornecendo conteúdo e *feedback* personalizados e melhorando o envolvimento do aluno e os resultados da aprendizagem. Além disso, a IA pode ajudar os educadores a otimizar os currículos com base nos dados de desempenho e auxiliar no monitoramento e na avaliação contínuos do desempenho dos alunos [48], [49]. A implementação da IA na educação levanta preocupações sobre privacidade, segurança e possíveis vieses, que precisam ser abordados para garantir o uso ético. Garantir a acessibilidade, a transparência e a justiça nos sistemas educacionais baseados em IA é fundamental para sua adoção bem-sucedida [50], [51].

2.1.2.6 Transporte

A IA está revolucionando o setor de transportes ao aprimorar os recursos dos veículos autônomos, em que a IA é fundamental para que os veículos autônomos percebam o ambiente, localizem, mapeiem e tomem decisões [52], [53], otimizem os sistemas de gerenciamento de tráfego [54], [55], aumentem a eficiência energética, reduzam as emissões de gases de efeito estufa e melhorem a autonomia dos veículos [56], [57]. A integração da IA com os sistemas *Vehicle-to-Everything* (V2X) e a IoT em soluções inteligentes de transporte está abrindo caminho para uma mobilidade mais segura, eficiente e ecológica [58], [59]. À medida que as tecnologias de IA continuarem avançando, seu impacto no transporte

provavelmente crescerá, levando a melhorias ainda maiores em segurança, eficiência e sustentabilidade.

2.2 Processamento de Linguagem Natural

O Processamento de Linguagem Natural é um subcampo da inteligência artificial que se concentra na interação entre os computadores e a linguagem humana. Ele permite que as máquinas entendam, interpretem e gerem a linguagem humana, facilitando aplicativos como *chatbots*, assistentes virtuais e serviços de tradução de idiomas.

O PLN fez avanços significativos em várias aplicações, especialmente no atendimento ao cliente e na educação [60], [61], [62]. O desenvolvimento de modelos e técnicas avançados, como o aprendizado baseado em solicitações, aumentou a capacidade da IA de gerar textos coerentes e contextualmente relevantes [63], no entanto, ainda há desafios para melhorar a qualidade do *dataset* e compreender o comportamento do usuário [61].

A evolução do PLN foi significativamente influenciada pelos avanços na aprendizagem de máquina, especialmente na aprendizagem profunda. As técnicas tradicionais de PLN dependiam muito de sistemas baseados em regras e modelos estatísticos, que frequentemente enfrentavam dificuldades com as complexidades e as nuances da linguagem humana. No entanto, a introdução de modelos de aprendizagem profunda, como redes neurais recorrentes e *Transformers*, revolucionou o campo ao permitir representações de linguagem e recursos de processamento mais sofisticados [64]. Esses modelos aproveitam grandes *datasets* para aprender padrões e relacionamentos dentro da linguagem, levando a um melhor desempenho em várias tarefas de PLN.

2.2.1 Técnicas de Processamento de Linguagem Natural

Antes que qualquer tarefa de PLN possa ser executada, é importante que os dados de texto bruto passem por um pré-processamento. Essa etapa é fundamental para limpar e preparar os dados para análise, pois afeta diretamente os resultados das aplicações do PLN. Os requisitos de pré-processamento dependem da natureza do *corpus* e da aplicação específica do PLN e, às vezes, as práticas convencionais precisam ser personalizadas para obter melhores resultados de análise [65].

A tokenização é o processo de dividir o texto em unidades menores chamadas *tokens*, que podem ser palavras, frases ou até mesmo caracteres. Essa etapa é fundamental, pois simplifica o texto e facilita a análise. A tokenização precisa pode melhorar significativamente a precisão da marcação de parte da fala e de outras tarefas posteriores [65].

A stemização (*stemming*) é uma técnica de processamento de linguagem natural usada para reduzir as palavras à sua forma original, removendo sufixos e prefixos. O

principal objetivo da stemização é agrupar diferentes formas de uma palavra para que possam ser analisadas como um único item. Por exemplo, as palavras “correndo”, “corredor” e “correu” podem ser reduzidas ao radical “corr”. Os algoritmos de stemização, como o *Porter Stemmer* [66], normalmente usam um conjunto de regras heurísticas para remover afixos das palavras, no entanto, os radicais resultantes nem sempre são palavras válidas no idioma [67].

A lematização é outra técnica de processamento de linguagem natural que visa reduzir as palavras à sua forma básica ou de dicionário, conhecida como lema. Diferentemente da stemização, a lematização considera o contexto e a análise morfológica das palavras, garantindo que a forma reduzida seja uma palavra válida. Por exemplo, “correndo” seria reduzido para “correr”, e “melhor” seria reduzido para “bom”. A lematização geralmente requer mais conhecimento e recursos linguísticos em comparação com a stemização, mas geralmente produz resultados mais precisos e significativos [67].

Stopwords são palavras comuns, como “um”, “o” e “é”, que têm pouco valor semântico e geralmente são excluídas das operações de indexação e pesquisa para aumentar a eficiência e a acurácia. A remoção de *stopwords* pode reduzir significativamente o tamanho do *corpus* de texto, geralmente em 35 a 45%, diminuindo assim a complexidade de tempo e espaço sem comprometer o desempenho [68]. A Figura 2.1 retrata o resultado da aplicação da remoção de *stopwords* e *stemming* em um pequeno texto.

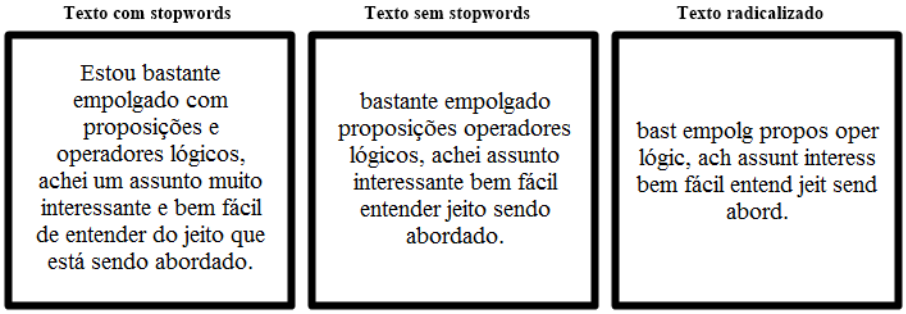

Figura 2.1 – Texto inicial, seguido de remoção de *stopwords* e *stemming*. Extraída de [69].

A normalização do texto envolve a conversão do texto em um formato padrão, o que inclui o uso de letras minúsculas, o tratamento da pontuação e a remoção da formatação. Essa etapa é essencial para garantir a consistência dos dados de texto, o que pode melhorar o desempenho dos modelos de PLN [65]. Por exemplo, no pré-processamento de dados de texto do idioma russo, a normalização foi proposta para padronizar caracteres e pontuação, melhorando assim a consistência e a confiabilidade dos dados [70].

O tratamento da pontuação e a identificação de *multiword expressions* (expressões com várias palavras) também são etapas importantes do pré-processamento. Essas técnicas ajudam a reter o significado semântico do texto, o que é fundamental para tarefas como tradução automática e raciocínio [65]. Por exemplo, na técnica *Bag-of-Words*, o tratamento

da pontuação e a criação de um dicionário mestre são etapas essenciais para a conversão de texto de forma livre em uma representação numérica [71].

A n-gramação envolve a criação de sequências de *n tokens*, que podem capturar o contexto e melhorar o desempenho dos modelos de PLN. Essa técnica é particularmente útil em tarefas como a previsão da próxima frase, em que o contexto das sequências de palavras é crucial. Por exemplo, na análise de sentimentos, a n-gramação tem sido usada para capturar o contexto de *multiword expressions*, melhorando assim a precisão dos modelos [72].

A Tabela 2.1 traz uma descrição sumarizada de cada uma das técnicas acima descritas.

Tabela 2.1 – Técnicas de Processamento de Linguagem Natural

Técnica	**Descrição**
Tokenização	Divisão do texto em unidades menores chamadas *tokens*, como palavras, frases ou caracteres.
Stemização	Redução das palavras à sua forma raiz, removendo sufixos e prefixos.
Lematização	Redução das palavras à sua forma básica (lema), levando em conta o contexto gramatical para garantir que a forma reduzida seja válida.
Remoção de *stopwords*	Exclusão de palavras comuns, como artigos e preposições, que têm pouco valor semântico, para reduzir a complexidade do texto.
Normalização de texto	Padronização do texto, convertendo para minúsculas, removendo formatação e pontuação para garantir consistência.
Tratamento de pontuação	Processamento da pontuação para preservar o significado semântico do texto.
Identificação de *multiword expressions*	Detecção de combinações de palavras que formam expressões compostas e carregam significado específico.
N-gramação	Criação de sequências de *n tokens* consecutivos para capturar o contexto em uma sequência de palavras.

2.3 Aprendizagem Profunda

O núcleo da aprendizagem profunda está no uso de redes neurais artificiais (RNAs), que são modelos computacionais inspirados na arquitetura do cérebro humano. Essas redes consistem em nós interconectados ou neurônios organizados em camadas, incluindo uma

camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada neurônio processa os dados de entrada e passa sua saída para a próxima camada, permitindo que o modelo aprenda representações de dados complexas por meio de técnicas de *backpropagation* e otimização. Essa arquitetura permite que os modelos de aprendizagem profunda se sobressaiam em tarefas que exigem extração e classificação de recursos, muitas vezes superando os algoritmos tradicionais de aprendizagem de máquina.

A aprendizagem profunda revolucionou vários campos, especialmente a visão computacional, empregando RNAs profundas (multicamadas) treinadas com *backpropagation*. Esse método requer grandes quantidades de exemplos de treinamento rotulados, mas alcança uma precisão de classificação impressionante, às vezes superando o desempenho humano [73]. O sucesso da aprendizagem profunda em tarefas como classificação de imagens, reconhecimento de fala e tradução de idiomas ressalta sua função como um importante facilitador da inteligência artificial [74].

A principal distinção entre a aprendizagem profunda e as abordagens tradicionais de aprendizagem de máquina está no processo de extração de recursos. Os métodos tradicionais de aprendizado de máquina, como o SVM e as árvores de decisão, geralmente exigem *feature engineering* (engenharia de recursos) manual, em que os especialistas do domínio identificam os recursos relevantes dos dados. Esse processo pode ser demorado e pode não capturar todas as nuances dos dados, levando a um desempenho abaixo do ideal do modelo.

Em contrapartida, os modelos de aprendizagem profunda aprendem automaticamente os recursos dos dados por meio de suas várias camadas, reduzindo significativamente a necessidade de intervenção manual e permitindo representações mais robustas e generalizadas. Por exemplo, as arquiteturas neurais profundas podem extrair recursos de alto nível automaticamente sem engenharia de recursos artesanal, ao contrário dos algoritmos tradicionais de aprendizado de máquina [73]. Foi demonstrado que esse recurso de extração automática de recursos dos modelos de aprendizagem profunda supera os métodos tradicionais em vários domínios, como análise de imagem e vídeo [74] e processamento de linguagem natural [75].

Além disso, as técnicas de aprendizagem profunda demonstraram desempenho superior em cenários complexos, como a classificação de imagens biológicas, devido a seus mecanismos de extração de recursos incorporados [76]. Também, a aprendizagem profunda foi reconhecida como uma ferramenta poderosa para tratar de problemas não lineares na classificação de imagens hiperespectrais, com os quais os métodos tradicionais de aprendizagem de máquina têm dificuldades devido às características complexas destes tipos de dados [77].

Os modelos de aprendizagem profunda são particularmente adequados para lidar com grandes *datasets*, o que é crucial no atual mundo orientado por dados. À medida

que o volume de dados aumenta, os algoritmos tradicionais de aprendizado de máquina geralmente têm dificuldades para manter o desempenho devido à sua dependência de conjuntos de recursos limitados. A aprendizagem profunda, no entanto, prospera em grandes *datasets*, aproveitando sua arquitetura para aprender com grandes quantidades de informações e melhorar a precisão. Esse recurso fez com que a aprendizagem profunda dominasse campos como o reconhecimento de imagens e de fala, em que grandes *datasets* rotulados estão disponíveis para treinamento.

Vários estudos destacaram as vantagens da aprendizagem profunda no manuseio de grandes *datasets*. Por exemplo, os modelos de aprendizagem profunda revolucionaram campos como visão computacional, processamento de linguagem natural e reconhecimento de fala, utilizando dados rotulados em grande escala para terem seus desempenhos aprimorados [78], [79]. O sucesso da aprendizagem profunda nessas áreas pode ser atribuído à alta capacidade dos modelos, ao aumento da potência computacional e à disponibilidade de dados rotulados em grande escala [80].

O treinamento de modelos de aprendizagem profunda geralmente requer recursos computacionais substanciais devido à complexidade dos modelos e ao volume de dados processados. Isso se deve principalmente ao fato de esses modelos usarem várias camadas para representar abstrações de dados, o que exige um poder computacional significativo, especialmente durante a fase de treinamento. Por exemplo, uma única inferência de um modelo de aprendizagem profunda pode exigir bilhões de operações de multiplicação e acumulação, tornando os modelos de aprendizado profundo extremamente exigentes em termos de computação e energia [81].

Além disso, as demandas computacionais dos aplicativos de aprendizagem profunda em vários campos, como reconhecimento de imagens, reconhecimento de voz e tradução, demonstraram ser muito dependentes do aumento da capacidade de computação. Essa dependência está se tornando insustentável do ponto de vista econômico, técnico e ambiental, exigindo o desenvolvimento de métodos mais eficientes do ponto de vista computacional [82].

Além disso, embora a aprendizagem profunda tenha feito avanços significativos em vários campos devido à disponibilidade de grandes *datasets* anotados, a aquisição desses *datasets* geralmente consome muitos recursos. Isso levou ao desenvolvimento de métodos como a aprendizagem ativa profunda, que visa a maximizar o desempenho do modelo e, ao mesmo tempo, minimizar o número de amostras anotadas necessárias [83].

2.3.1 Principais Arquiteturas

As principais arquiteturas da aprendizagem profunda incluem, entre outros, redes neurais convolucionais, redes neurais recorrentes, redes de crença profunda e *autoencoders*.

As redes neurais convolucionais (*convolutional neural networks* — CNNs) [84] são uma classe de modelos de aprendizagem profunda particularmente adequados para tarefas que envolvem dados espaciais, como reconhecimento de imagens e vídeos. Elas ganharam muita atenção devido à sua capacidade de aprender automaticamente e de forma adaptável hierarquias espaciais de recursos por meio de *backpropagation*, utilizando vários blocos de construção, como camadas de convolução, camadas de agregação (*pooling*) e camadas totalmente conectadas [85], conforme exposto na Figura 2.2. As camadas de convolução aplicam um conjunto de filtros aos dados de entrada, capturando padrões locais e hierarquias espaciais, enquanto as camadas de *pooling* reduzem a dimensionalidade dos mapas de recursos, tornando o cálculo mais eficiente e robusto às variações espaciais [86]. As CNNs evoluíram significativamente, com avanços no *design* das camadas, nas funções de ativação, nas funções de perda, na regularização e nas técnicas de otimização, levando a um desempenho de ponta em várias aplicações [87]. Elas têm sido aplicadas com sucesso em diversos campos, como visão computacional, reconhecimento de fala e processamento de linguagem natural, demonstrando sua versatilidade e eficácia. Além disso, as CNNs foram adaptadas para lidar com diferentes tipos de dados, incluindo dados 1-D, 2-D e multidimensionais, expandindo ainda mais sua aplicabilidade [88]. Apesar de seu sucesso, desafios como o excesso de ajuste e a necessidade de grandes *datasets* anotados permanecem, o que leva a pesquisas contínuas para desenvolver modelos mais eficientes e generalizáveis [85].

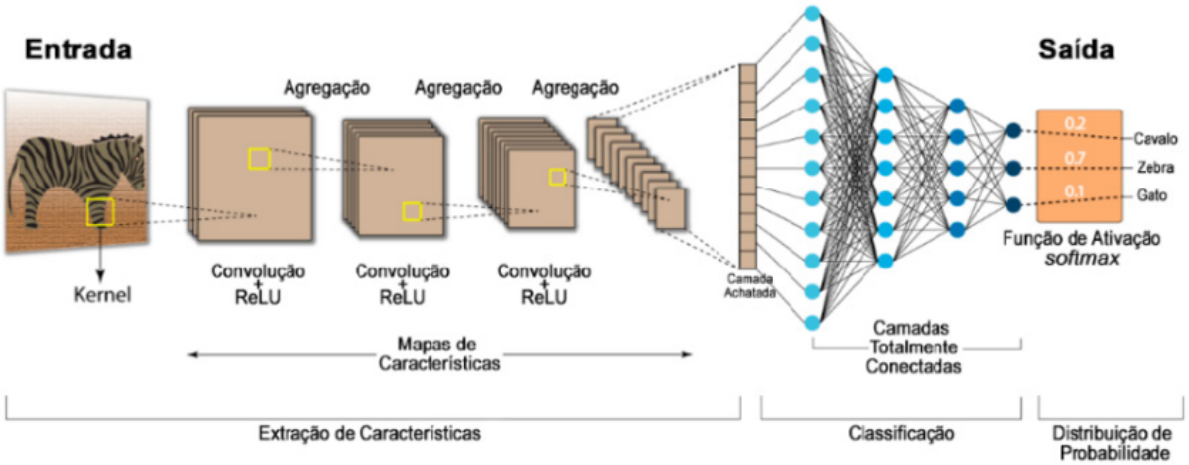

Figura 2.2 – Arquitetura de uma Rede Neural Convolucional. Extraída de [89].

As redes neurais recorrentes (*recurrent neural networks* — RNNs), cuja arquitetura é visualmente apresentada na Figura 2.3, são uma classe de redes neurais artificiais projetadas para reconhecer padrões em sequências de dados, como texto, áudio e dados de séries temporais. Diferentemente das redes neurais tradicionais de alimentação, as RNNs têm conexões que formam ciclos direcionados, o que lhes permite manter uma memória de entradas anteriores. Isso as torna particularmente eficazes para tarefas em que o contexto e as informações sequenciais são cruciais. No entanto, as RNNs padrão

enfrentam desafios, como os problemas de desvanecimento e explosão de gradiente, que prejudicam sua capacidade de aprender dependências de longo prazo. Para resolver esses problemas, foram introduzidas as redes *long short-term memory* (LSTM) [90] e *gated recurrent units* (GRUs) [91], que incorporam funções de portas para gerenciar o fluxo de informações e capturar efetivamente as dependências de longo prazo [92], conforme a Figura 2.4. As RNNs foram aplicadas em vários domínios, incluindo reconhecimento de fala, saúde preditiva e até mesmo sistemas de recomendação, onde podem modelar dados baseados em sessões para fornecer recomendações mais precisas [93]. Apesar de seu uso generalizado, as RNNs apresentam desafios significativos de treinamento, principalmente em ambientes com recursos limitados, exigindo o desenvolvimento de técnicas eficientes de treinamento e compressão [94]. De modo geral, as RNNs são ferramentas versáteis com amplas aplicações, mas exigem projeto e treinamento cuidadosos para superar as limitações inerentes e maximizar seu potencial [95].

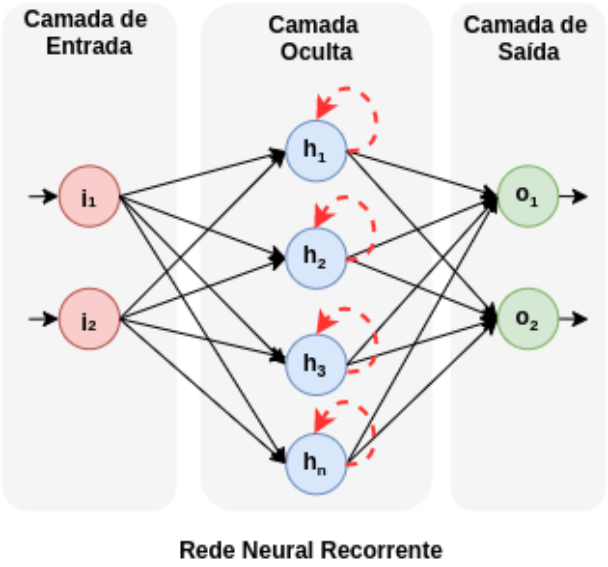

Figura 2.3 – Arquitetura de uma Rede Neural Recorrente. Extraída de [96].

As redes de crença profunda (*deep belief networks* — DBNs), representadas pela Figura 2.5, são uma classe de modelos de redes neurais generativas que consistem em várias camadas de variáveis ocultas, introduzidas por [98]. Essas redes são construídas usando um modelo probabilístico conhecido como Máquina de Boltzmann Restrita (*restricted Boltzmann machine* — RBM) para cada camada, o que facilita a inferência e tem sido usado com eficácia para treinar modelos mais profundos [99]. As DBNs são particularmente notáveis por sua capacidade de aprender representações internas ricas a partir de dados sensoriais por meio de aprendizagem não supervisionada, o que contribuiu significativamente para a revolução da aprendizagem profunda [100]. O treinamento de DBNs geralmente envolve um algoritmo de aprendizado não supervisionado, guloso e em camadas, seguido de um *fine-tuning* (ajuste fino) usando métodos de aprendizado supervisionado [101], [102]. Essa abordagem permite que as DBNs superem desafios como baixa velocidade

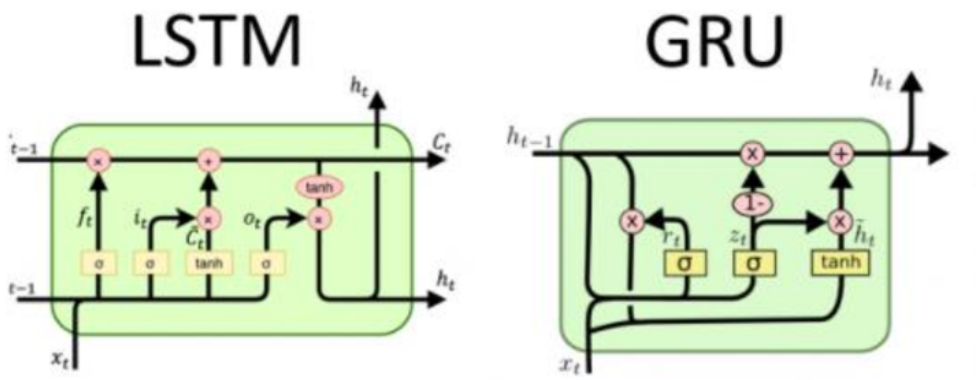

Figura 2.4 – Arquiteturas das redes LSTM e GRU. Extraída de [97]. Imagem adaptada dos trabalhos de [90] e [91].

de treinamento e sobreajuste, que são comuns em redes neurais profundas [103]. Além disso, as DBNs foram aplicadas com sucesso em vários domínios, incluindo classificação de imagens, reconhecimento de fala e compreensão de linguagem natural, demonstrando sua versatilidade e eficácia [102].

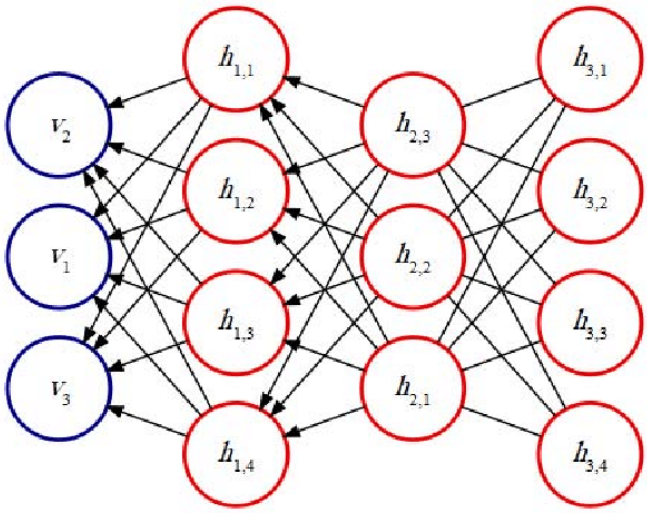

Figura 2.5 – Arquitetura das Redes de Crença Profunda. Extraída de [104].

Os *autoencoders* são um tipo de arquitetura de rede neural usada principalmente para tarefas de aprendizagem não supervisionada, em que o objetivo é aprender representações eficientes dos dados. Eles consistem em duas partes principais, conforme a Figura 2.6: um codificador e um decodificador. O codificador comprime os dados de entrada em um espaço latente de dimensão inferior, enquanto o decodificador tenta reconstruir os dados originais a partir dessa representação comprimida. Esse processo permite que os

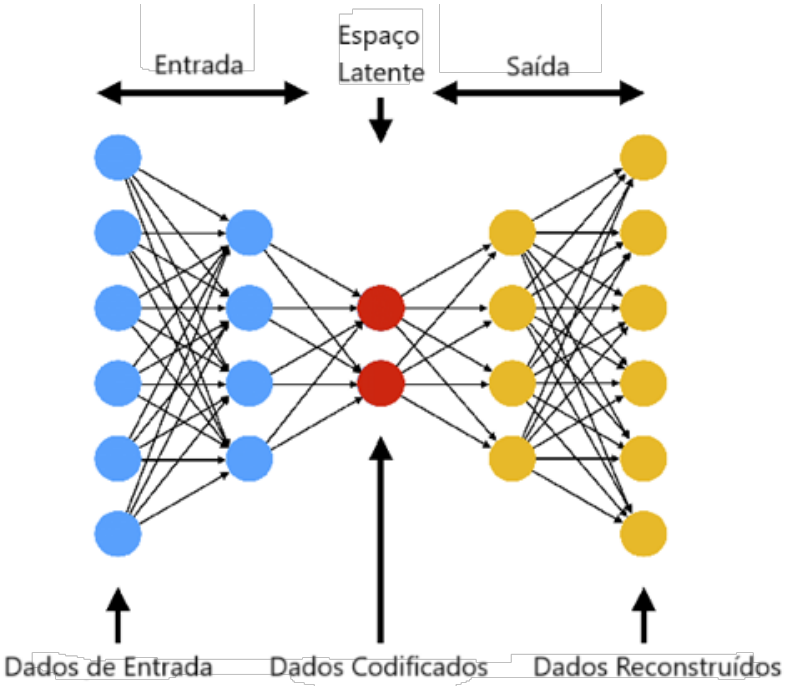

Figura 2.6 – Arquitetura de um *Autoencoder*. Adaptado de [108].

codificadores automáticos capturem os recursos mais importantes dos dados, o que os torna úteis para tarefas como redução de dimensionalidade, detecção de anomalias e redução de ruído dos dados [105]. Variações de *autoencoders*, como os *autoencoders* orientados por lógica, incorporam operações de lógica *fuzzy* para aprimorar os processos de codificação e decodificação [106], enquanto os *autoencoders* quânticos, por outro lado, são projetados para compactar dados quânticos, aproveitando os algoritmos de otimização clássicos para treinamento [107]. Essas diversas aplicações destacam a versatilidade e a importância dos *autoencoders* no aprendizado de máquina moderno e no processamento de dados.

2.3.2 Processamento de Linguagem Natural Aplicado à Língua Portuguesa

A aplicação do PLN à língua portuguesa apresenta desafios únicos, principalmente devido à sua diversidade lexical e ambiguidades inerentes.

2.3.2.1 Ambiguidade Lexical

A ambiguidade lexical surge quando uma palavra tem vários significados, o que pode gerar confusão na compreensão do contexto. Em português, esse fenômeno é particularmente acentuado devido à riqueza do vocabulário e à natureza polissêmica do idioma. Por exemplo, a palavra “banco” pode se referir a uma instituição financeira ou a um lugar para sentar, dependendo do contexto. Essas ambiguidades complicam tarefas como a desambiguação, em que o objetivo é determinar o sentido correto de uma palavra com base em seu uso em uma frase. Estudos recentes mostraram que a ambiguidade pode prejudicar significativamente o desempenho dos sistemas de PLN, exigindo modelos avançados que possam efetivamente

desambiguar os significados no contexto [109].

2.3.2.2 Diversidade Lexical

O português é caracterizado por um alto grau de diversidade lexical, que pode ser atribuído à sua evolução histórica e à influência de dialetos regionais. Essa diversidade apresenta desafios significativos para as aplicações do PLN. A presença de sinônimos, variações regionais e expressões coloquiais pode levar a inconsistências na forma como o idioma é processado.

A evolução histórica do português levou a uma rica variedade de dialetos regionais e variações lexicais. Por exemplo, um estudo sobre o português falado contemporâneo nas áreas urbanas do Funchal e nas áreas rurais da Ilha da Madeira destaca a coexistência de formas dialetais com variantes padrão do português europeu. Esse estudo mostra que os falantes de áreas urbanas apresentam mais flexibilidade e variabilidade em suas escolhas lexicais em comparação com os de áreas rurais, que tendem a ser mais conservadores. Essa variabilidade e estabilidade nos significados lexicais ressaltam a heterogeneidade da língua portuguesa e suas identidades regionais [110].

Essas variações regionais e a presença de formas dialetais representam desafios significativos para as aplicações do PLN. As ferramentas de PLN geralmente enfrentam vieses de representação, que podem resultar em comportamento discriminatório. A maioria dos esforços de pesquisa existentes em PLN justo concentrou-se em idiomas anglo-saxões, deixando uma lacuna na abordagem dessas questões para a língua portuguesa. Há mesmo um apelo na comunidade científica para estimular mais pesquisas em PLN justa especificamente para o português, a fim de garantir que essas ferramentas possam lidar com a diversidade lexical do idioma de forma eficaz [111].

2.3.2.3 Iniciativas

Várias iniciativas surgiram para enfrentar os desafios do PLN em português. Um modelo notável é o BERTimbau, que é uma variante da arquitetura BERT ajustada para o português brasileiro. Esse modelo foi utilizado em várias aplicações, incluindo análise de sentimentos, extração de aspectos e detecção de discurso de ódio, demonstrando sua versatilidade e eficácia na compreensão das complexidades do idioma [18], [112].

Além disso, o desenvolvimento de modelos especializados para textos clínicos, como os que utilizam a classificação BI-RADS, destaca o potencial do PLN em domínios específicos. Esses modelos empregam técnicas de aprendizagem por transferência para adaptar representações linguísticas pré-treinadas a tarefas especializadas, melhorando assim a precisão e a eficiência no processamento de registros médicos [113]. Essas iniciativas ressaltam a importância de adaptações específicas ao contexto em aplicações do PLN para o português.

Além disso, o desenvolvimento de *datasets* e *corpora* para a língua portuguesa tem recebido atenção significativa. Exemplos notáveis incluem o BrWaC (*Brazilian Portuguese Web as Corpus*) [114], o ASSIN (Avaliação de Similaridade Semântica e INferência textual) [115] e seu sucessor ASSIN2 [116]. O *corpus* BrWaC, que consiste em uma grande coleção de páginas da *Web* em português do Brasil, foi utilizado para pré-treinamento de modelos como o T5, demonstrando desempenho superior em tarefas como similaridade de frases e associação quando comparado a modelos multilíngues [117]. Os *datasets* ASSIN fornecem dados estruturados para avaliar a similaridade semântica, com o ASSIN2 oferecendo um *benchmark* mais extenso que facilitou o desenvolvimento de modelos de última geração nesse domínio [118], [119].

2.3.2.4 Direções Futuras

Para garantir práticas de PLN justas e éticas para a língua portuguesa, é essencial estabelecer uma agenda de pesquisa dedicada. Essa agenda deve se concentrar na identificação e na atenuação de vieses, aproveitando as estruturas éticas existentes e abordando os desafios exclusivos apresentados pela língua portuguesa. Com isso, será possível desenvolver modelos de PLN que sejam linguisticamente robustos e eticamente sólidos, promovendo a equidade e a justiça na representação da linguagem [111].

2.4 Representação de Palavras

A representação de palavras é um conceito fundamental no Processamento de Linguagem Natural, fornecendo um meio de transformar dados textuais em formatos numéricos adequados para o processamento computacional. Ao longo do tempo, várias técnicas foram desenvolvidas para melhorar a expressividade e a eficácia dessas representações.

2.4.1 *Bag-of-Words*

O modelo de saco-de-palavras (*Bag-of-Words* — BoW) reduz o texto a um multiconjunto não ordenado de palavras, descartando a estrutura sintática e semântica e preservando a frequência lexical. Seja um documento d ser representado como um vetor $\mathbf{v}_d \in \mathbb{R}^{|V|}$, em que V é um vocabulário fixo. Cada componente v_i corresponde à frequência do termo t_i em d, conforme demonstrado pela Equação 2.1:

$$v_i = \text{count}(t_i, d) \tag{2.1}$$

Como exemplo, na frase “Gosto de jogar futebol.”, o respectivo BoW representado como um objeto JSON é `{"Gosto": 1, "de": 1, "jogar": 1, "futebol": 1}`.

Embora o BoW permita a classificação e a recuperação simples de documentos, suas limitações são significativas:

- **Esparsidade**: Os vetores de alta dimensão dominam os recursos computacionais, com a maioria das entradas sendo zero para grandes vocabulários;
- **Cegueira semântica**: Sinônimos (por exemplo, “carro” e “automóvel”) e palavras polissêmicas (por exemplo, “banco”) são confundidos;
- **Ignorância de contexto**: A ordem das palavras e as relações sintáticas são apagadas, tornando frases como “cachorro morde homem” indistinguíveis de “homem morde cachorro”.

2.4.2 Frequência do Termo–inverso da Frequência nos Documentos

O TF-IDF refina o BoW ponderando os termos com base em seu poder de discriminação em um corpus D. A pontuação TF-IDF para o termo t no documento d é definida pela Equação 2.2:

$$\text{tf-idf}(t, d, D) = \text{tf}(t, d) \times \log\left(\frac{N}{\text{df}(t)}\right) \tag{2.2}$$

em que N é o número total de documentos, $\text{tf}(t, d)$ é a frequência de termos e $\text{df}(t)$ é a frequência de documentos de t.

Considere-se um conjunto de três documentos:

- **Documento 1**: "A maçã é uma fruta deliciosa"
- **Documento 2**: "A banana é uma fruta saudável"
- **Documento 3**: "Eu gosto de maçã e banana"

Para calcular o TF-IDF da palavra “maçã” no Documento 1, é necessário efetuar os seguintes cálculos:

$$\text{TF}(\text{“maçã”}, D_1) = \frac{\text{Número de vezes que “maçã” aparece em } D_1}{\text{Número total de palavras em } D_1} = \frac{1}{5} = 0.2$$

$$\text{IDF}(\text{“maçã”}) = \log\left(\frac{3}{2}\right) = \log(1.5) \approx 0.18$$

$$\text{TF-IDF}(\text{“maçã”}, D_1) = 0.2 \times 0.18 = 0.036$$

Apesar de sua utilidade na recuperação de informações, o TF-IDF herda as limitações do BoW e introduz novos problemas:

- **Pesos estáticos**: O TF-IDF não consegue se adaptar às mudanças semânticas em *corpora* dinâmicos;
- **Viés centrado nos documentos**: os pesos globais do IDF ignoram as nuances contextuais locais.

2.4.3 *Word Embeddings*

A representação de palavras em formato vetorial, conhecida como *word embeddings*, revolucionou o processamento de linguagem natural ao permitir que máquinas manipulem palavras de forma numérica. Em vez de representá-las como índices em um vocabulário, os *embeddings* atribuem a cada palavra um vetor denso de dimensão fixa, no qual palavras semanticamente semelhantes possuem representações próximas no espaço vetorial. Diferentes técnicas foram desenvolvidas para gerar *embeddings* de palavras, cada uma com características e vantagens próprias, sendo três as mais influentes: Word2Vec, GloVe e FastText.

O Word2Vec [120] é uma das primeiras e mais conhecidas abordagens para aprendizado de *embeddings*. Ele baseia-se em redes neurais simples para aprender representações vetoriais de palavras a partir de grandes corpora de texto. Existem duas variações principais desse modelo, representadas na Figura 2.7:

- **CBOW (*Continuous Bag-of-Words*)**: prevê uma palavra-alvo com base nas palavras ao seu redor dentro de uma janela de contexto.
- ***Skip-Gram***: faz o inverso do CBOW, tentando prever as palavras de contexto a partir de uma palavra-alvo.

Para otimizar o aprendizado, o Word2Vec pode utilizar técnicas como *negative sampling* (técnica que tem como objetivo maximizar a semelhança das palavras no mesmo contexto e minimizá-la quando elas ocorrem em contextos diferentes) ou *hierarchical softmax* (técnica que organiza as palavras em uma estrutura de árvore, com base em sua frequência de ocorrência, na qual as palavras mais frequentes são armazenadas mais perto da raiz), que tornam o treinamento mais eficiente ao evitar cálculos custosos sobre todo o vocabulário. Os *embeddings* gerados pelo Word2Vec capturam relações semânticas e sintáticas entre palavras, permitindo que operações como “rei” - “homem” + “mulher” resultem em um vetor próximo ao de “rainha”.

Diferente do Word2Vec, que aprende representações a partir de janelas de contexto local, o GloVe (*Global Vectors*) [121] adota uma abordagem baseada em estatísticas globais

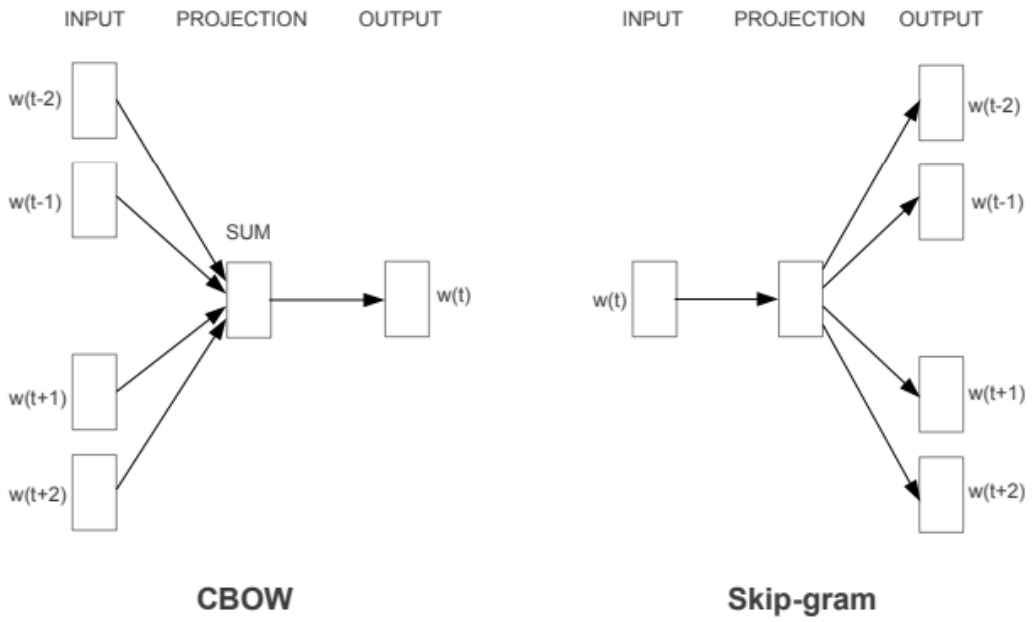

Figura 2.7 – Arquiteturas CBOW e *Skip-gram*. Extraída de [120].

do *corpus*. Ele constrói uma matriz de coocorrência que conta quantas vezes cada par de palavras aparece junto e aprende acerca dos *embeddings* otimizando uma função de custo que ajusta essas contagens de forma eficiente. A vantagem do GloVe está na sua capacidade de capturar melhor relações semânticas globais, tornando-o mais adequado para tarefas em que o significado das palavras depende de padrões amplos de coocorrência.

O FastText [122] expande a abordagem do Word2Vec ao representar palavras como conjuntos de n-gramas de caracteres, em vez de tratá-las como unidades indivisíveis. Isso permite que palavras com raízes ou sufixos semelhantes tenham representações mais próximas, o que é especialmente útil para idiomas morfologicamente ricos ou para lidar com palavras raras e desconhecidas. Por exemplo, a palavra "correndo" pode ser dividida em n-gramas como "cor", "rre", "ren" e "endo". O vetor da palavra final é então obtido a partir da soma dos vetores desses n-gramas, conforme demonstrado pela Figura 2.8. Com isso, o FastText é mais robusto para lidar com palavras novas, tornando-o uma escolha ideal para aplicações que envolvem vocabulários dinâmicos ou dados ruidosos.

2.4.4 Limitações

Word embeddings, embora poderosas, têm várias limitações que afetam seu desempenho e confiabilidade em vários aplicações.

- *Word embeddings* são altamente sensíveis a variações no corpus de treinamento. Pequenas alterações nos dados podem levar a diferenças significativas nos *embeddings* resultantes, especialmente quando se usam corpora menores. Essa instabilidade sugere

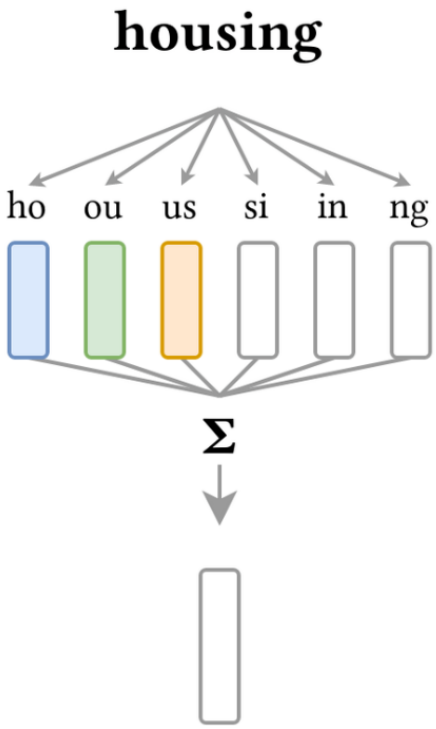

Figura 2.8 – Funcionamento do FastText. Adaptado de [123]

que confiar em um único modelo de incorporação pode ser enganoso, e recomenda-se calcular a média de várias amostras para obter resultados mais confiáveis [124];

- *Word embeddings* geralmente não têm dimensões interpretáveis, o que dificulta a compreensão dos recursos semânticos que elas codificam. Isso contribui para a natureza de "caixa preta" das tarefas que usam essas incorporações, pois os motivos de seu desempenho permanecem obscuros [125];
- *Word embeddings* podem conter e amplificar vieses presentes nos dados de treinamento, como estereótipos e preconceitos. Esses vieses podem ser transferidos para outros modelos de aprendizado de máquina, o que apresenta desafios para implementações algorítmicas justas [126];
- Mesmo para palavras relativamente frequentes, as *embeddings* podem ser instáveis, afetando seu desempenho em tarefas posteriores. Vários fatores contribuem para essa instabilidade, exigindo uma consideração cuidadosa em sua aplicação [127];
- Técnicas como word2vec e GloVe podem não produzir *embeddings* de alta qualidade a partir de corpora pequenos ou esparsos, limitando sua eficácia em determinados contextos [128].

2.5 Arquitetura *Transformer*

O advento dos modelos *Transformer* marcou um ponto de virada significativo no campo do PLN. Introduzida por [129], a arquitetura *Transformer* alterou fundamentalmente o cenário do PLN ao permitir que os modelos processassem sequências de texto em paralelo, em vez de sequencialmente, como era a norma com arquiteturas anteriores, como redes neurais recorrentes e redes LSTM. Essa capacidade de processamento paralelo é atribuída, em grande parte, ao mecanismo de autoatenção (*self-attention*), que permite que o modelo pondere a importância de diferentes palavras em uma frase em relação umas às outras, capturando assim dependências complexas e relações contextuais de forma mais eficaz do que seus predecessores.

Um dos avanços mais notáveis trazidos pelos modelos *Transformer* é sua capacidade de lidar com dependências de longo alcance no texto. Os modelos tradicionais geralmente têm dificuldade em manter o contexto em sequências longas, o que leva a um desempenho prejudicado em tarefas que exigem uma compreensão de relações textuais mais amplas. Em contrapartida, os *Transformers* utilizam a autoatenção para se concentrar dinamicamente em partes relevantes da entrada, independentemente de sua posição na sequência. Esse recurso se mostrou particularmente benéfico em aplicações como resumo de documentos e *question answering*, em que a compreensão do contexto das informações espalhadas por várias frases é crucial [130].

2.5.1 Funcionamento

Diferente dos modelos tradicionais que dependem de redes neurais recorrentes, a arquitetura *Transformer* adota o mecanismo de autoatenção para avaliar a importância de cada palavra em uma frase em relação a todas as outras. Esse recurso permite ao modelo capturar o contexto global da frase, compreendendo de forma mais precisa as dependências e relações entre as palavras. Em contraste com as RNNs, que processam sequências de forma sequencial e sofrem com dificuldades em capturar dependências de longo prazo, a arquitetura *Transformer* utiliza a autoatenção para capturar de forma mais eficaz essas dependências ao processar toda a sequência simultaneamente. Essa abordagem possibilita o processamento paralelo de sequências, acelerando o treinamento e melhorando o desempenho em tarefas complexas [131].

A arquitetura *Transformer*, representada graficamente pela Figura 2.9, é baseada em um modelo codificador-decodificador (*encoder-decoder*) , onde o codificador processa a sequência de entrada e gera representações contextuais que capturam seu significado. O decodificador, por sua vez, utiliza essas representações para produzir a sequência de saída de forma coerente e precisa. Esta estrutura é particularmente eficaz em tarefas como tradução automática, onde a entrada em um idioma precisa ser compreendida (codificada)

antes de ser traduzida (decodificada) para outro idioma.

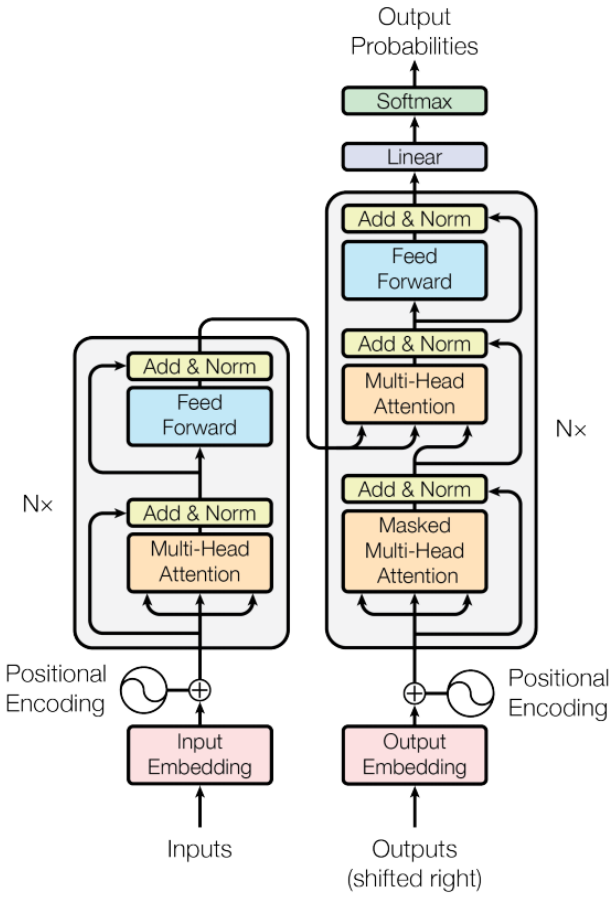

Figura 2.9 – Arquitetura Transformer. Extraída de [129].

Um elemento fundamental para o processamento de sequências na arquitetura *Transformer* é a representação das palavras através de *embeddings*. Tal como nos *word embeddings* anteriormente citados, um *embedding*, ou vetor de *embedding*, é a representação vetorial de uma palavra ou token em um espaço de características $\mathbb{R}^d$, onde d é a dimensão do modelo, e palavras com significados ou contextos semelhantes possuem vetores mais próximos em termos de similaridade cossenal. Formalmente, para um vocabulário V, cada palavra $w \in V$ é mapeada para um vetor $e_w \in \mathbb{R}^d$.

Para preservar a informação sobre a ordem das palavras na sequência, o que é crucial para a compreensão contextual, o modelo adiciona *encodings* posicionais aos *embeddings* de entrada. Estes são definidos através de funções sinusoidais, conforme as Equações 2.3 e 2.4:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{2.3}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{2.4}$$

onde $pos \in [0, n-1]$ representa a posição na sequência de comprimento n, e $i \in [0, d/2-1]$ indica a dimensão do *encoding*. O *embedding* final de cada *token* x_{pos} é dado pela soma do *embedding* da palavra com o *encoding* posicional:

$$x_{pos} = e_w + PE_{pos} \tag{2.5}$$

No núcleo do mecanismo de atenção da arquitetura *Transformer* está o conceito de *scaled dot-product attention* (atenção de produto escalar em escala), responsável por calcular a relevância entre cada par de palavras em uma sequência. A atenção é aplicada a partir da matriz de *embeddings* de entrada $X \in \mathbb{R}^{n \times d}$, onde n é o comprimento da sequência e d é a dimensão do modelo. A operação central da atenção é representada pela Equação 2.6:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.6}$$

Aqui, Q, K e V são projeções da entrada X, obtidas por meio de transformações lineares:

- $Q = XW^Q$: representam as "perguntas" (*queries*) que cada palavra ou *token* faz sobre outras palavras na sequência, identificando o que é relevante no contexto;
- $K = XW^K$: servem como "identificadores" (*keys*) que permitem a correspondência entre palavras, ajudando a responder às perguntas dos *queries* e capturando a essência de cada palavra;
- $V = XW^V$: contém os "valores" (*values*), ou seja, as informações que serão ponderadas pela atenção.

Nessas projeções, d_k representa a dimensão das matrizes Q e K, ou seja, a quantidade de características que cada palavra "consulta" ou "compara" em outra. O fator de escala $\sqrt{d_k}$ é usado para ajustar a magnitude dos valores em QK^T, evitando que valores excessivamente altos prejudiquem o aprendizado.

A operação de atenção em modelos Transformers usa a função softmax para transformar pontuações em probabilidades, normalizando as relações entre as palavras. Essas pontuações representam a relevância de cada palavra em relação a outra palavra específica da sequência. Em outras palavras, cada pontuação indica o quanto uma palavra pode contribuir para o significado contextual de outra.

A função softmax, aplicada ao vetor de pontuações z, é dada por:

$$\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{2.7}$$

Cada pontuação z_i passa pela função exponencial e^{z_i}, o que garante que todos os valores fiquem positivos e amplia a diferença entre eles, destacando as pontuações mais altas. Em seguida, o denominador $\sum_{j=1}^{n} e^{z_j}$ calcula a soma de todos os valores exponenciais, normalizando o resultado para formar uma distribuição de probabilidades. Isso faz com que as palavras mais relevantes, com pontuações mais altas, recebam probabilidades proporcionais maiores, permitindo que o modelo se concentre nas palavras que mais contribuem para o contexto da sequência e capture de forma eficaz as interdependências entre elas.

Para capturar diferentes aspectos das relações entre tokens, a arquitetura *Transformer* utiliza múltiplas cabeças de atenção em paralelo, um mecanismo chamado de *multihead attention* (atenção com múltiplas cabeças), representado graficamente pela Figura 2.10 junto da *scaled dot-product attention*. Esse processo é formalizado pelas Equações 2.8 e 2.9:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{2.8}$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.9}$$

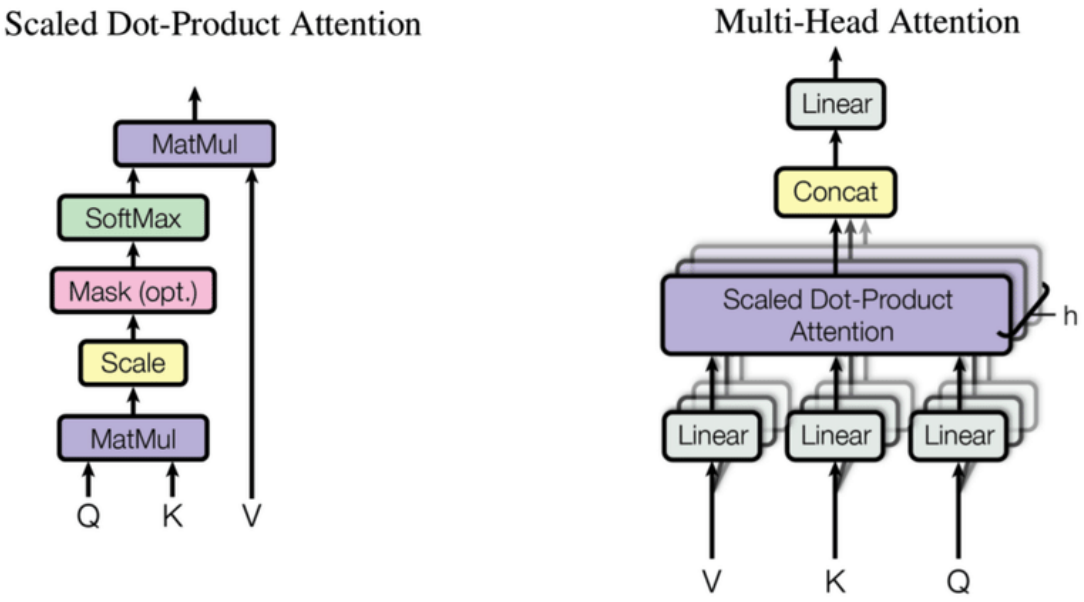

Figura 2.10 – *Scaled dot-product attention* (atenção de produto escalar em escala) e *Multihead attention* (atenção com múltiplas cabeças). Extraída de [129].

onde h representa o número de cabeças de atenção. Cada cabeça processa a entrada a partir de diferentes "perspectivas" ou projeções, determinadas por conjuntos únicos de pesos para as matrizes Q, K e V. Esse uso de cabeças múltiplas permite que o modelo capture diferentes aspectos do contexto, o que é particularmente valioso em tarefas onde múltiplas perspectivas são necessárias [132], como na tradução de sentenças complexas. Ao final, os resultados das cabeças são concatenados e multiplicados pela matriz W^O, que realiza a projeção final para a dimensão do modelo. A dimensão de cada projeção d_k é ajustada para d/h, onde d é a dimensão total do modelo e h é o número de cabeças de atenção, permitindo que o modelo divida uniformemente a capacidade de atenção entre as diferentes cabeças.

A equação 2.10 define a máscara de atenção utilizada no decodificador, um componente crucial para manter a causalidade na arquitetura *Transformer*. Ela garante que cada palavra ou *token* atual só possa se basear em *tokens* anteriores, impedindo que informações futuras influenciem o contexto corrente. Essa causalidade é essencial para gerar uma sequência coerente e estruturada de maneira auto-regressiva:

$$M_{ij} = \begin{cases} 0 & \text{, se } i \geq j \\ -\infty & \text{, caso contrário} \end{cases} \tag{2.10}$$

onde M_{ij} representa o elemento na posição (i, j) da matriz de máscara. Quando $i \geq j$, torna-se possível a atenção (valor 0), e quando $i < j$, é bloqueada a atenção (valor $-\infty$), que após a aplicação da função softmax se torna efetivamente zero.

Para garantir um treinamento estável e eficiente, a arquitetura *Transformer* emprega um mecanismo chamado *Add & Norm* após cada subcamada, tanto no codificador quanto no decodificador. Esse processo é formalizado na Equação 2.11, enquanto a definição específica da normalização de camada é apresentada na Equação 2.12:

$$(x + \text{Sublayer}(x)) \tag{2.11}$$

Na Equação 2.11, o termo $x + \text{Sublayer}(x)$ indica que o valor original da entrada, x, é somado ao resultado de uma transformação específica aplicada a essa entrada, que pode ser chamada de "subcamada" (*sublayer*). Esse processo de somar a entrada original ao resultado da transformação é conhecido como conexão residual, e ele ajuda a evitar problemas comuns em redes neurais profundas, como a degradação do gradiente. Em termos práticos, a conexão residual permite que a informação original da entrada passe adiante, mesmo depois de modificada pela subcamada.

Após essa soma, o valor resultante passa por uma etapa chamada normalização de camada, ou *LayerNorm*, representada pela Equação 2.12. A normalização de camada é um processo que ajusta a escala dos valores de entrada de modo que tenham média zero e

variância uniforme, o que facilita o aprendizado da rede ao longo das camadas. Na prática, isso significa que as ativações das camadas não irão variar muito em termos de amplitude, o que estabiliza a rede.

Na Equação 2.12, temos a seguinte expressão para a normalização de camada:

$$\text{LayerNorm}(x) = \gamma \odot \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{2.12}$$

Aqui, o termo μ representa a média dos valores de entrada, e σ^2 é a variância. Eles são usados para ajustar o centro e a dispersão dos dados, respectivamente. O termo ϵ é um pequeno valor adicionado para evitar problemas de divisão por zero — normalmente, um valor como 1×10^{-6} é suficiente para garantir estabilidade.

Além disso, os parâmetros γ e β são "aprendíveis" pelo modelo, o que significa que eles são ajustados ao longo do treinamento para que a rede aprenda a melhor escala (γ) e o melhor deslocamento (β) para normalizar a camada.

Após o mecanismo de atenção e sua respectiva camada *Add & Norm*, cada bloco do codificador e decodificador contém uma rede *feed-forward* que processa cada posição independentemente. Cada uma dessas redes é composta por duas camadas lineares intercaladas pela função de ativação ReLU (*Rectified Linear Unit*), que transforma apenas valores positivos, garantindo que as representações de características permaneçam consistentes para as próximas camadas. Esta rede, também seguida por sua própria camada *Add & Norm*, é definida pela equação 2.13:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{2.13}$$

Esta é uma rede neural de duas camadas com ativação ReLU ($\max(0, \cdot)$), onde W_1 e W_2 são matrizes de peso, e b_1 e b_2 são vetores de viés. A primeira transformação expande a dimensão por um fator de 4 [133], e a segunda a reduz de volta à dimensão original do modelo.

Esta expansão por um fator de 4 significa que o tamanho do vetor de entrada é temporariamente multiplicado por 4 na primeira etapa da rede. Em outras palavras, se o vetor de entrada tiver 512 componentes, ele será expandido para 2048 componentes. Essa expansão permite que a rede crie uma representação mais detalhada e rica dos dados, pois cada componente do vetor processado pode armazenar mais informações.

Embora o mecanismo de atenção apresente uma complexidade superior à de RNNs de $O(n^2 \cdot d)$, onde n é o comprimento da sequência e d é a dimensão do modelo, a capacidade de paralelização dos modelos *Transformer* compensa o custo computacional, proporcionando um treinamento mais rápido e eficiente. O termo quadrático n^2 surge porque cada posição precisa atender a todas as outras posições, resultando em uma matriz

de atenção $n \times n$. No entanto, diferentemente das RNNs, todas estas operações podem ser realizadas simultaneamente em *hardware* moderno.

Para garantir que o treinamento do modelo comece bem e os parâmetros se ajustem de forma eficiente, os pesos das diversas camadas do modelo são configurados inicialmente seguindo uma estratégia específica de [134], descrita na equação 2.14:

$$W_l \sim \mathcal{N}\left(0, \sqrt{\frac{2}{n_l + n_{l+1}}}\right) \tag{2.14}$$

Essa fórmula significa que os pesos da camada l são escolhidos a partir de uma distribuição normal com uma média de zero. A variação dos valores dos pesos é controlada por um cálculo específico, que depende do número de conexões de entrada e de saída da camada. Esse cálculo é feito com os tamanhos da camada anterior e da próxima camada, indicados por n_l e n_{l+1}, respectivamente.

Esse método para definir a variância inicial dos pesos é essencial para evitar problemas que dificultam o aprendizado do modelo, como o desvanecimento ou a explosão dos gradientes. Esses problemas ocorrem quando os valores das atualizações durante o treinamento se tornam muito pequenos ou muito grandes, dificultando o ajuste dos parâmetros do modelo. Ajustando a variância dos pesos dessa forma, o modelo começa a aprender de maneira mais estável e eficaz desde as primeiras iterações, promovendo um treinamento mais consistente.

2.6 Aprendizado por Transferência

A aprendizagem por transferência surgiu como um paradigma transformador no campo da aprendizagem automática, especialmente no PLN. Essa abordagem permite que os modelos aproveitem o conhecimento adquirido em uma tarefa para melhorar o desempenho em outra tarefa, geralmente relacionada. O princípio fundamental por trás da aprendizagem por transferência é utilizar modelos pré-treinados que foram desenvolvidos em grandes *datasets*, permitindo assim a aplicação efetiva desses modelos a tarefas específicas com disponibilidade limitada de dados. Isso é particularmente vantajoso em PLN, em que os *datasets* anotados geralmente são escassos e caros de se obter [135], [136], [137]. Um exemplo de aplicação desse paradigma pode ser visto na Figura 2.11, onde uma rede pré-treinada com o *dataset* ImageNet [138] (composto por figuras agrupadas em dezenas de milhares de categorias) tem parte desse pré-treinamento aproveitado para detecção e diagnóstico de leucemia.

O conceito de aprendizagem por transferência está fundamentado na ideia de que o conhecimento adquirido ao resolver um problema pode ser benéfico ao abordar um problema diferente, mas relacionado. Isso é particularmente relevante no contexto da aprendizagem

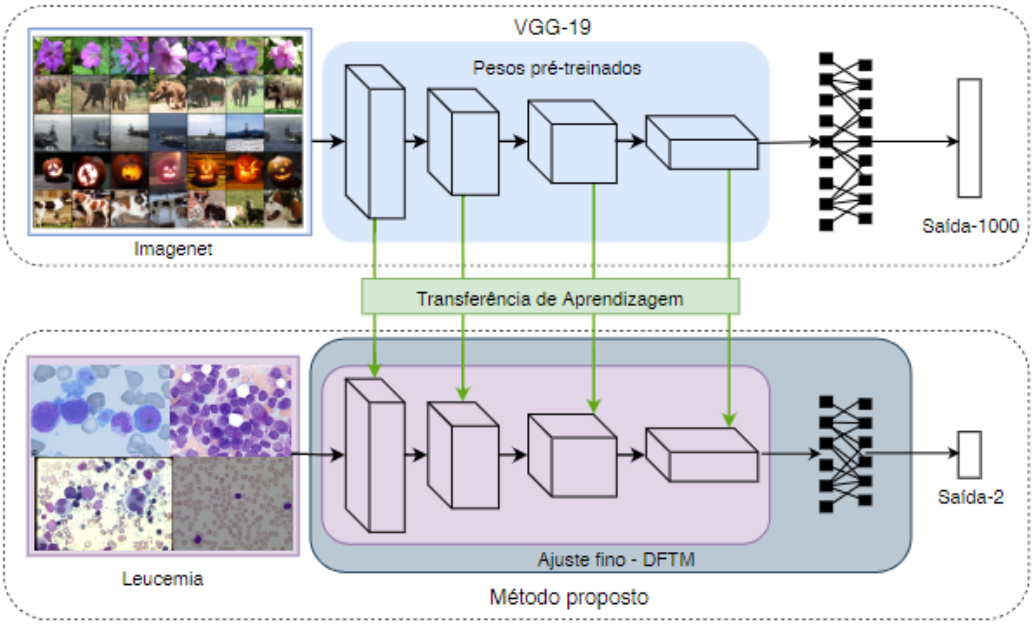

Figura 2.11 – Exemplo de aplicação do paradigma de aprendizado por transferência. Extraída de [139].

profunda, em que modelos treinados em *datasets* extensos podem capturar padrões e representações intrincados de linguagem. Os dois principais estágios da aprendizagem por transferência envolvem o pré-treinamento em um grande *corpus* para desenvolver uma compreensão geral da linguagem e o *fine-tuning* em uma tarefa específica para adaptar os recursos do modelo às nuances dessa tarefa. Com efeito, a aprendizagem por transferência tem sido aplicada com eficácia na aprendizagem por reforço profundo para melhorar a eficiência e a eficácia, aproveitando o conhecimento especializado externo [140].

A aprendizagem por transferência no PLN geralmente envolve dois processos principais: pré-treinamento e *fine-tuning*. Durante a fase de pré-treinamento, um modelo aprende a prever palavras mascaradas em uma frase ou a determinar a próxima palavra em uma sequência, desenvolvendo, assim, uma rica compreensão da estrutura e da semântica da linguagem [141]. Após a conclusão do pré-treinamento, o modelo passa por um *fine-tuning*, no qual é adaptado a uma tarefa específica usando um *dataset* menor e específico da tarefa. Esse estágio geralmente envolve o ajuste dos parâmetros do modelo para otimizar o desempenho na tarefa de destino [142].

Apesar de seus sucessos, a aprendizagem por transferência em PLN não está isenta de desafios. Um problema significativo é a possibilidade de transferência negativa, em que o conhecimento da tarefa de origem afeta negativamente o desempenho na tarefa de destino. Isso pode ocorrer quando as tarefas são muito diferentes ou quando o modelo pré-treinado não é ajustado adequadamente [143]. A dependência de grandes modelos pré-treinados também levanta preocupações sobre a eficiência computacional e a sustentabilidade ambiental, devido ao alto custo financeiro e à significativa pegada de carbono associados ao

treinamento e *fine-tuning* desses modelos [144]. Para mitigar esses problemas, pesquisadores têm explorado soluções como *knowledge distillation*, que resulta em modelos menores e mais eficientes, como o DistilBERT [145]. Além disso, técnicas de aprendizagem por transferência, como módulos adaptadores, ajudam a reduzir o número de parâmetros treináveis, mantendo a eficiência computacional sem sacrificar o desempenho [146].

Pesquisas futuras em aprendizagem por transferência para PLN provavelmente se concentrarão em aprimorar a adaptabilidade dos modelos pré-treinados a uma variedade mais ampla de tarefas e domínios. Isso é especialmente relevante em áreas especializadas, como o PLN clínico, onde a linguagem altamente técnica dificulta a transferência direta de modelos treinados em domínios gerais [147]. Além disso, investigações em métodos de transferência entre idiomas estão ganhando destaque, explorando como modelos treinados em uma língua podem ser aplicados de forma eficaz em outra.

2.6.1 Aprendizado *Zero-Shot* e *Few-Shot*

Os paradigmas de aprendizado *zero-shot* e *few-shot* estendem o aprendizado por transferência ao permitirem que modelos generalizem para novas classes ou tarefas com pouco ou nenhum dado específico, aproveitando o conhecimento adquirido em tarefas anteriores.

Os modelos tradicionais de aprendizado de máquina dependem muito de *datasets* grandes e anotados, que muitas vezes não são práticos de se obter para todas as classes possíveis. O aprendizado *zero-shot* tem como objetivo classificar instâncias de classes não vistas aproveitando as informações semânticas das classes vistas, enquanto o aprendizado *few-shot* amplia esse conceito permitindo que o modelo aprenda com alguns exemplos das novas classes. Este capítulo se aprofunda nas metodologias, nos aplicativos e nos desafios associados à aprendizagem zero e de poucos exemplos.

2.6.1.1 Aprendizado *Zero-Shot*

O aprendizado *zero-shot* visa classificar instâncias de classes que não foram vistas durante a fase de treinamento. Isso é feito por meio da utilização de informações auxiliares, como atributos semânticos ou descrições textuais, para preencher a lacuna entre as classes vistas e não vistas. A ideia central é mapear as classes vistas e não vistas em um espaço semântico compartilhado, permitindo que o modelo infira a classe de uma instância não vista com base em sua representação semântica.

Um dos principais desafios do aprendizado *zero-shot* é o problema de mudança de domínio, em que a distribuição das classes vistas difere significativamente da distribuição das classes não vistas. Isso pode levar a um desempenho ruim da generalização. Além disso, a qualidade do espaço semântico e a função de projeção desempenham um papel

fundamental no sucesso dos modelos de aprendizado *zero-shot* [148].

2.6.1.2 Aprendizado *Few-Shot*

O objetivo do aprendizado *few-shot* é reconhecer novas classes com apenas alguns exemplos rotulados. Esse paradigma é particularmente útil em situações em que a coleta de um grande número de amostras rotuladas é impraticável. Em geral, os modelos de aprendizagem de poucas tentativas são treinados usando técnicas de meta-aprendizagem, em que o modelo aprende a se adaptar rapidamente a novas tarefas com dados limitados [149].

O aprendizado *few-shot* enfrenta vários desafios, incluindo a dificuldade de aprender representações robustas a partir de dados limitados e a necessidade de técnicas eficazes de aumento de dados. Além disso, o desempenho dos modelos de aprendizado *few-shot* pode ser altamente sensível à escolha de exemplos e à qualidade do processo de meta-aprendizagem [150].

2.7 Processamento de Linguagem Natural aplicado a Finanças

A integração das técnicas de PLN na análise financeira facilita a extração de *insights* de grandes quantidades de dados textuais não estruturados, como relatórios de lucros, artigos de notícias e registros regulatórios.

2.7.1 Automatização de Relatórios Financeiros

A automação dos relatórios financeiros por meio do PLN representa um avanço significativo na eficiência e na precisão das divulgações financeiras. [151] destaca que o PLN pode simplificar o processo de relatório automatizando a extração e a classificação de informações relevantes de documentos financeiros. Essa automação não só aumenta a conformidade com as estruturas regulatórias, mas também reduz a possibilidade de erro humano.

Além disso, [152] discute como as técnicas de mineração de texto, quando combinadas com a modelagem de análise financeira, podem gerar *insights* valiosos a partir de divulgações de demonstrações financeiras. Ao empregar métodos como tokenização e análise de sentimentos, os analistas podem pré-processar com eficácia os dados textuais, levando a processos de tomada de decisão mais informados. Essa sinergia entre PLN e relatórios financeiros ressalta o potencial de maior transparência e eficiência no setor financeiro.

2.7.2 Análise de Sentimentos em Mercados Financeiros

A análise de sentimentos, uma aplicação crítica do PLN, desempenha um papel vital na compreensão da dinâmica do mercado e do comportamento do investidor. [9] apresentam o FinBERT-2020, um modelo pré-treinado de representação de linguagem financeira projetado especificamente para mineração de textos financeiros, que apresentou resultados promissores em tarefas de classificação de sentimentos. Esse modelo aproveita técnicas de aprendizagem profunda para analisar sentimentos expressos em notícias financeiras e mídias sociais, fornecendo *insights* que podem influenciar os movimentos do mercado de ações. Além disso, o estudo de [153] enfatiza a capacidade de processar grandes *datasets* de diversas fontes, incluindo mídia social e artigos de notícias, permite uma compreensão mais detalhada das tendências do mercado e do sentimento dos investidores, o que é crucial para uma previsão financeira eficaz.

2.7.3 Previsão Financeira

A aplicação do PLN na previsão financeira ganhou força, especialmente no contexto da previsão de preços de ações. A pesquisa de [154] descreve o surgimento da previsão financeira baseada em linguagem natural (PLN), que utiliza técnicas de PLN para prever tendências do mercado de ações com base em dados textuais. Essa abordagem foi ainda mais refinada por meio da integração de modelos avançados como o RoBERTa, conforme demonstrado na estrutura FAST-SCAN, que combina a análise de sentimentos com a previsão de séries temporais para aumentar a precisão da previsão [155].

2.7.4 Desafios e Direções Futuras

Apesar dos avanços promissores nas aplicações do PLN em finanças, ainda há vários desafios. A necessidade de grandes *datasets* e a eliminação de vieses na análise de sentimentos são fatores críticos que devem ser abordados para aumentar a confiabilidade dos modelos de PLN [153]. Além disso, a integração de abordagens multimodais pode melhorar significativamente a precisão das previsões financeiras [156].

Pesquisas futuras precisam se concentrar no desenvolvimento de modelos de PLN mais sofisticados que possam lidar com as complexidades da linguagem e do contexto financeiro. A exploração de modelos multilíngues de PLN, conforme proposto por [157], também poderia facilitar a análise de dados financeiros em diferentes idiomas, ampliando o escopo das aplicações do PLN em finanças globais.

2.8 Modelos de Linguagem Pré-Treinados

Os modelos de linguagem pré-treinados revolucionaram o campo do PLN, fornecendo estruturas robustas para compreender e gerar linguagem humana. Esses modelos aproveitam grandes quantidades de dados de texto não anotados para aprender representações contextuais da linguagem, que podem ser ajustadas para tarefas específicas. A introdução de modelos como o BERT estabeleceu um novo padrão em PLN, permitindo avanços significativos em várias aplicações, incluindo análise de sentimentos, *question answering* e tradução automática [1], [5], [158].

Os modelos de linguagem pré-treinados são arquiteturas de rede neural que avançaram significativamente no campo do PLN ao aprender representações de linguagem a partir de grandes *corpora* de dados de texto. Esses modelos utilizam técnicas de aprendizado autossupervisionado, em que preveem partes dos dados de entrada com base em outras partes, o que lhes permite capturar padrões e estruturas complexos dentro do idioma. Por exemplo, o BERT emprega uma técnica de modelagem de linguagem mascarada, em que palavras aleatórias em uma frase são mascaradas, e o modelo aprende a prever essas palavras mascaradas com base no contexto ao redor. Essa abordagem bidirecional permite que o modelo compreenda o contexto tanto do lado esquerdo quanto do lado direito, aprimorando seus recursos de compreensão [159], [160]. A eficácia desses modelos foi demonstrada em várias tarefas de PLN, alcançando o desempenho mais avançado em áreas como classificação de texto, *question answering* e resumo de textos [161], [162].

Os modelos de linguagem pré-treinados alteraram fundamentalmente o cenário das tarefas de compreensão e geração de linguagem: antes do advento dos modelos de linguagem pré-treinados, os sistemas de PLN geralmente dependiam muito de recursos artesanais e arquiteturas específicas de tarefas, o que limitava sua escalabilidade e adaptabilidade. Em contrapartida, os modelos de linguagem pré-treinados permitem uma abordagem mais generalizada, em que um único modelo pode ser ajustado para várias tarefas com modificações mínimas. Essa mudança é destacada pelo sucesso de modelos como o BERT e o GPT, que alcançaram o desempenho mais avançado em uma variedade de tarefas de PLN ao aprender representações genéricas e latentes de linguagem a partir de grandes *corpora* [159], [160]. A capacidade de pré-treinar grandes quantidades de dados de texto e, em seguida, fazer o *fine-tuning* em tarefas específicas possibilitou avanços significativos em áreas como resumo de texto, *question answering* e classificação de texto [161].

Uma das vantagens mais notáveis dos modelos de linguagem pré-treinados é sua capacidade de apresentar bom desempenho mesmo com dados rotulados limitados. As abordagens tradicionais de aprendizado de máquina geralmente exigem *datasets* rotulados extensos para obter um desempenho satisfatório. No entanto, os modelos de linguagem pré-treinados, com base em seu pré-treinamento em *corpora* grandes e não anotados, podem aproveitar esse pré-treinamento extenso para generalizar de forma eficaz para novas tarefas,

geralmente exigindo apenas alguns exemplos ao executar tarefas posteriores. Esse recurso é particularmente vantajoso em domínios em que os dados rotulados são escassos, como idiomas com poucos recursos ou campos especializados, como o processamento de textos biomédicos. Por exemplo, os modelos de linguagem pré-treinados demonstraram melhorias significativas nas tarefas de reconhecimento de entidades nomeadas (NER) com poucos recursos, aproveitando as representações pré-treinadas e o *fine-tuning* em dados mínimos no domínio [163].

Além disso, os modelos de linguagem pré-treinados demonstraram desempenho notável em uma ampla gama de *benchmarks* e tarefas. Por exemplo, o BERT obteve resultados de última geração em várias tarefas de PLN, incluindo o Stanford Question Answering Dataset (SQuAD) e o *benchmark* General Language Understanding Evaluation (GLUE) [1]. Esse desempenho é atribuído aos ricos *embeddings* contextuais aprendidos durante a fase de pré-treinamento, que encapsulam uma riqueza de conhecimento linguístico.

2.8.1 Funcionamento

O processo de desenvolvimento de um modelo de linguagem pré-treinado geralmente envolve duas fases principais: pré-treinamento e *fine-tuning*. Durante a fase de pré-treinamento, o modelo é exposto a um grande *corpus* de texto, onde aprende a prever palavras mascaradas e a entender estruturas de frases sem rótulos específicos da tarefa. Essa fase é fundamental, pois permite que o modelo desenvolva uma compreensão básica da linguagem.

Uma vez concluído o pré-treinamento, o modelo entra na fase de *fine-tuning*, em que é adaptado a tarefas específicas usando *datasets* menores e rotulados. Essa fase envolve o ajuste dos parâmetros do modelo para otimizar o desempenho na tarefa de destino, como a classificação de sentimentos ou o reconhecimento de entidades nomeadas. A flexibilidade dessa abordagem de duas fases é um dos principais motivos da ampla adoção dos modelos de linguagem pré-treinados na comunidade de PLN.

2.8.2 Modelos de Domínio Misto e de Domínio Específico

A distinção entre modelos de domínio misto e específico no pré-treinamento reflete abordagens diferentes para otimizar a performance em tarefas de Processamento de Linguagem Natural, dependendo da necessidade de generalização ampla ou de especialização em um campo específico.

Os modelos de linguagem pré-treinados em domínios mistos são pré-treinados em diversos *corpora* que abrangem vários domínios, permitindo que eles desenvolvam uma ampla compreensão linguística. Esse recurso de generalização é benéfico para tarefas que exigem conhecimento de vários campos, pois permite que o modelo aproveite informações

de diferentes contextos. Ao serem expostos a uma ampla gama de padrões e terminologias linguísticas durante o pré-treinamento, esses modelos podem lidar de forma eficaz com uma variedade de tarefas sem a necessidade de ajustes extensivos específicos do domínio. Essa ampla abordagem de pré-treinamento garante que os modelos não sejam excessivamente especializados, mantendo assim a flexibilidade e a adaptabilidade em diferentes aplicativos.

Os modelos de linguagem pré-treinados específicos do domínio são adaptados a campos específicos por meio do pré-treinamento em *datasets* representativos desse domínio. Essa abordagem é particularmente eficaz em áreas especializadas, como o processamento de textos biomédicos, em que a linguagem e a terminologia podem diferir significativamente do uso geral da linguagem. O pré-treinamento direcionado permite que esses modelos capturem o conhecimento específico do domínio e os padrões linguísticos que seriam ignorados pelos modelos de domínio misto. Ao se concentrarem em *corpora* específicos do domínio, esses modelos podem obter ganhos substanciais no desempenho, pois estão mais bem equipados para entender e processar o vocabulário e o contexto exclusivos do domínio. Esse método não só aumenta a capacidade do modelo de executar tarefas específicas dentro do domínio, mas também reduz a necessidade de um *fine-tuning* extenso, tornando-o uma abordagem mais eficiente e eficaz para aplicativos especializados.

2.8.3 Desafios e Limitações

Apesar dos avanços notáveis trazidos pelos modelos de linguagem pré-treinados, vários desafios permanecem. Um problema significativo é o custo computacional associado ao pré-treinamento de modelos grandes, que pode ser proibitivo para organizações e grupos de pesquisa menores [164]. Como resultado, há um interesse crescente no desenvolvimento de técnicas de pré-treinamento mais eficientes que possam reduzir os requisitos de recursos e, ao mesmo tempo, manter os níveis de desempenho. Técnicas como *knowledge distillation* (destilação de conhecimento), processo que envolve a transferência de conhecimento de um modelo grande e complexo (professor) para um modelo menor e mais eficiente (aluno), e a poda (*pruning*), técnica que se concentra na redução do tamanho do modelo por meio da remoção de parâmetros redundantes ou menos importantes, estão sendo exploradas como possíveis soluções para esse problema.

Outro desafio são as implicações éticas do uso de modelos de linguagem pré-treinados, principalmente no que diz respeito às tendências presentes nos dados de treinamento. Estudos demonstraram que os modelos de linguagem pré-treinados podem inadvertidamente aprender e propagar preconceitos sociais, o que pode levar a resultados injustos ou discriminatórios nos aplicativos. Para lidar com esses preconceitos, é necessária uma pesquisa contínua sobre técnicas de *debiasing* e o desenvolvimento de *datasets* de treinamento mais representativos que reflitam diversas perspectivas. Técnicas como o *debiasing* modular, que envolve a atualização de apenas partes específicas do modelo,

têm se mostrado promissoras na atenuação de vieses sem comprometer o desempenho do modelo [165].

2.9 IA Generativa

A IA generativa é um subcampo da inteligência artificial focado na criação de sistemas capazes de gerar novos conteúdos, como imagens, músicas, textos e outros, que se assemelham aos resultados criados por humanos. O princípio central da IA generativa está em sua capacidade de aprender padrões a partir de vastos conjuntos de dados e gerar novos conteúdos que exibam características semelhantes. Isso é obtido por meio de vários modelos e técnicas, incluindo redes adversariais generativas (*generative adversarial networks* — GANs), *autoencoders* variacionais (*variational autoencoders* — VAEs) e modelos baseados na arquitetura *Transformer* [166], [167].

2.9.1 Principais Técnicas

2.9.1.1 Redes Adversariais Generativas

As GANs [168] consistem em duas redes neurais: um gerador e um discriminador. O gerador cria novas instâncias de dados, enquanto o discriminador as avalia. As duas redes são treinadas simultaneamente em um processo competitivo, em que o gerador tem como objetivo produzir dados indistinguíveis dos dados reais, e o discriminador se esforça para diferenciar os dados reais dos gerados, conforme demonstrado pela Figura 2.12 [167].

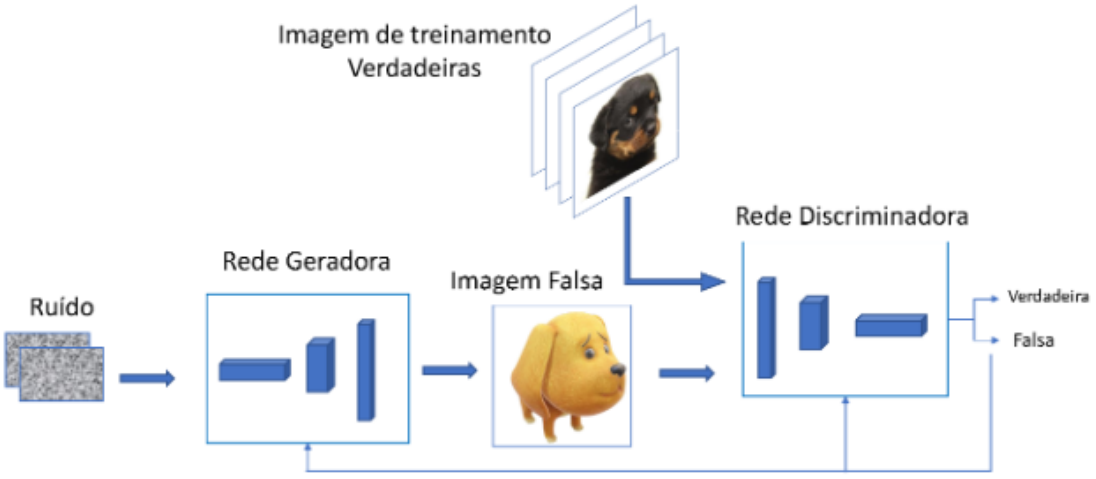

Figura 2.12 – Arquitetura de uma Rede Adversarial Generativa. Extraída de [169].

2.9.1.2 *Autoencoders* Variacionais

Os *Autoencoders* Variacionais (*Variational Autoencoders* — VAEs) utilizam uma arquitetura de codificador-decodificador, demonstrada na Figura 2.13, na qual o codificador

comprime os dados de entrada em um espaço latente, enquanto o decodificador reconstrói os dados a partir dessa representação latente. O diferencial dos VAEs é a introdução de uma abordagem probabilística, onde o espaço latente é modelado como uma distribuição, geralmente gaussiana. Essa característica permite que os VAEs gerem novas amostras semelhantes aos dados de entrada ao realizar amostragem a partir dessa distribuição. Essa habilidade os torna especialmente eficazes em tarefas como geração de imagens e síntese de textos [167].

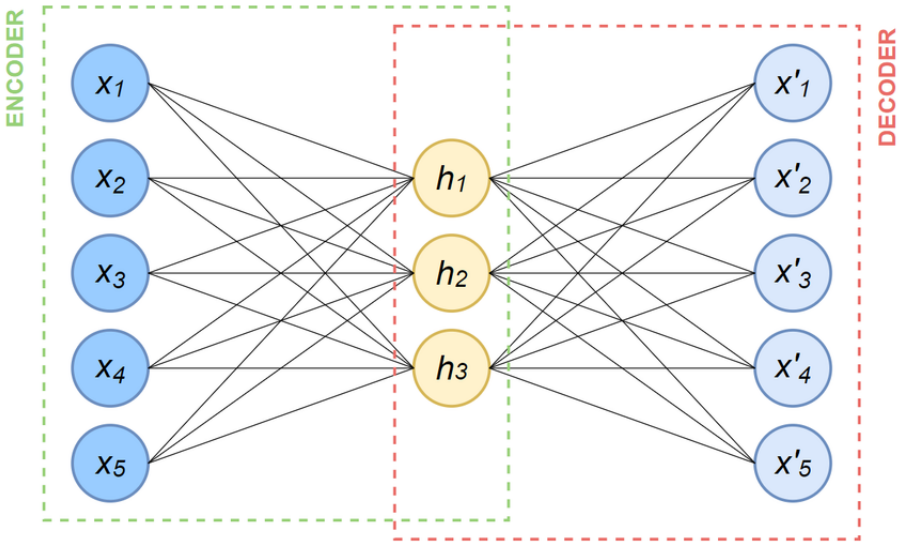

Figura 2.13 – Arquitetura de um *Autoencoder* Variacional. Extraída de [170].

2.9.1.3 *Transformers*

Os *Transformers*, especialmente aqueles usados em grandes modelos de linguagem, revolucionaram a IA generativa ao permitir a geração de textos coerentes e contextualmente relevantes. Esses modelos usam os mecanismos de autoatenção para processar e gerar sequências de dados, o que os torna altamente eficazes para tarefas como geração de texto e processamento de linguagem natural [167], [171].

2.9.2 Inovações

A IA generativa tem apresentado inovações significativas em vários domínios, incluindo síntese de imagens, geração de textos e interações multimodais. Esses avanços levaram ao desenvolvimento de ferramentas e aplicativos de última geração que estão transformando os setores e os campos de pesquisa.

2.9.2.1 Síntese de Imagens

Os recentes avanços na síntese de imagens foram impulsionados por modelos como GANs e VAEs. Esses modelos permitiram a criação de imagens altamente realistas, que

têm aplicações em arte, entretenimento e até mesmo na área da saúde. Por exemplo, as GANs têm sido usadas para gerar imagens médicas sintéticas para fins de treinamento, reduzindo a necessidade de grandes conjuntos de dados anotados [167], [171].

2.9.2.2 Geração de Texto

No âmbito da geração de texto, os modelos baseados em transformadores, como o GPT-3 e o GPT-4, estabeleceram novos padrões de referência. Esses modelos podem gerar texto semelhante ao humano, executar tarefas linguísticas complexas e até mesmo participar de conversas significativas. Suas aplicações vão desde a criação de conteúdo até a automação do atendimento ao cliente [171], [172].

2.9.2.3 Interações Multimodais

A IA generativa multimodal envolve a integração de vários tipos de dados, como texto, imagens e áudio, para criar um conteúdo mais rico e interativo. Essa abordagem levou ao desenvolvimento de ferramentas como o DALL-E [173], que gera imagens a partir de descrições textuais (como a Figura 2.14, gerada pelo DALL-E 3 por meio do *prompt* "Dê-me uma imagem que mostre o contraste entre interessante e desinteressante"), e o Jukebox [174], que cria músicas com base em especificações de gênero e artista. Essas inovações estão expandindo as possibilidades de expressão criativa e geração de conteúdo [175], [176].

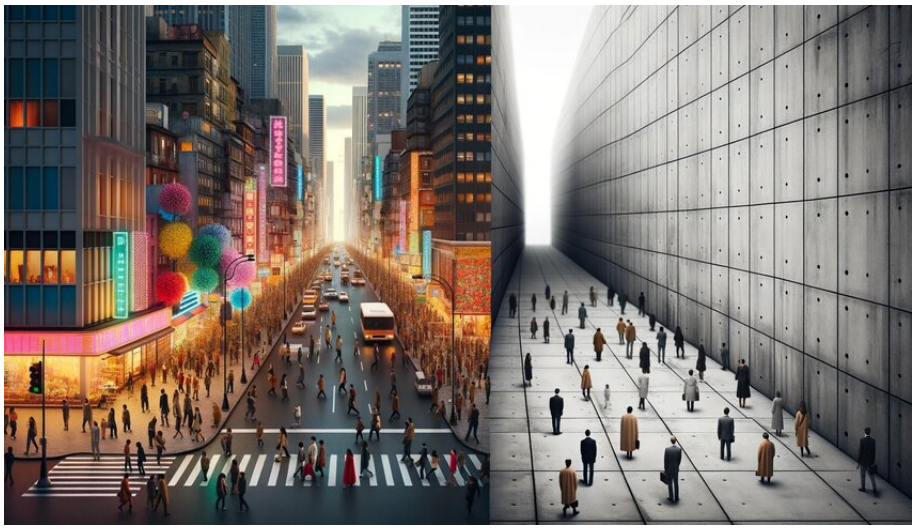

Figura 2.14 – Imagem gerada pelo modelo DALL-E 3. Extraída de [177].

2.9.3 Modelos e Aplicações

- ***Stable Diffusion***: Um modelo que gera imagens de alta qualidade por meio do refinamento iterativo de uma entrada ruidosa [178]. A Figura 2.15 traz um exemplo de imagem gerada por este modelo;

- **Make-A-Video**: Uma ferramenta que gera conteúdo de vídeo a partir de descrições textuais, demonstrando o potencial da IA generativa na produção de vídeo [179];
- **DALL-E**: gera imagens a partir de descrições textuais, permitindo que os usuários criem conteúdo visual com base em sua imaginação;
- **Runway ML**: uma plataforma que oferece várias ferramentas de IA generativas para criar imagens, vídeos e outras mídias, facilitando fluxos de trabalho criativos para artistas e designers [171].

Figura 2.15 – Imagem gerada pelo modelo *Stable Diffusion*. Extraída de [180].

2.9.4 Considerações Éticas e Sociais

À medida que a IA generativa continua a evoluir, torna-se cada vez mais importante abordar suas implicações éticas e sociais. Um aspecto fundamental é garantir o desenvolvimento e a implantação responsáveis de modelos de IA generativa, com foco na atenuação de problemas como parcialidade, justiça e transparência. Além disso, melhorar a interpretabilidade desses modelos é fundamental para compreender seus processos de tomada de decisão e aumentar sua robustez contra ataques adversários. Outra consideração importante envolve a avaliação do impacto da IA generativa em profissões criativas, como arte e música. Isso inclui a exploração de estratégias para integrar ferramentas de IA em fluxos de trabalho humanos de forma a complementar, e não substituir, a criatividade humana [166], [176].

2.10 Modelos de Linguagem Autoregressivos

O desenvolvimento de modelos autorregressivos (MARs) de linguagem tem suas raízes na era inicial da modelagem estatística de linguagem, quando os modelos de n-gramas eram a abordagem dominante. Esses modelos seguiam o princípio de prever a próxima palavra em uma sequência com base nas $n-1$ palavras anteriores, estabelecendo a base para as primeiras tarefas de previsão de linguagem [181]. À medida que o campo avançava, a integração das redes neurais provocou uma mudança significativa, com as redes neurais recorrentes surgindo como o método de referência para a modelagem autorregressiva. As RNNs ganharam popularidade por sua capacidade de processar sequências de comprimento arbitrário e capturar dependências temporais. Posteriormente, as redes LSTM ganharam destaque, solucionando problemas de gradiente decrescente e possibilitando a modelagem de contextos mais longos [92], possibilitando ainda mais avanços nos recursos dos modelos autorregressivos durante seu tempo.

O momento decisivo para os modelos autorregressivos ocorreu com a introdução da arquitetura *Transformer*. Essa arquitetura substituiu as RNNs por mecanismos de autoatenção, permitindo a paralelização durante o treinamento e melhorando significativamente o desempenho em tarefas como a tradução automática. A capacidade da arquitetura *Transformer* de modelar dependências em longas distâncias no texto sem as restrições sequenciais das RNNs levou ao surgimento de modelos como o *Axial Transformer*, que obteve resultados de última geração em *benchmarks* de modelagem generativa [182]. A evolução dos modelos estatísticos tradicionais para os modelos autorregressivos baseados em aprendizagem profunda representa uma mudança de paradigma na forma como a linguagem é processada e gerada, conforme evidenciado pelos avanços significativos em termos de eficiência e precisão em vários aplicativos [183].

2.10.1 Visão Geral da Arquitetura

Os modelos de linguagem autorregressivos geralmente consistem em uma arquitetura de decodificador, em que o modelo gera a sequência de saída um *token* por vez com base na sequência de entrada. O componente principal do decodificador autorregressivo é o mecanismo de autoatenção, que permite que o modelo pondere a importância de diferentes tokens na sequência de entrada ao gerar o próximo *token*. A natureza autorregressiva desses modelos é caracterizada pelo processo de geração, em que cada *token* é produzido com base nos tokens gerados anteriormente. Formalmente, a probabilidade de gerar uma sequência $x = (x_1, x_2, \ldots, x_n)$ pode ser expressa como na Equação 2.15:

$$P(x) = \prod_{i=1}^{n} P(x_i | x_1, x_2, \ldots, x_{i-1}) \tag{2.15}$$

que, expandida, pode ser lida como a Equação 2.16:

$$P(x) = P(x_1) \times P(x_2|x_1) \times P(x_3|x_1, x_2) \times \cdots \times P(x_n|x_1, x_2, \ldots, x_{n-1}) \tag{2.16}$$

Ou seja, a probabilidade total de gerar a sequência inteira é obtida multiplicando as probabilidades condicionais de cada token x_i, dado que todos os *tokens* anteriores $x_1, x_2, \ldots, x_{i-1}$ já foram gerados. Esse processo reflete a natureza autoregressiva dos modelos: cada novo *token* gerado usa como base o histórico dos *tokens* anteriores.

2.10.2 Diferenças em Relação aos Modelos Não Autorregressivos e Bidirecionais

Os modelos autorregressivos diferem fundamentalmente dos modelos não autorregressivos (MNARs) e dos modelos bidirecionais. Os modelos não autorregressivos geram todos os *tokens* em uma sequência simultaneamente, em vez de sequencialmente. Essa abordagem permite melhorias significativas na velocidade durante a inferência, pois elimina a necessidade de geração *token* a *token* [184]. No entanto, os MNARs geralmente têm dificuldades para capturar dependências entre *tokens*, o que pode levar a resultados de menor qualidade em comparação com seus equivalentes autorregressivos [185].

Os modelos bidirecionais, exemplificados pelo BERT, utilizam uma abordagem de modelagem de linguagem mascarada, em que o modelo é treinado para prever *tokens* mascarados com base em seu contexto circundante. Embora isso permita uma compreensão mais abrangente do texto de entrada, ele não se presta a tarefas geradoras da mesma forma que os modelos autorregressivos. A principal distinção está no processo de geração: os modelos autorregressivos geram texto da esquerda para a direita, enquanto os modelos bidirecionais são projetados principalmente para tarefas de compreensão e classificação [186].

A Tabela 2.2 apresenta um sumário das diferenças entre esses modelos.

2.10.3 Métodos de Treinamento

Durante o treinamento, o modelo aprende a maximizar a verosimilhança dos dados de treinamento ajustando seus parâmetros por meio de *backpropagation*. Uma função de perda comumente usada no treinamento de modelo autorregressivos é a log-verossimilhança negativa (ou *Negative Log-Likelihood Loss*, NLLLoss) definida na Equação 2.17 como:

$$L(\theta) = -\sum_{i=1}^{n} \log P(x_i|x_1, x_2, \ldots, x_{i-1}; \theta) \tag{2.17}$$

onde $L(\theta)$ é a função de perda que o modelo tenta minimizar: θ são os parâmetros do modelo (por exemplo, os pesos em uma rede neural); o sinal negativo é aplicado porque

Tabela 2.2 – Comparação entre Modelos Autoregressivos, Não-Autoregressivos e Bidirecionais

Recurso	**MARs**	**MNARs**	**Modelos Bidirecionais**
Método de Geração	Gera sequencialmente um *token* de cada vez	Gera todos os *tokens* simultaneamente	Gera representações para todos os *tokens* simultaneamente
Compreensão Contextual	Unidirecional (da esquerda para a direita)	Tipicamente unidirecional	Bidirecional
Eficiência	Mais lento devido à natureza sequencial	Geração mais rápida	Não projetado para geração
Aplicações	Geração de texto, sistemas de diálogo	Tradução em tempo real	Classificação e compreensão de texto

é usada a log-verossimilhança negativa: minimizar essa função é equivalente a maximizar a verossimilhança da sequência, ou seja, aumentar a probabilidade de o modelo gerar a sequência correta de *tokens*; $\sum_{i=1}^{n}$ denota a soma sobre todos os *tokens* da sequência de entrada $x = (x_1, x_2, \ldots, x_n)$, onde n é o número total de *tokens*; $P(x_i|x_1, x_2, \ldots, x_{i-1}; \theta)$ é a probabilidade condicional de gerar o *token* x_i, dado o histórico de *tokens* anteriores $(x_1, x_2, \ldots, x_{i-1})$, condicionada pelos parâmetros θ do modelo; $\log P(x_i|x_1, x_2, \ldots, x_{i-1}; \theta)$ é o logaritmo da probabilidade condicional, usado para evitar problemas numéricos e tornar a função de perda mais estável e diferenciável.

Na prática, os modelos autorregressivos são treinados em grandes *corpora* de dados de texto, o que lhes permite aprender padrões e estruturas complexas inerentes à linguagem humana. O processo de treinamento envolve alimentar o modelo com sequências de *tokens* e ajustar seus parâmetros com base nas discrepâncias entre os *tokens* previstos e os reais. Esse processo iterativo continua até o modelo convergir para um estado em que possa gerar um texto que se assemelhe aos dados de treinamento.

Várias metodologias foram desenvolvidas para aprimorar o desempenho de modelos autorregressivos, incluindo aprendizagem curricular e estruturas professor-aluno. De acordo com a aprendizagem curricular, a exposição do modelo a tarefas cada vez mais complexas pode aumentar a eficácia do treinamento, aumentando gradualmente a dificuldade das tarefas nas quais o modelo é treinado [187]. Nas estruturas professor-aluno, modelos menores (alunos) são treinados para imitar os resultados de modelos maiores e mais complexos (professores), o que ajuda a destilar o conhecimento e melhorar a eficiência do modelo do aluno sem comprometer significativamente o desempenho [188]. De acordo com o treinamento conjunto autorregressivo e não autorregressivo, a combinação de métodos de treinamento autorregressivo e não autorregressivo pode aprimorar o desempenho do modelo em ambos os modos e aliviar as discrepâncias de distribuição [189].

2.10.4 Geração de Texto

O processo de geração de texto em modelos autorregressivos é inerentemente sequencial. Ao receber uma entrada inicial, o modelo gera o primeiro *token* com base nas probabilidades aprendidas. Esse *token* é então anexado à sequência de entrada, e o processo se repete até que um critério de parada seja atendido, como a geração de um *token* de fim de sequência [190]. A geração pode ser influenciada por várias estratégias de amostragem, que determinam como o próximo *token* é selecionado a partir da distribuição prevista.

Diversas estratégias de amostragem podem ser empregadas durante a geração de texto para equilibrar diversidade e coerência: *greedy sampling*, que seleciona a palavra com maior probabilidade a cada passo; *beam search*, que mantém várias sequências candidatas e seleciona a mais provável; *top-k sampling*, que produz amostras a partir das k palavras mais prováveis; e *nucleus sampling*, que produz amostras a partir do menor conjunto de palavras cuja probabilidade cumulativa excede um limite (p).

2.10.5 Aplicações e Limitações

Os modelos de linguagem autorregressivos têm encontrado aplicações em uma ampla gama de domínios, incluindo tradução automática, resumo de texto e agentes de conversação. Na tradução automática, modelos como o GPT-3 demonstraram capacidades notáveis na tradução de textos entre idiomas, aproveitando seu treinamento extensivo em *corpora* multilíngues [191]. Esses modelos podem alcançar uma qualidade de tradução competitiva, especialmente para idiomas com muitos recursos, e podem ser aprimorados ainda mais por meio de abordagens híbridas que os combinam com outros sistemas de tradução [192]. Da mesma forma, na sumarização de textos, os modelos autorregressivos podem condensar documentos extensos em resumos concisos, mantendo as informações essenciais, demonstrando sua capacidade de gerar textos coerentes e contextualmente relevantes [193].

No âmbito dos agentes de conversação, os modelos autorregressivos permitem a geração de respostas contextualmente relevantes em sistemas de diálogo, aprimorando a experiência do usuário por meio de interações naturais. Esses modelos, que preveem a próxima palavra em uma sequência com base no contexto anterior, demonstraram ser eficazes na imitação de padrões de conversação semelhantes aos humanos, prevendo e ajustando-se continuamente ao fluxo do diálogo [194]. Além disso, os modelos autorregressivos têm sido empregados em aplicações de escrita criativa, onde ajudam os autores a gerar ideias ou até mesmo narrativas inteiras. Essa versatilidade é demonstrada por sua capacidade de gerar textos coerentes e contextualmente apropriados, tornando-os ferramentas valiosas em vários contextos criativos e práticos [195]. A adaptabilidade desses modelos em diferentes domínios ressalta seu potencial para aprimorar as interações com os usuários e apoiar os

processos criativos.

Apesar de seu sucesso, os modelos autorregressivos têm várias limitações. Um problema significativo é a velocidade de inferência; a natureza sequencial da geração autorregressiva pode ser lenta, principalmente para sequências longas, pois cada *token* deve ser gerado um após o outro [196]. Além disso, esses modelos são propensos ao acúmulo de erros, em que os erros nas previsões iniciais podem se propagar e se amplificar nas etapas subsequentes, levando a um desempenho degradado ao longo do tempo [197]. Para enfrentar alguns desses desafios, foram propostos modelos híbridos que combinam abordagens autorregressivas e não autorregressivas, com o objetivo de aproveitar os pontos fortes de ambos os métodos e, ao mesmo tempo, atenuar seus respectivos pontos fracos [198].

2.11 Grandes Modelos de Linguagem — LLMs

Os grandes modelos de linguagem (*large language models* — LLMs) revolucionaram o campo do PLN desde seu início, especialmente após o lançamento do ChatGPT em novembro de 2022. Esses modelos aproveitam grandes quantidades de dados de texto e arquiteturas complexas para executar uma ampla gama de tarefas de linguagem, incluindo geração de texto, resumo, tradução e muito mais.

2.11.1 Introdução

O advento dos LLMs representou um marco significativo na inteligência artificial, permitindo que as máquinas compreendam e gerem textos semelhantes aos humanos. Esses modelos são criados com base nos princípios de aprendizagem profunda e arquiteturas de transformadores, o que lhes permite capturar padrões complexos em dados de linguagem. O processo de treinamento envolve o ajuste de bilhões de parâmetros com base em vastos conjuntos de dados, o que leva a modelos capazes de compreender a linguagem para fins gerais.

Os LLMs, como a série GPT da OpenAI, demonstraram capacidades notáveis em tarefas de processamento de linguagem natural e muito mais. Esses modelos são projetados para capturar os padrões e as estruturas estatísticas presentes nos dados de treinamento, o que lhes permite gerar respostas coerentes e contextualmente relevantes [199], [200]. O sucesso dos LLMs levou a um grande fluxo de contribuições de pesquisa, abrangendo diversos tópicos, como inovações arquitetônicas, melhores estratégias de treinamento, melhorias no comprimento do contexto, *fine-tuning*, LLMs multimodais e muito mais [201].

Uma das características mais notáveis dos LLMs é sua capacidade de generalização em várias tarefas. Por exemplo, modelos como o GPT-3 e o LaMDA [202] podem manter diálogos com humanos sobre muitos tópicos após uma preparação mínima com alguns

exemplos [203]. Essa versatilidade é um passo mais próximo dos extraordinários recursos da linguagem humana, permitindo que esses modelos executem tarefas que antes exigiam modelos de rede separados. A capacidade dos LLMs de entender e gerar textos semelhantes aos humanos gerou uma imaginação sem limites acerca do futuro ambiente de IA para humanos [204].

O rápido desenvolvimento dos LLMs também levou à sua aplicação em vários campos, inclusive em sistemas de recomendação. Ao aproveitar o poder dos LLMs, os pesquisadores aprimoraram a capacidade dos sistemas de recomendação de entender os interesses dos usuários e capturar informações textuais secundárias, resultando em recomendações mais personalizadas e precisas [205]. Essa integração de LLMs em diferentes domínios ressalta seu impacto transformador nos campos de inteligência artificial e aprendizado de máquina.

2.11.2 Principais Famílias

Os modelos GPT (*Generative Pre-trained Transformers*), desenvolvidos pela OpenAI, avançaram significativamente o processamento de linguagem natural, permitindo que as máquinas compreendam e gerem textos semelhantes aos humanos. Esses modelos, especialmente o GPT-3, com seus 175 bilhões de parâmetros, estabeleceram novos padrões de qualidade e versatilidade na geração de texto em várias tarefas. Os modelos GPT são baseados na arquitetura *Transformer* e utilizam o aprendizado não supervisionado de diversos textos da Internet, o que os torna altamente eficazes em tarefas de processamento de linguagem [206]. Apesar de seus recursos impressionantes, os modelos de GPT enfrentam desafios, como imprecisões factuais e a possibilidade de produzir conteúdo tendencioso ou prejudicial devido à natureza de seus dados de treinamento [207]. Além disso, há preocupações quanto à propriedade intelectual e ao plágio, pois a GPT-3 pode gerar conteúdo que não se distingue da escrita humana [208]. Na área da saúde, o potencial da GPT-3 está sendo explorado, mas sua implementação exige uma consideração cuidadosa dos vieses, dos custos operacionais e da conformidade com os regulamentos [209], [210]. O desenvolvimento de modelos como BioGPT e VL-GPT demonstra a expansão dos aplicativos de GPT em domínios especializados, como geração de textos biomédicos e tarefas multimodais, respectivamente [211], [212]. De modo geral, embora os modelos de GPT tenham transformado o processamento de idiomas, são necessárias pesquisas contínuas para abordar suas limitações e explorar todo o seu potencial em vários campos.

Outra família importante é a LLaMA (*Large Language Model Meta AI*) do Meta, que inclui várias iterações projetadas para eficiência e desempenho. O LLaMA [213], o LLaMA 2 [214] e o LLaMA 3 [215] foram desenvolvidos para oferecer um bom desempenho com menos parâmetros, sendo treinados em conjuntos de dados disponíveis publicamente, o que garante uma ampla aplicabilidade. Versões refinadas, como Alpaca e Vicuna, aprimoram

os recursos de acompanhamento de instruções — a capacidade do modelo de seguir comandos ou responder a tarefas específicas com precisão. O Alpaca [216] utiliza métodos de autoinstrução, nos quais o modelo gera dados adicionais com base em exemplos fornecidos, enquanto o Vicuna [217] incorpora o *feedback* do usuário para melhorias iterativas. O Koala [218] e o Mistral [219] exploram o equilíbrio entre tamanho e desempenho, sendo que o Mistral se concentra na eficiência do treinamento e, ao mesmo tempo, mantém a alta precisão da tarefa. O Mixtral [220] combina várias técnicas de treinamento de seus predecessores para otimizar o desempenho em diferentes tarefas de PLN. No domínio biomédico, os modelos LLaMA ajustados por instruções, como os que usam o *dataset* BioInstruct, apresentam ganhos significativos de desempenho em tarefas como *question answering*, extração de informações e geração de texto, superando outros modelos ajustados com dados específicos do domínio [221]. O LLaMA-Adapter apresenta um método de adaptação leve para um *fine-tuning* eficiente, usando demonstrações de autoinstrução e um mecanismo de atenção inicializado a zero, obtendo respostas de alta qualidade comparáveis a modelos totalmente ajustados, como o Alpaca [222]. Para idiomas que não sejam o inglês, foram desenvolvidos métodos para aumentar os recursos do LLaMA, como a extensão de seu vocabulário para textos, por exemplo, em chinês, o que aumenta significativamente sua proficiência na compreensão e geração de conteúdo nesse idioma [223]. Em aplicações médicas, o AlpaCare, um modelo LLaMA ajustado usando um conjunto de dados de instruções médicas diversas, demonstra desempenho superior em tarefas médicas e *benchmarks* de domínio geral, superando as linhas de base existentes em termos de correção e utilidade [224]. É possível notar, portanto, que esses modelos e adaptações destacam os esforços contínuos para equilibrar eficiência, desempenho e aplicabilidade em vários domínios e idiomas.

O PaLM (*Pathways Language Model*), desenvolvido pelo Google [225], é um avanço significativo em modelos de linguagem de grande porte, com 540 bilhões de parâmetros e utilizando o sistema *Pathways* para aprender de forma eficiente com várias tarefas simultaneamente. Esse modelo é excelente em *few-shot learning*, obtendo resultados de última geração em vários *benchmarks* de geração e compreensão de linguagem, e demonstra um bom desempenho em tarefas de raciocínio, muitas vezes superando o desempenho humano em determinados *benchmarks* [226]. Os recursos multilíngues do PaLM são notáveis, com um desempenho impressionante de tradução automática, embora ainda fique atrás dos sistemas supervisionados de última geração [227], [228], [229]. Apesar de seus pontos fortes, o tamanho do modelo resulta em custos computacionais substanciais, levantando preocupações sobre acessibilidade e impacto ambiental [225]. Os esforços para reduzir esses custos incluem métodos como o UL2R, que melhoram a eficiência do dimensionamento, obtendo desempenho semelhante com recursos computacionais reduzidos [230]. Além disso, o PaLM 2, um sucessor do PaLM, oferece recursos multilíngues e de raciocínio aprimorados com maior eficiência computacional, permitindo uma implantação mais ampla e uma

interação mais rápida [231].

O Bard Gemini, um modelo de IA de conversação desenvolvido pelo Google [232], integra recursos multimodais, permitindo que ele processe textos e imagens, aumentando assim sua capacidade de entender o contexto e criar interações mais envolventes com o usuário. Esse modelo, anteriormente conhecido como Google Bard, foi avaliado por seu desempenho, usabilidade e recursos de integração, com foco em implicações éticas, como privacidade e segurança de dados. Ele utiliza métodos avançados de aprendizado de máquina, incluindo transformadores e processamento multimodal, para melhorar sua compreensão e geração de linguagem natural em diversos contextos. Apesar de seus recursos avançados, o Bard Gemini pode apresentar limitações: em ambientes clínicos, embora gere respostas que possam parecer confiáveis, notou-se que ele fabrica citações e resumos, levantando preocupações sobre sua confiabilidade como fonte de informações clínicas [233]. Além disso, as habilidades multimodais do Bard Gemini foram avaliadas por meio de várias tarefas visuais, revelando sua proficiência em algumas áreas, como a resolução de CAPTCHAs visuais, mas também destacando limitações em tarefas que exigem análise visual detalhada [234].

2.12 Técnicas de Otimização e Regularização de Modelos de Linguagem

O advento dos modelos de linguagem pré-treinados revolucionou o campo do PLN, permitindo que os modelos aproveitem grandes quantidades de dados para melhorar o desempenho em várias tarefas. No entanto, a eficácia desses modelos é significativamente influenciada pelas técnicas de otimização e regularização. Essas técnicas são essenciais para aprimorar o desempenho e a generalização dos modelos de linguagem pré-treinados, principalmente atenuando o excesso de ajuste e garantindo que os modelos aprendam padrões relevantes dos dados. Este capítulo se aprofunda nos meandros da otimização e da regularização em modelos de linguagem pré-treinados, explorando várias técnicas e suas implicações para o desempenho do modelo.

2.12.1 Otimização

A otimização no contexto dos modelos de linguagem pré-treinados refere-se ao processo de ajuste dos parâmetros do modelo para minimizar uma função de perda durante o treinamento. A escolha do algoritmo de otimização pode afetar muito a velocidade de convergência e a qualidade do modelo final. Por exemplo, o cenário de otimização dos modelos de linguagem pré-treinados geralmente é complexo devido à alta dimensionalidade do espaço de parâmetros e à natureza não convexa das funções de perda envolvidas. Conforme observado por [235], o paradigma de pré-treinamento e *fine-tuning* se tornou

uma abordagem padrão em PLN, em que os modelos são primeiro pré-treinados em grandes *corpora* e depois ajustados em tarefas específicas. Esse processo de duas etapas exige estratégias de otimização eficazes para garantir que o modelo retenha o conhecimento adquirido durante o pré-treinamento e, ao mesmo tempo, se adapte às nuances da tarefa de destino.

Para evitar o ajuste excessivo ao fazer o *fine-tuning* de modelos de linguagem pré-treinados em *datasets* menores, é fundamental implementar estratégias como o *early stopping* (parada antecipada), que ajuda a evitar que o modelo memorize os dados de treinamento, interrompendo o processo de treinamento quando o desempenho em um conjunto de validação começa a se degradar, garantindo assim uma melhor generalização para dados não vistos [236]. Na literatura, é também relatado que o uso de métodos de *fine-tuning* adaptativos que ajustam dinamicamente o número de épocas de treinamento e as taxas de aprendizagem com base no tamanho do *dataset* demonstrou melhorar o desempenho, a estabilidade e a eficiência, principalmente para *datasets* pequenos [237].

2.12.1.1 Adam e AdamW

Os otimizadores Adam e AdamW são fundamentais no cenário da aprendizagem profunda, especialmente no âmbito do PLN. O Adam, apresentado por [238], é um algoritmo de otimização da taxa de aprendizagem adaptativa que calcula taxas de aprendizagem adaptativas individuais para diferentes parâmetros a partir de estimativas de primeiro e segundo momentos dos gradientes, o que permite o treinamento eficiente de redes neurais profundas, especialmente em cenários com grandes *datasets* e parâmetros. Esse método é particularmente benéfico em tarefas de PLN, em que a complexidade e o tamanho dos dados podem levar a desafios de convergência e estabilidade durante o treinamento.

O AdamW [239], uma variante do otimizador Adam, foi desenvolvido para tratar de algumas das limitações associadas ao algoritmo Adam original, principalmente no que se refere à forma como o decaimento do peso é aplicado. No Adam original, o decaimento do peso é incorporado diretamente à atualização do gradiente, o que pode levar a interações não intencionais entre a taxa de aprendizado e o decaimento do peso, o que pode afetar o processo de otimização. O AdamW desacopla esses dois componentes, aplicando a redução de peso separadamente da atualização do gradiente. Essa dissociação permite uma regularização mais eficaz e um melhor desempenho de generalização, o que é especialmente benéfico em aplicações de PLN, em que o ajuste excessivo é uma preocupação significativa devido à alta dimensionalidade dos dados de texto e à complexidade dos modelos de linguagem. Estudos demonstraram que o AdamW pode levar a um melhor desempenho e a uma convergência mais rápida em várias tarefas de aprendizagem profunda em comparação com o otimizador Adam original [240], [241]. Além disso, a aplicação do AdamW no PLN não se limita apenas à eficiência do treinamento; ele também aumenta a capacidade do

modelo de generalizar em diferentes tarefas. Estudos também demonstraram que os modelos treinados com AdamW apresentam melhor desempenho em dados fora da distribuição, o que é crucial para aplicações do mundo real em que o modelo pode encontrar distribuições de dados não vistas [242].

2.12.1.2 RAdam

O otimizador RAdam (*Rectified Adam*) [243] representa um avanço notável nos algoritmos de otimização, especialmente para aplicações de aprendizagem profunda, como o PLN. O RAdam aborda as limitações dos otimizadores tradicionais, como o Adam, aprimorando a estabilidade da convergência e o desempenho em várias tarefas. Sua principal inovação é um mecanismo de taxa de aprendizagem adaptável que retifica as taxas de aprendizagem com base na variação do gradiente, levando a um processo de treinamento mais estável e confiável. Esse recurso é especialmente benéfico em tarefas de PLN, em que o desempenho do modelo é altamente sensível à escolha do otimizador e de seus hiperparâmetros. Estudos demonstraram que o RAdam geralmente supera o Adam e outras variantes em termos de estabilidade e eficiência, o que a torna a escolha preferida para muitas aplicações de aprendizagem profunda [244], [245].

No contexto do PLN, foi demonstrado que o RAdam aprimora o treinamento de várias arquiteturas, incluindo *Transformers* e redes neurais recorrentes. A capacidade do otimizador de manter um equilíbrio entre a exploração e o aproveitamento durante o processo de treinamento permite que ele converse com mais rapidez e eficácia do que seus antecessores. Por exemplo, estudos demonstraram que o RAdam supera os otimizadores tradicionais em tarefas como sumarização de texto, em que a complexidade dos dados linguísticos pode levar a desafios no treinamento de modelos [246].

2.12.1.3 AdamP

O otimizador AdamP é uma variante do otimizador Adam projetado para lidar com o decaimento prematuro de tamanhos de etapas efetivas em otimizadores de descida de gradiente baseados em momentum quando aplicados a pesos invariantes de escala. Esse problema surge porque a combinação de *momentum* e invariância de escala leva a uma rápida redução nos tamanhos de etapas efetivas, o que pode resultar em um desempenho de modelo abaixo do ideal. O AdamP atenua esse problema removendo o componente radial, ou a direção de aumento da norma, em cada etapa do otimizador. Esse ajuste mantém as direções de atualização efetivas enquanto altera os tamanhos das etapas, preservando assim as propriedades originais de convergência dos otimizadores de descida de gradiente. Avaliações empíricas em vários *benchmarks*, incluindo visão computacional, modelagem de linguagem e tarefas de classificação de áudio, demonstraram que o AdamP melhora consistentemente o desempenho em relação ao Adam [247].

2.12.1.4 MADGRAD

O MADGRAD (*Momentumized, Adaptive, Dual Averaged GRADient*) [248] é um método de otimização que combina adaptatividade com forte desempenho de generalização para tarefas de aprendizado profundo. Diferentemente de outros métodos adaptativos como o Adam, que às vezes têm desempenho inferior ao SGD em problemas de classificação de imagens, o MADGRAD mantém excelente performance em diversos domínios de aprendizado profundo. O método é construído sobre a formulação de *dual averaging* do AdaGrad [249], aprimorado através de várias modificações-chave: uma sequência de gradiente ponderada para contrapor a sequência de tamanho do passo, a incorporação de *momentum* e um denominador de raiz cúbica que lida melhor com gradientes iniciais grandes. Em *benchmarks* abrangendo visão computacional, reconstrução de imagens de ressonância magnética e processamento de linguagem natural, o MADGRAD consistentemente iguala ou supera tanto o SGD quanto o Adam, demonstrando que métodos adaptativos podem ser eficazes sem sacrificar a capacidade de generalização.

2.12.2 Regularização

A regularização é um aspecto fundamental do treinamento de modelos de linguagem pré-treinados, com o objetivo de evitar o excesso de ajuste e aumentar a robustez do modelo. Várias técnicas de regularização foram propostas, cada uma com sua abordagem exclusiva para restringir a complexidade do modelo. Um método amplamente usado é o *Dropout*, que desativa aleatoriamente um subconjunto de neurônios durante o treinamento para evitar a coadaptação [250]. Essa técnica incentiva o modelo a aprender recursos mais robustos que não dependem de nenhum subconjunto específico de neurônios, melhorando assim a generalização.

O decaimento de pesos (*weight decay*) é outra estratégia de regularização eficaz que penaliza pesos grandes em um modelo, desencorajando, assim, o modelo a se ajustar ao ruído nos dados de treinamento e promovendo modelos mais simples que se generalizam melhor para dados não vistos. Essa técnica é particularmente relevante para modelos de linguagem pré-treinados, que têm um grande número de parâmetros e são propensos a se ajustar demais se não forem gerenciados adequadamente [251]. Além disso, o treinamento contraditório surgiu como uma técnica de regularização poderosa, em que os modelos são treinados para serem robustos contra perturbações nos dados de entrada. Esse método aumenta a capacidade de generalização do modelo em diferentes domínios e reduz a sensibilidade a exemplos adversos [252].

2.12.2.1 Reinicialização de Camadas

Os pesquisadores descobriram que os modelos baseados em *Transformers* podem produzir resultados significativamente diferentes quando diferentes sementes aleatórias são

usadas, sendo a inicialização do peso particularmente sensível à escolha da semente [236]. Várias técnicas foram introduzidas para lidar com essa instabilidade, como a reinicialização de camadas e a regularização *Mixout*.

O conceito de reinicialização de camadas, uma descoberta importante na pesquisa de visão computacional, sugere que as camadas inferiores pré-treinadas normalmente adquirem recursos mais gerais. Por outro lado, as camadas mais altas, que estão mais próximas da saída, geralmente se concentram em tarefas específicas de pré-treinamento [253]. Pesquisas recentes sobre *Transformers* revelaram que, embora a utilização de toda a rede possa ser a abordagem mais eficaz em alguns casos, ela pode dificultar o processo de treinamento e afetar negativamente o desempenho geral [254]. Esse *insight* levou a abordagens mais matizadas no *fine-tuning*, em que diferentes camadas podem ser tratadas de forma diferente com base em sua função no modelo.

2.12.2.2 Regularização *Mixout*

Mixout é uma técnica de regularização projetada para melhorar o *fine-tuning* de modelos de linguagem pré-treinados em grande escala. Diferentemente dos métodos tradicionais, como *DropConnect* [255] e *Dropout* [250], que definem os parâmetros como zero com uma certa probabilidade, o *Mixout* substitui os parâmetros por seus valores pré-treinados durante o treinamento com uma probabilidade denotada por p.

É possível ver, da esquerda para a direita, na Figura 2.16 o seguinte: primeiro há uma rede neural com seus valores pré-treinados originais (indicados pelas arestas vermelhas); depois, uma rede neural com *Dropout*, onde há um certo número de pesos que são zerados (indicados pelas arestas pontilhadas). Além disso, as arestas de cor preta representam os valores que foram alterados em relação, aos valores pré-treinados originais, mediante tarefas de *fine-tuning*; por fim, uma rede neural com *Mixout*, a qual, após o *fine-tuning*, manteve alguns valores de pesos originais da rede neural quando pré-treinada.

Essa abordagem ajuda a manter a estabilidade e a precisão do modelo durante a fase de *fine-tuning*. As evidências empíricas sustentam que o *Mixout* melhora o desempenho dos modelos pré-treinados, principalmente quando a quantidade de dados de treinamento é limitada. Os pesquisadores demonstraram que o *Mixout* aumenta a estabilidade e a precisão média de modelos como o BERT em várias tarefas de PLN [256].

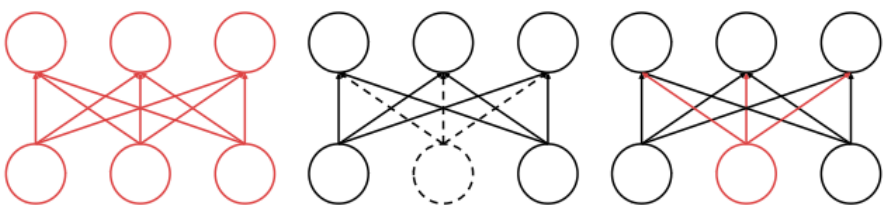

Figura 2.16 – Ilustração da Regularização *Mixout*. Extraída de [256].

2.12.2.3 Média Estocástica dos Pesos

A Média Estocástica dos Pesos (*Stochastic Weight Averaging* — SWA) [257] é uma técnica de otimização que aprimora a generalização das redes neurais, calculando a média dos pesos dos modelos amostradas em diferentes pontos durante o treinamento. Esse método estimula a convergência para mínimos mais regulares, que estão associados a um melhor desempenho de generalização [258]. No contexto dos modelos de processamento de linguagem natural, foi demonstrado que o SWA melhora a generalização sem custos computacionais adicionais, superando os métodos tradicionais como a destilação de conhecimento [259]. Ela também ajuda a melhorar a robustez contra mudanças de distribuição, particularmente em modelos de linguagem grandes e *fine-tuning* em *datasets* pequenos e, ao explorar ótimos mais amplos, ajuda a obter previsões mais confiáveis e precisas em tarefas de PLN [260], [261].

2.12.2.4 Decaimento da Taxa de Aprendizado por Camada

Uma técnica adicional para regularizar os modelos de linguagem pré-treinados é o *fine-tuning* discriminativo, que envolve a atribuição de diferentes taxas de aprendizado a várias camadas do modelo durante o treinamento. Isso ocorre porque diferentes camadas capturam diferentes tipos de informações [253], [262]. O *fine-tuning* discriminativo, também conhecido como decaimento da taxa de aprendizado por camada (*layer-wise learning rate decay* — LLRD), implementa uma estratégia de uso de taxas de aprendizado mais altas para as camadas superiores e taxas mais baixas para as camadas inferiores ao ajustar os modelos de linguagem pré-treinados. Essa abordagem visa modificar as camadas superiores relacionadas à tarefa de pré-treinamento e, ao mesmo tempo, manter as camadas inferiores que contêm informações mais generalizadas [254]. Pesquisas recentes apoiam a eficácia da LLRD no aprimoramento do desempenho do modelo em várias tarefas [263], [264]; ao implementar uma taxa de aprendizado mais alta para a camada superior e diminuir progressivamente a taxa de aprendizado para cada camada subsequente em direção à base, a LLRD permite que o modelo equilibre o *fine-tuning* específico da tarefa nas camadas superiores e, ao mesmo tempo, preserve o conhecimento geral codificado nas camadas inferiores. Essa abordagem foi empregada com sucesso em modelos de linguagem pré-treinados como XLNet [265] e ELECTRA [263], demonstrando sua eficácia na adaptação a novas tarefas e, ao mesmo tempo, retendo conhecimentos valiosos da fase de pré-treinamento.

2.13 Considerações Finais

Neste capítulo, foram discutidos diversos conceitos fundamentais relacionados à Inteligência Artificial (IA) e suas aplicações em áreas como Processamento de Linguagem

Natural, Visão Computacional, Robótica, Saúde, Finanças, Educação e Transporte. O entendimento dos avanços e limitações presentes na literatura sobre esses temas revelou-se crucial para o desenvolvimento da tese.

Os *insights* obtidos a partir dos estudos revisados não apenas destacaram o potencial transformador da IA em múltiplos setores, mas também evidenciaram os desafios éticos e práticos que acompanham sua implementação. A evolução da IA, desde suas origens até as técnicas de aprendizagem profunda atuais, demonstra a importância de um conhecimento abrangente sobre as tecnologias envolvidas e suas implicações.

Essas considerações fornecem uma base sólida para a continuidade da pesquisa, permitindo uma análise crítica das abordagens existentes e a identificação de novas oportunidades para inovação e melhoria nos processos. Assim, o conhecimento adquirido neste capítulo servirá como um importante insumo para as etapas subsequentes da tese.

3 Estado da Arte

Este capítulo oferece uma análise detalhada do estado da arte, incluindo uma revisão crítica das contribuições científicas mais recentes e relevantes. Além disso, realiza um estudo comparativo entre as obras mais significativas, culminando em uma síntese dos principais pontos discutidos.

3.1 BERT

O BERT é um dos modelos de linguagem mais influentes desenvolvidos pela equipe do Google AI em 2018, introduzido como uma inovação na arquitetura *Transformer*. O modelo revolucionou o PLN ao ser o primeiro a adotar uma abordagem de pré-treinamento bidirecional e não supervisionada, permitindo a construção de representações contextualizadas das palavras em ambos os sentidos.

3.1.1 Arquitetura

A arquitetura do BERT é fundamentalmente construída no modelo *Transformer*, utilizando apenas o componente codificador, que depende exclusivamente de mecanismos de atenção para capturar dependências complexas entre palavras. Esse *design* permite que o BERT modele de forma eficaz o contexto das palavras em uma frase, levando a um desempenho superior em várias tarefas de processamento de linguagem natural [1], [266]. Um dos recursos mais notáveis do BERT é sua bidirecionalidade, que o diferencia dos modelos unidirecionais que processam o texto sequencialmente, da esquerda para a direita ou da direita para a esquerda. Ao considerar o contexto completo de uma palavra em ambas as direções, o BERT pode obter uma compreensão mais detalhada e abrangente dos significados e das relações entre as palavras [267], [268]. Essa abordagem bidirecional tem sido fundamental para o sucesso do BERT em uma ampla gama de aplicações, desde a análise de sentimentos até a sumarização de textos, e o estabeleceu como um novo padrão em modelos de representação de linguagem [269], [270].

O BERT utiliza o mecanismo de autoatenção da arquitetura *Transformer* para calcular o relacionamento entre as palavras em uma sequência de texto, permitindo que o modelo atribua diferentes pesos às palavras dependendo de sua relevância contextual. O cálculo da atenção baseia-se em três vetores principais: consulta (*query*), chave (*key*) e valor (*value*). A pontuação de atenção é determinada pela fórmula dada pela Equação 2.6.

3.1.2 Objetivos de Pré-treinamento

O BERT possui dois objetivos que norteiam seu pré-treinamento: o MLM e o NSP.

No MLM (*Masked Language Modeling*), uma porcentagem das palavras em uma sequência é substituída aleatoriamente por um *token* especial `[MASK]`. O objetivo do modelo é prever essas palavras mascaradas com base no contexto circundante. Para isso, a função softmax transforma as saídas do modelo em uma distribuição de probabilidade sobre todo o vocabulário, ajudando o modelo a identificar a palavra mais provável para substituir o *token* `[MASK]`. A função softmax é expressa pela Equação 2.7, onde z_i é a pontuação *logit* para o *token* i. Essa pontuação é uma estimativa bruta gerada pelo modelo, indicando o "peso" ou "importância" atribuída ao *token* i como possível substituto do `[MASK]`. Em outras palavras, é a saída do modelo antes de ser normalizada pela softmax, representando o quanto o modelo "acredita" que esse *token* específico corresponde ao contexto da posição mascarada; N é o número total de *tokens* no vocabulário. O denominador da equação é a soma das pontuações exponenciais de todos os *tokens*, garantindo que a softmax gere uma distribuição de probabilidade que totaliza 1.

Ao aplicar a softmax, o modelo converte esses *logits* em probabilidades para cada *token* do vocabulário. Assim, ele consegue escolher a palavra com maior probabilidade para preencher o *token* `[MASK]`, com base no contexto bidirecional. Essa abordagem permite que o BERT aprenda representações contextualizadas bidirecionais, o que tem se mostrado altamente vantajoso em tarefas de PLN [271].

O objetivo de *Next Sentence Prediction* (NSP) é o segundo objetivo, projetado para ajudar o modelo a entender as relações entre as frases. Durante o treinamento, o modelo recebe pares de frases e precisa prever se a segunda frase é uma continuação lógica da primeira ou não. Essa tarefa é essencial para tarefas como *question answering* e inferência de linguagem natural [272].

A tarefa MLM do BERT emprega o *Subword Word Masking* (SWM) para mascarar subpalavras (ou *tokens*) aleatoriamente. Por exemplo, em inglês, uma palavra como `Transformers` pode ser dividida em unidades de subpalavras, como `Transform` e `ers`. Embora o SWM tenha demonstrado potencial, é essencial reconhecer seus desafios, como determinar a granularidade ideal das unidades de subpalavras para mascarar e prever. Essa tarefa é complexa e pode exigir uma consideração cuidadosa das características morfológicas do idioma [273].

Várias outras estratégias de mascaramento foram desenvolvidas para melhorar o desempenho dos MLMs:

- **Mascaramento de intervalos (*Span Masking*)**: O SpanBERT amplia o BERT ao mascarar intervalos aleatórios contíguos de texto em vez de *tokens* aleatórios. Esse

método treina as representações do limite do intervalo para prever todo o conteúdo do intervalo mascarado, o que demonstrou ganhos substanciais em tarefas como *question-answering* e resolução de correferências [274];

- **Mascaramento de PMI (PMI-*Masking*)**: O mascaramento de informações mútuas pontuais (*Pointwise Mutual Information* — PMI) envolve a seleção de *tokens* para mascarar com base em suas informações mútuas com outros *tokens* na sequência. Essa estratégia visa mascarar os *tokens* que fornecem o contexto mais informativo para as representações de aprendizado [275];

- **Mascaramento de palavras inteiras (WWM — *Whole Word Masking*)**: Em vez de mascarar *tokens* de subpalavras individuais, o WWM mascara palavras inteiras. Essa estratégia mostrou melhorias significativas em várias tarefas de PLN, fornecendo uma tarefa de previsão mais desafiadora e contextualmente rica [276];

- **Modelagem de linguagem mascarada descritiva (DMLM - *Descriptive Masked Language Modeling*)**: Essa estratégia aprimora o MLM fornecendo uma base semântica explícita. O modelo deve prever a palavra mais provável em um contexto, dada a definição da palavra; esta estratégia difere do padrão adotado pelo BERT porque usa a semântica das palavras, em vez do contexto bidirecional. Por exemplo, dada a frase "Eu estava indo para _", se fosse fornecida como definição "instituição financeira", o modelo teria de prever a palavra "banco"; se, em vez disso, fosse fornecida "litoral", o modelo deveria prever "praia" [277].

3.1.3 Tokenização

A tokenização é uma etapa do pré-processamento no PLN que envolve a divisão do texto em unidades menores, ou *tokens*, que podem ser palavras, subpalavras ou caracteres. Essa etapa é essencial para preparar *tokens* de entrada para modelos de linguagem profunda, pois afeta significativamente o desempenho de tarefas subsequentes de PLN [278], [279], [280]. No contexto do BERT, a tokenização desempenha um papel vital na forma como o modelo interpreta e processa a linguagem. O BERT emprega um método de tokenização específico chamado *WordPiece*, que foi projetado para lidar com uma ampla variedade de vocabulário e gerenciar com eficácia palavras fora do vocabulário (*out-of-vocabulary* — OOV) [281]. Originalmente desenvolvido pelo Google para tradução automática, o *WordPiece* se tornou parte integrante da arquitetura do BERT, permitindo que ele obtivesse resultados notáveis em várias tarefas de PLN, aproveitando uma estratégia de *longest-match-first* (primeira correspondência mais longa) para tokenizar o texto de forma eficiente [282]. Esse método é particularmente benéfico para idiomas morfologicamente ricos, nos quais os métodos tradicionais de tokenização podem ter dificuldades [280], [283].

O processo de tokenização no BERT pode ser dividido em várias etapas:

1. **Pré-tokenização**: A pré-tokenização é uma etapa de pré-processamento em PLN que segmenta o texto bruto em unidades intermediárias para prepará-lo para uma tokenização final eficiente e precisa. Essa segmentação preliminar aborda elementos linguísticos específicos, otimizando o texto para tokenização e análise em tarefas como tradução automática ou análise de dependência. As técnicas comuns na pré-tokenização incluem o tratamento de contrações (por exemplo, `can't` para `can` e `'t`), a separação de pontuação, a segmentação de palavras hifenizadas e o processamento de caracteres especiais e abreviações. Os desafios incluem ambiguidade na segmentação e considerações específicas do idioma, mas a pré-tokenização eficaz aprimora as tarefas de PLN posteriores, garantindo que o texto seja bem estruturado para análise;

2. **Segmentação e Correspondência de Tokens**: Após a pré-tokenização, o algoritmo *WordPiece* segmenta o texto em possíveis *tokens* com base em espaços em branco e pontuação, e então tenta corresponder cada segmento ao seu vocabulário. A construção do vocabulário começa com caracteres individuais como base e vai aprendendo, de forma iterativa, combinações de *tokens* frequentes, que são mescladas até atingir um tamanho de vocabulário desejado (geralmente em torno de 30.000 *tokens* para o BERT em inglês). Na prática, o algoritmo *WordPiece* aplica uma estratégia de *longest-match-first*, que tenta casar a sequência de caracteres mais longa possível à esquerda de cada segmento com um item conhecido do vocabulário. Isso ajuda a lidar com palavras complexas ou raras, dividindo-as em subpalavras mais comuns ou caracteres individuais, o que é especialmente vantajoso para línguas com rica morfologia ou para jargões técnicos [284];

3. **Tokenização em Subpalavras**: Se uma palavra completa é encontrada no vocabulário, ela é mantida como está; caso contrário, o algoritmo a decompõe em subpalavras ou caracteres individuais, o que ajuda a lidar com palavras OOV. Por exemplo, a palavra `playing` pode ser tokenizada em `play` e `##ing` onde `##` indica que essa parte continua o *token* anterior. Essa flexibilidade permite que o BERT trabalhe com uma variedade de entradas linguísticas sem precisar de um vocabulário exaustivo;

4. **Tratamento de Palavras OOV e Casos Especiais**: Quando palavras não são encontradas no vocabulário, o *WordPiece* as decompõe em unidades reconhecíveis, preservando a integridade semântica até certo ponto, mas ocasionalmente perdendo um pouco do sentido contextual. *Tokens* desconhecidos são mapeados para `[UNK]`, números são separados em dígitos individuais, e espaços em branco são preservados com *tokens* especiais. Esses processos aprimoram a capacidade do BERT de lidar com estruturas linguísticas variadas e palavras complexas, embora possam ocorrer divisões de *tokens* não intencionais, especialmente em contextos técnicos;

5. **Adição de *Tokens* Especiais**: Por fim, o BERT adiciona *tokens* específicos, como `[CLS]` no início de cada sequência e `[SEP]` entre sentenças ou no final. O *token* `[CLS]` é a primeira posição na sequência, capturando uma representação resumida de toda a frase (ou frases) depois de passar por todas as camadas da arquitetura *Transformer*, contendo o contexto de toda a frase, o que a torna muito útil para tarefas de classificação ou tarefas que exigem compreensão em nível de sentença, enquanto `[SEP]` ajuda a delimitar os limites das sentenças, conforme a implementação original do BERT.

Como ilustrado na Figura 3.1, cada *token* resultante da tokenização é convertido em uma soma de três componentes principais:

- ***token embeddings***, que codificam o significado do *token* em si;
- ***segment embeddings***, que indicam a qual sentença o *token* pertence (importante em tarefas que envolvem pares de sentença);
- ***position embeddings***, que fornecem ao modelo informações sobre a posição do *token* na sequência,

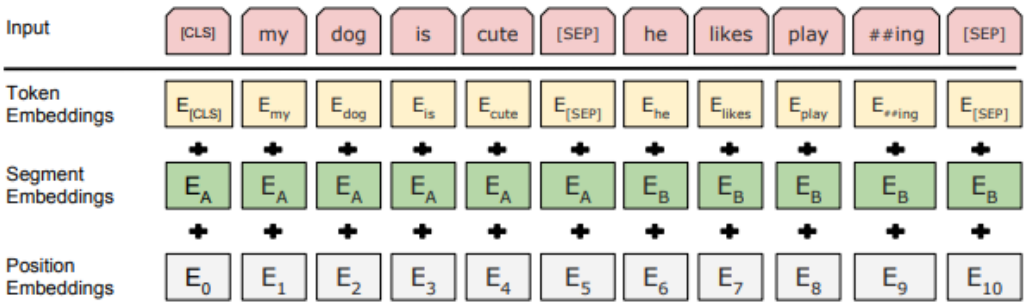

Figura 3.1 – Representação do processo de tokenização no BERT. Extraída de [1].

Por exemplo, no caso de `playing`, dividida em `play` e `##ing`, cada subpalavra recebe seu próprio *token embedding*, mas compartilha o mesmo *segment embedding* e tem um *position embedding* que reflete sua localização na frase. Esse mecanismo reforça a importância da tokenização em subpalavras, pois mesmo ao lidar com palavras raras ou fora do vocabulário, o BERT consegue manter parte do contexto ao representar cada fragmento individualmente, sem precisar de um vocabulário completamente exaustivo.

O método de tokenização *WordPiece* no BERT traz vantagens significativas para lidar com diversos tipos de linguagem. Um dos principais benefícios é sua flexibilidade: a abordagem de tokenização em subpalavras permite que o BERT processe palavras novas e variações de forma eficiente, sem precisar de um vocabulário exaustivo. Além disso, o algoritmo é projetado para ser eficiente, minimizando a complexidade computacional e

alcançando complexidade de tempo linear $O(n)$, o que é particularmente vantajoso para grandes *datasets*.

No entanto, esse método também apresenta desafios. Como ele se baseia na estratégia de *longest-match-first* (correspondência da sequência mais longa), às vezes pode levar a decisões de tokenização subótimas, especialmente ao lidar com linguagem de nicho ou pequenos erros ortográficos que não se encaixam bem no vocabulário existente. Outro problema é a possível perda de significado semântico quando palavras fora do vocabulário são divididas em subpalavras. Componentes individuais podem não carregar o mesmo peso contextual do termo original, o que leva a uma perda parcial de significado em certos contextos. Apesar dessas limitações, a tokenização *WordPiece* continua sendo uma abordagem eficaz, equilibrando flexibilidade e eficiência para dar suporte à adaptabilidade do BERT em várias tarefas de PLN.

3.1.4 Hiperparâmetros Arquitetônicos

A arquitetura do BERT é definida por vários hiperparâmetros importantes que governam sua funcionalidade e adaptabilidade a várias tarefas de processamento de linguagem natural. Esses hiperparâmetros, conforme mostrado nas Tabelas 3.1 e 3.2, incluem elementos fundamentais, como o tamanho do vocabulário, o número e as dimensões das camadas ocultas e os parâmetros relacionados aos mecanismos de atenção e às incorporações posicionais.

Tabela 3.1 – Hiperparâmetros principais do modelo BERT (parte 1).

Hiperparâmetro	**Descrição**
Tamanho do vocabulário (*vocabulary size*)	Número de *tokens* únicos, incluindo palavras, subpalavras e especiais. Define o tamanho das embeddings.
Dimensão das camadas ocultas (*hidden size*)	Dimensionalidade das embeddings e dos estados ocultos. Tamanhos maiores aumentam a capacidade, mas exigem mais recursos.
Número de camadas ocultas (*number of hidden layers*)	Número de blocos *Transformer* no modelo. Mais camadas capturam padrões complexos, mas aumentam o custo.
Número de cabeças de atenção (*number of attention heads*)	Número de cabeças de atenção em cada camada do modelo.

Tabela 3.2 – Hiperparâmetros principais do modelo BERT (parte 2).

Hiperparâmetro	**Descrição**
Dimensão da camada intermediária (*intermediate size*)	Tamanho da camada feed-forward dentro de cada bloco *Transformer*. Tipicamente maior que a dimensão oculta.
Probabilidade de *Dropout* nas camadas ocultas (*hidden dropout probability*)	Probabilidade de desligamento de neurônios nas camadas ocultas durante o treinamento. Ajuda a prevenir overfitting.
Máximo de *embeddings* posicionais (*max position embeddings*)	Comprimento máximo da sequência de entrada que o modelo pode processar.

3.1.5 RoBERTa

O RoBERTa [285], abreviação de *Robustly optimized BERT approach* (Abordagem BERT com Otimização Robusta), representa um avanço significativo em relação ao modelo BERT original. Uma das principais diferenças entre o RoBERTa e o BERT está na otimização do processo de pré-treinamento. O RoBERTa foi desenvolvido por meio de uma reavaliação da metodologia de pré-treinamento do BERT, o que levou a várias melhorias importantes.

Em primeiro lugar, o RoBERTa aumenta a quantidade de dados de treinamento e a duração do treinamento. Enquanto o BERT foi pré-treinado em um *dataset* de 16 GB, o RoBERTa usa um *dataset* muito maior, de 160 GB, que inclui dados de fontes como *Common Crawl News*, *WebText* e outras. Esse extenso *dataset* permite que o RoBERTa aprenda uma representação mais abrangente da linguagem. Além disso, o RoBERTa estende o tempo de treinamento e aumenta o tamanho do lote, o que ajuda a melhorar a convergência e a compreensão mais robusta da linguagem.

Outra diferença significativa é a remoção do objetivo de NSP usado no BERT. Essa tarefa no BERT foi projetada para ajudar o modelo a entender a relação entre as frases, mas foi considerada menos eficaz. O RoBERTa elimina essa tarefa e, em vez disso, concentra-se apenas no objetivo do MLM, que se mostrou mais benéfico para o pré-treinamento.

O RoBERTa também emprega *dynamic masking* (mascaramento dinâmico), em que o padrão de mascaramento muda durante o treinamento, ao contrário do BERT, que usa um padrão de mascaramento estático. Essa abordagem dinâmica garante que o modelo não fique muito acostumado a um conjunto específico de *tokens* mascarados, o que leva a uma melhor generalização.

Em termos de desempenho, o RoBERTa supera consistentemente o BERT em vários *benchmarks*. Por exemplo, no *benchmark* GLUE (*General Language Understanding*

Evaluation) [286][1], o RoBERTa obtém resultados de última geração, superando o desempenho do BERT. Essa melhoria também é evidente em outras tarefas, como o *Stanford Question Answering Dataset* (SQuAD) [287], [288][2], como é possível ver na Tabela 3.3.

Tabela 3.3 – Comparação entre BERT e RoBERTa. Dados levantados a partir dos trabalhos de [1] e [285].

Tarefa	**Métrica**	**BERT**	**RoBERTa**
GLUE			
MNLI-m	Acurácia	86,7	90,8
MNLI-mm	Acurácia	85,9	90,2
QQP	F1/Acurácia	72,1/89,3	74,3/90,2
QNLI	Acurácia	92,7	95,4
SST-2	Acurácia	94,9	96,7
CoLA	Correlação de Matthews	60,5	67,8
STS-B	Correlação de Pearson/Spearman	87,6/86,5	92,2/91,9
MRPC	F1/Acurácia	89,3/85,4	92,3/89,8
RTE	Acurácia	70,1	88,2
WNLI	Acurácia	65,1	89
SQuAD			
SQuAD v1.1	Exact Match/F1	84,1/90,9	88,9/94,6
SQuAD v2.0	Exact Match/F1	79,0/81,8	86,5/89,4

3.1.5.1 XLM-RoBERTa

O XLM-RoBERTa [17], uma variante do RoBERTa, é uma abordagem projetada para lidar com vários idiomas de forma eficaz. Ao contrário do BERT, que se concentra principalmente no inglês, o XLM-RoBERTa é treinado em um *dataset* multilíngue diversificado, o que o torna altamente versátil para várias tarefas de PLN em diferentes idiomas.

Uma das principais diferenças entre a XLM-RoBERTa e o BERT são os dados e a metodologia de treinamento. O XLM-RoBERTa aproveita um *corpus* maior e mais diversificado (pré-treinado em 2,5 TB de dados filtrados do CommonCrawl[3]), que inclui dados de 100 idiomas, aumentando sua capacidade de generalização.

Uma das principais diferenças entre o XLM-R e o BERT é a escala e a diversidade dos dados de treinamento. Enquanto o BERT foi treinado em textos em inglês, o *corpus* de treinamento do XLM-R inclui uma grande variedade de idiomas, o que aumenta significativamente sua capacidade de apresentar bom desempenho em *benchmarks* multilíngues. Por exemplo, o XLM-R supera o BERT multilíngue (mBERT) por uma margem substancial, alcançando uma melhoria média de precisão de +14,6% no *benchmark* XNLI [289] e um aumento médio de +13% na pontuação F1 no *benchmark* MLQA [290].

[1] Tabela de classificação dos modelos disponível em https://gluebenchmark.com/leaderboard.
[2] Tabela de classificação dos modelos disponível em https://rajpurkar.github.io/SQuAD-explorer/.
[3] https://commoncrawl.org/

Essas melhorias são particularmente acentuadas em idiomas com poucos recursos, em que o XLM-R mostra um ganho de precisão de 15,7% para o suaíli e um ganho de 11,4% para o urdu no *benchmark* XNLI.

3.1.6 DistilBERT

A destilação é uma técnica usada no aprendizado de máquina para transferir conhecimento de um modelo grande e complexo (geralmente chamado de "professor") para um modelo menor e mais eficiente (o "aluno"). Esse processo tem como objetivo manter a maior parte dos benefícios de desempenho do modelo maior e, ao mesmo tempo, reduzir significativamente seu tamanho e seus requisitos computacionais. O conceito de destilação tem sido particularmente útil no campo do PLN, em que grandes modelos pré-treinados, como o BERT, estabeleceram novos padrões de referência, mas muitas vezes consomem muitos recursos para serem implementados na prática.

O DistilBERT [145] é uma versão destilada do BERT, projetada para ser menor, mais rápida, mais barata e mais leve, mantendo a maioria dos recursos de desempenho do BERT. A principal motivação por trás do DistilBERT é tornar os poderosos recursos de compreensão de linguagem do BERT acessíveis em ambientes com recursos limitados, como dispositivos de ponta ou situações com orçamentos computacionais limitados. O DistilBERT consegue isso aproveitando a destilação de conhecimento durante a fase de pré-treinamento, reduzindo o tamanho do modelo do BERT em 40%, mantendo 97% de seus recursos de compreensão de linguagem e sendo 60% mais rápido.

Embora o BERT seja um modelo grande e poderoso com um número significativo de parâmetros, o DistilBERT foi projetado para ser uma versão mais eficiente. As principais diferenças entre o BERT e o DistilBERT incluem:

- **Tamanho do modelo**: O DistilBERT é 40% menor que o BERT, o que o torna mais adequado para implantação em ambientes com recursos limitados;
- **Velocidade**: O DistilBERT é 60% mais rápido que o BERT, o que é crucial para aplicativos que exigem processamento em tempo real;
- **Eficiência de treinamento**: O pré-treinamento do DistilBERT é mais viável computacionalmente, o que pode ser uma vantagem significativa em termos de recursos computacionais e tempo.

O DistilBERT é melhor do que outros modelos de linguagem pré-treinados de várias maneiras:

- **Eficiência**: O tamanho menor e a velocidade de inferência mais rápida do DistilBERT o tornam mais prático para aplicativos do mundo real, especialmente aqueles que

exigem computações no dispositivo [291]. Essa eficiência não é obtida à custa do desempenho, pois o DistilBERT mantém 97% dos recursos de compreensão de linguagem do BERT;

- **Generalização**: O DistilBERT pode ser ajustado para uma ampla gama de tarefas, de forma semelhante às suas contrapartes maiores. Essa versatilidade é demonstrada em vários estudos em que o DistilBERT foi aplicado com sucesso a tarefas como reconhecimento de entidades nomeadas em dados de saúde [292] e análise de sentimentos financeiros [293];

- **Viés reduzido**: Curiosamente, alguns estudos descobriram que modelos destilados como o DistilBERT apresentam menos viés estereotipado de gênero em comparação com seus modelos professores. Isso pode ser uma consideração importante para aplicativos em que a equidade e a imparcialidade são fundamentais [294].

3.1.7 DeBERTa

O DeBERTa [295], abreviação de *Decoding-enhanced BERT with Disentangled Attention* (BERT aprimorado por decodificação com atenção desagregada), representa um avanço significativo no campo dos modelos de linguagem pré-treinados. Ele apresenta duas novas técnicas: o mecanismo de atenção desagregada e o decodificador de máscara aprimorado, que, juntos, melhoram o desempenho do modelo em várias tarefas de PLN.

O DeBERTa se diferencia do BERT por codificar cada palavra na camada de entrada de forma diferente. Em contraste com o método do BERT, que usa um único vetor que combina a incorporação de palavras e posições, o DeBERTa, através do mecanismo de atenção desagregada, usa dois vetores distintos para capturar o conteúdo e a posição de cada palavra. Os pesos de atenção entre as palavras são calculados usando matrizes separadas que consideram o conteúdo e as posições relativas. Essa abordagem inovadora permite que o modelo se concentre na relação entre o conteúdo e a posição do texto, levando a um melhor desempenho em tarefas como análise de sentimentos [296].

O DeBERTa usa uma técnica distinta conhecida como *enhanced masked decoder* (decodificador mascarado aprimorado), que o diferencia do uso do BERT de codificações posicionais absolutas na camada de entrada. Em vez de incorporar posições absolutas na camada de entrada, o DeBERTa integra posições absolutas após cada camada da arquitetura *Transformer*, logo antes da camada Softmax, que prevê *tokens* mascarados. Essa estratégia permite que o modelo capture posições relativas em todas as camadas da arquitetura *Transformer* e, ao mesmo tempo, use posições absolutas como informações suplementares durante a decodificação. Ao adotar essa abordagem, o DeBERTa pode introduzir informações adicionais valiosas durante o pré-treinamento, aprimorando a utilização de dados posicionais pelo modelo [297].

O DeBERTa no seu pré-treinamento utilizou dados da *Wikipedia* (*dump* de 12GB da versão inglesa), o *BookCorpus* (6GB), o *OPENWEBTEXT* (38GB de conteúdo público do *Reddit*) e o *STORIES* (31GB, sendo um subconjunto do *CommonCrawl*). Após a deduplicação, o tamanho final dos dados totalizou 78GB.

O DeBERTa passou por várias iterações, com a DeBERTaV2 e DeBERTaV3 trazendo melhorias significativas. A DeBERTaV2 introduziu otimizações no processo de treinamento e na arquitetura do modelo, aumentando ainda mais seu desempenho. O DeBERTaV3, por sua vez, substituiu a tarefa de mask language modeling (MLM) pela *replaced token detection* (RTD), uma tarefa de pré-treinamento mais eficiente no uso de amostras. Além disso, o DeBERTaV3 introduziu o compartilhamento de *gradient-disentangled embedding sharing* (compartilhamento de *embeddings* com gradientes desagregados) para evitar a dinâmica de "cabo de guerra" observada em modelos anteriores, onde as perdas de treinamento do discriminador e do gerador puxavam os *embeddings* de *tokens* em direções opostas. Essas melhorias levaram a ganhos expressivos na eficiência do treinamento e no desempenho do modelo, estabelecendo novos patamares de referência em várias tarefas de PLN [298], como demonstrado pelos avanços em *benchmarks* como MNLI, SQuAD v2.0 e RACE.

3.1.8 BERTimbau

O BERTimbau [18], um modelo BERT especializado para o português brasileiro, utiliza o BrWaC como seu *corpus* de pré-treinamento. Ao utilizar o BrWaC, o BERTimbau captura as nuances e as complexidades do português brasileiro, tornando-o particularmente eficaz para várias tarefas de PLN.

Uma das melhorias significativas que o BERTimbau oferece em relação a outros modelos linguísticos pré-treinados em português, como o mBERT, é sua capacidade de entender e processar melhor as características linguísticas específicas do português brasileiro. Essa especialização é crucial porque o mBERT, embora versátil, é treinado em uma grande variedade de idiomas, o que pode diminuir sua eficácia em um único idioma. Em contrapartida, o treinamento focado do BERTimbau no BrWaC permite que ele obtenha maior precisão e desempenho em tarefas específicas do português.

A eficácia do BERTimbau é evidente em seu desempenho em tarefas de referência, como ASSIN2 e MiniHAREM [299], quando comparado a modelos baseados em redes neurais recorrentes e combinações (*ensembles*) de modelos baseados na arquitetura *Transformer*. O BERTimbau demonstrou um desempenho superior no ASSIN2 e, da mesma forma, se sobressaiu na tarefa MiniHAREM, que se concentra no reconhecimento de entidades nomeadas em português, melhorando significativamente os resultados do estado da arte.

3.1.9 SEC-BERT e BusinessBERT

Embora vários modelos de linguagem sejam comumente citados na pesquisa de NLP financeira, eles representam uma classe heterogênea de modelos com objetivos, *corpora* de treinamento e domínios variados (por exemplo, sentimento em notícias ou *question answering* em relatórios). Essa inconsistência dificulta seu uso como uma linha de base unificada e confiável. Por outro lado, o SEC-BERT e o BusinessBERT oferecem escopos bem definidos e complementares — visando a registros regulatórios e textos amplos de negócios/setor, respectivamente — proporcionando mais clareza e relevância ao comparar um modelo proposto. Além disso, muitas variantes encontradas na literatura, como o FinBERT, enfatizam tarefas restritas, como a análise de sentimentos, que já são abordadas ou incluídas nos objetivos mais amplos do SEC-BERT e do BusinessBERT, reforçando a decisão de priorizar esses dois modelos como linhas de base comparativas.

O SEC-BERT [10] é uma variante de domínio específico do BERT desenvolvida para a tarefa de marcação XBRL em relatórios financeiros (trata-se de uma marcação de dados financeiros com etiquetas padronizadas, permitindo automação, interoperabilidade e análise eficiente de relatórios contábeis). Como as empresas de capital aberto devem usar *tags* padronizadas baseadas em XML para envios regulatórios, a marcação eficaz é crucial. A tokenização tradicional do BERT tem dificuldades com dados numéricos, por isso o SEC-BERT apresenta três versões especializadas: SEC-BERT-BASE, treinado com documentos financeiros e dotado da mesma arquitetura que o BERT, SEC-BERT-NUM, que substitui todos os números por um *token* `[NUM]`, e SEC-BERT-SHAPE, que os substitui por *placeholders* baseados em formas (por exemplo, `53.2` → `[XX.X]`). Treinado em 200.000 registros da SEC, o SEC-BERT entende melhor a terminologia financeira e lida com a complexidade de 139 tipos de entidades no dataset FINER-139. O SEC-BERT-SHAPE superou o desempenho do BERT de uso geral e de modelos financeiros como o FinBERT, alcançando 82,1% de F1 micro.

O BusinessBERT [300], por sua vez, é um modelo de linguagem adaptado para tarefas de PLN relacionadas a negócios. Diferentemente dos modelos gerais, ele incorpora o conhecimento específico do setor usando duas inovações: treinamento em 2,23 bilhões de *tokens* de diversas fontes de negócios e adição da classificação do setor industrial como um objetivo de pré-treinamento. Essa abordagem permite que o BusinessBERT capture a terminologia comercial diferenciada e melhore o desempenho em tarefas como classificação de texto, reconhecimento de entidades nomeadas e *question answering*. Ele supera consistentemente o BERT-Base e o FinBERT, com ganhos de desempenho que variam de 1% a 9,69%, usando significativamente menos dados de pré-treinamento (28% a 54% menos *tokens*). Os pesquisadores também demonstraram que sua abordagem sensível ao setor melhora outros modelos, incluindo o RoBERTa e o LLaMA 2.

3.1.10 Limitações e Desafios Técnicos

O BERT e suas variantes revolucionaram o campo do PLN ao obter resultados de última geração em várias tarefas. No entanto, esses modelos apresentam várias limitações e desafios técnicos que precisam ser resolvidos para que haja mais avanços.

Um dos principais desafios do BERT e de modelos semelhantes é a superparametrização, que leva a altos custos computacionais e ao uso da memória. O grande número de parâmetros torna esses modelos intensivos em recursos, exigindo recursos significativos de GPU/TPU para treinamento e inferência. Esse problema é particularmente destacado no desenvolvimento do ALBERT, que introduz técnicas de redução de parâmetros para diminuir o consumo de memória e aumentar a velocidade de treinamento, tornando o modelo mais escalável [301].

Embora o BERT seja pré-treinado em grandes *corpora* diversos, como o *BookCorpus* e a Wikipedia, ele frequentemente carece do conhecimento específico de tarefas e domínios necessários para um desempenho ideal em aplicações especializadas. Essa limitação exige ajustes adicionais de *fine-tuning* ou a incorporação de sentenças auxiliares e *corpora* relacionados ao domínio para melhorar o desempenho, como demonstrado, por exemplo, pelo modelo BERT4TC [302]. A necessidade de estratégias extensivas de *fine-tuning*, incluindo ajustes na taxa de aprendizado, comprimento da sequência e seleção do vetor de estados ocultos, complica ainda mais a implantação de modelos BERT em tarefas específicas.

Os modelos BERT são predominantemente treinados em idiomas com grande disponibilidade de recursos, o que apresenta desafios significativos para aplicações multilíngues e em idiomas com poucos recursos. O desempenho do mBERT costuma ser inferior ao de modelos específicos para cada idioma, devido à limitada e pouco organizada disponibilidade de dados para línguas com menos recursos. Por exemplo, o Bangla-BERT foi desenvolvido para enfrentar esses desafios, sendo pré-treinado em um extenso *dataset* em bengali e alcançando resultados superiores ao mBERT [303].

3.2 GPT

A progressão dos modelos *Generative Pre-trained Transformers* (GPT), do GPT ao GPT-4, representa uma evolução significativa no campo do PLN. Cada iteração introduziu aprimoramentos na arquitetura, na diversidade de dados de treinamento e nas abordagens de *fine-tuning*, levando a um melhor desempenho na compreensão da linguagem e nos recursos de geração. Esta análise abrangente se aprofundará nas distinções entre esses modelos, suas aplicações, limitações e implicações éticas, principalmente em contextos especializados, como finanças e configurações linguísticas diferenciadas.

3.2.1 Arquitetura

A arquitetura dos modelos GPT, apresentada originalmente por [304], é fundamentada em um componente *Transformer* unicamente decodificador, que opera de forma autorregressiva, prevendo cada *token* subsequente com base nos *tokens* anteriores. O GPT-1, com 117 milhões de parâmetros, evidenciou a viabilidade de grandes modelos para PLN, possibilitando a geração de texto e *question answering*.

O GPT-2 [4], lançado em 2019 com 1,5 bilhão de parâmetros, consolidou o potencial dos modelos de larga escala, gerando textos mais coerentes e contextualmente relevantes. Esse aumento no tamanho do modelo permitiu maior desempenho em tarefas como geração de texto e inferência, suportando sequências de até 1024 *tokens* e reforçando a capacidade de entender o contexto.

Em 2020, o lançamento do GPT-3 [5] com 175 bilhões de parâmetros marcou um avanço significativo no processamento de linguagem natural ao introduzir os recursos de aprendizagem *zero-shot* e *few-shot*, permitindo que o modelo generalizasse tarefas a partir de apenas alguns exemplos. Essa evolução permitiu que o modelo fosse usado sem a necessidade de *fine-tuning*, expandindo significativamente suas aplicações para várias tarefas, como tradução, resposta a perguntas complexas e outras tarefas de compreensão. A arquitetura do GPT-3 permite que ele tenha um bom desempenho em muitos *benchmarks* de PLN puramente por meio da interação de texto, sem nenhuma atualização de gradiente ou *fine-tuning* específico da tarefa, demonstrando um forte desempenho em tarefas que exigem raciocínio imediato ou adaptação de domínio. Além disso, o GPT-3 demonstrou uma capacidade surpreendente de indução de padrões abstratos, superando até mesmo as capacidades humanas em algumas tarefas de raciocínio analógico [305]. A capacidade do modelo de gerar texto semelhante ao humano também foi observada, com avaliadores humanos achando difícil distinguir entre artigos escritos pelo GPT-3 e aqueles escritos por humanos.

A partir do GPT-3, desenvolveu-se uma nova iteração: o GPT-3.5. Os modelos GPT-3.5, como text-davinci-002, text-davinci-003 e gpt-3.5-turbo, apresentaram desempenho superior em tarefas de compreensão de linguagem natural *(natural language understanding* — NLU) em comparação com seus equivalentes do GPT-3, como davinci e text-davinci-001. A introdução da aprendizagem por reforço com feedback humano (*reinforcement learning from human feedback* — RLHF) no GPT-3.5 também aprimorou a capacidade dos modelos de gerar respostas semelhantes às humanas. No entanto, essas melhorias nem sempre são lineares ou consistentes em todas as tarefas [306].

O GPT-4 [6], lançado em 2023, representa um grande avanço nas capacidades dos modelos de linguagem em larga escala, ao incorporar um número estimado em trilhões de parâmetros e ao ser treinado com dados multimodais, que incluem tanto texto quanto

imagens. Esse aprimoramento permitiu que o GPT-4 gerasse respostas mais precisas e contextualmente relevantes em uma variedade de tarefas complexas. O suporte do modelo para multimodalidade permite interpretar e gerar conteúdos que abrangem diferentes formatos, como texto e imagens, ampliando sua utilidade em áreas como a saúde, onde pode auxiliar no diagnóstico clínico ao analisar imagens médicas e dados textuais de pacientes [307], [308]. Além disso, o desempenho aprimorado do GPT-4 em tarefas de alinhamento ético e controle de respostas é evidente quando comparado a seus predecessores, demonstrando maior precisão em exames especializados, como os exames de conselhos médicos. A integração de modelos de linguagem com *visual encoders*, como visto em projetos como o MiniGPT-4, também exemplifica a capacidade do GPT-4 de realizar tarefas como gerar descrições detalhadas de imagens e criar sites a partir de esboços desenhados à mão [309]. Esses avanços destacam o potencial do GPT-4 como uma versão inicial de inteligência artificial geral, capaz de resolver tarefas novas e complexas em diversas áreas sem a necessidade de instruções específicas [310].

A Tabela 3.4 traz uma comparação dos *corpora* utilizados no treinamento de cada versão do GPT.

3.2.2 Tokenização

Os modelos GPT utilizam a codificação por pares de bytes (*byte pair encoding* — BPE) para tokenização, o que equilibra de forma eficaz o equilíbrio entre o tamanho do vocabulário e a capacidade de lidar com palavras raras. A BPE, originalmente concebida como um método de compactação, foi adaptada para o processamento de linguagem natural para dividir o texto em unidades de subpalavras, permitindo que os modelos gerem e compreendam uma variedade maior de vocabulário. Esse método funciona mesclando iterativamente os pares mais frequentes de bytes ou caracteres em um texto, o que permite lidar com eficiência com palavras comuns e raras. Os avanços na implementação da BPE melhoraram sua complexidade de tempo de execução, tornando-o mais eficiente para tarefas de processamento de texto em grande escala [311]. Essa combinação de eficiência e eficácia torna a BPE uma ferramenta valiosa no processo de tokenização para modelos de GPT.

3.2.3 Hiperparâmetros Arquitetônicos

Os parâmetros dos modelos baseados em GPT regem vários aspectos do comportamento do modelo, inclusive como ele processa a entrada, gera respostas e controla o nível de criatividade ou aleatoriedade em sua saída. Esses parâmetros permitem que os usuários adaptem o desempenho do modelo de acordo com necessidades específicas, como limitar o comprimento da resposta, aumentar a coerência ou orientar o modelo para resultados mais focados ou diversificados. Conforme mostrado na Tabela 3.5, cada

Tabela 3.4 – Comparação dos Corpora de Treinamento dos Modelos GPT. Adaptado de [304], [4] e [5].

Modelo	**Corpora de Treinamento**	**Tamanho do Dataset**	**Observações**
GPT-1 (2018)	BooksCorpus	7.000 livros (∼5GB)	Primeiro modelo da série
GPT-2 (2019)	WebText	∼8 milhões de documentos (∼40GB)	*Dataset* criado através de scraping de páginas web (links com pelo menos 3 upvotes no Reddit)
GPT-3 (2020)	CommonCrawl filtrado (60%), WebText2 (22%), Books1 (8%), Books2 (4%), Wikipedia (3%)	Common Crawl: 410B *tokens*; WebText2: 19B *tokens*; Books1: 8B *tokens*; Books2: 4B *tokens*; Wikipedia: 3B *tokens*. Total: 499B *tokens*. Tamanho final: ∼570GB	Filtragem extensiva do CommonCrawl (45TB → 60GB). Deduplicação no nível do documento, tanto dentro de *datasets* quanto entre *datasets* diferentes. Vocabulário de 50.257 *tokens* (assim como no GPT-2). Predominantemente em inglês (93% das palavras)
GPT-4 (2023)	Não divulgado publicamente	Não divulgado publicamente	A OpenAI não revelou detalhes sobre o *dataset* de treinamento. Especula-se que seja significativamente maior que o GPT-3. Inclui dados multimodais (texto e imagens)

parâmetro tem uma função distinta na influência da qualidade geral e da relevância do conteúdo gerado.

Tabela 3.5 – Hiperparâmetros que influenciam o comportamento e a saída dos modelos GPT.

Parâmetro	**Descrição**
`model` (*modelo*)	Especifica o modelo GPT a ser utilizado, como `gpt-3.5-turbo` ou `gpt-4`. Esse parâmetro define qual versão do modelo será chamada.
`messages` (*mensagens*)	A lista de mensagens que forma a conversação. Cada mensagem inclui um `role` (papel) e o `content` (conteúdo). Pode incluir um histórico de interações passadas.
`temperature` (*temperatura*)	Controla a aleatoriedade das respostas geradas. Valores mais baixos (ex: 0.0) resultam em respostas mais determinísticas; valores mais altos (ex: 1.0), mais criativas.
`max_tokens` (*máximo de tokens*)	Define o número máximo de *tokens* para a resposta gerada. Limita a quantidade de texto produzido.
`top_p` (*top-p*)	Parâmetro de amostragem por núcleo (*nucleus sampling*). Valores baixos tornam a geração mais focada; valores altos aumentam a diversidade.
`frequency_penalty` (*penalidade de frequência*)	Penaliza tokens que aparecem frequentemente, ajudando a reduzir repetições no texto gerado.
`presence_penalty` (*penalidade de presença*)	Penaliza a introdução de novos tópicos ainda não abordados, incentivando respostas mais focadas.
`stop` (*sequências de parada*)	Define uma ou mais sequências que interrompem a geração de texto. Útil para controlar quando o modelo deve parar de responder.

3.2.4 Limitações e Desafios Técnicos

Os modelos GPT, sobretudo os modelos 3, 3.5 e 4 possuem limitações e desafios já bastante abordados na literatura.

Os modelos GPT-3, como o davinci e o text-davinci-001, são conhecidos por seu tamanho enorme, o que resulta em custos computacionais e de armazenamento extremamente altos. Isso os torna menos acessíveis para muitos usuários e limita sua usabilidade àqueles com recursos computacionais significativos [312].

O GPT-3 também enfrenta desafios relacionados à complexidade do treinamento e aos vieses inerentes. O processo de treinamento do modelo consome muitos recursos e, apesar do treinamento extensivo, ele ainda pode produzir resultados tendenciosos ou incorretos, conhecidos como alucinações [313].

Quanto ao GPT-3.5, foi observado que o comportamento e o desempenho do modelo mudam ao longo do tempo com as atualizações. Por exemplo, o desempenho do GPT-3.5 em tarefas como problemas matemáticos e geração de código variou significativamente entre março de 2023 e junho de 2023, demonstrando a natureza dinâmica desses modelos

e a necessidade de monitoramento contínuo [314].

Apesar dos avanços, os modelos GPT-3.5 ainda enfrentam desafios em termos de robustez e estabilidade. Estudos demonstraram que o GPT-3.5 pode apresentar quedas significativas de desempenho em tarefas como inferência de linguagem natural e análise de sentimentos quando submetida a várias transformações de texto. Isso indica que, embora o GPT-3.5 supere os modelos de *fine-tuning* em algumas áreas, ele ainda tem dificuldades com a robustez e a generalização [315].

O GPT-4, apesar de seus avanços, enfrenta preocupações éticas e de privacidade significativas. A capacidade do modelo de gerar textos altamente realistas levanta questões relacionadas à desinformação, violações de privacidade e ao uso ético de conteúdo gerado por IA. Além disso, o tamanho maior do modelo GPT-4, que ultrapassa um trilhão de parâmetros, exige ainda mais potência computacional e dados para treinamento e implantação. [316].

Todos os modelos GPT, incluindo GPT-3, GPT-3.5 e GPT-4, sofrem com a falta de interpretabilidade. Compreender como esses modelos chegam a determinados resultados é um desafio, o que complica seu uso em aplicações críticas, onde a transparência é essencial [317].

A implantação de grandes modelos GPT é complexa e geralmente requer ciclos de desenvolvimento fechados, o que restringe sua acessibilidade e levanta preocupações sobre o desenvolvimento e uso responsáveis [318].

3.3 Resumo do Estado da Arte e Abordagem do Modelo Proposto

As Tabelas 3.7 e 3.7 apresentam um resumo dos modelos de estado da arte apresentados no Capítulo 3:

Tabela 3.6 – Resumo dos Modelos de Linguagem – Parte 1

Modelo	**Ano**	**Características**	**Corpus**	**Limitações**
BERT	2018	Bidirecional; previsão de palavras mascaradas; tokenização WordPiece.	BooksCorpus, Wikipédia	Unidirecionalidade limitada e uso intensivo de recursos.
RoBERTa	2019	Otimização do BERT com mais dados; sem próxima frase; mascaramento dinâmico.	Common Crawl News, WebText	Maior demanda computacional.
XLM-RoBERTa	2020	Variante multilíngue do RoBERTa.	CommonCrawl	Alto custo; menos especialização por idioma.
DistilBERT	2020	Compacto via destilação; 40% menor e 60% mais rápido.	Semelhante ao BERT	Menor precisão.
DeBERTa	2021	Atenção desemaranhada; decodificação mascarada aprimorada.	BookCorpus, Wikipedia, OpenWebText, Stories	Treinamento e uso complexos.
BERTimbau	2023	BERT para o português brasileiro.	BrWaC	Pouca eficácia fora do domínio do português.
SEC-BERT	2022	Especializado em relatórios financeiros XBRL.	200 mil arquivos da SEC	Restrito ao domínio financeiro.

Tabela 3.7 – Resumo dos Modelos de Linguagem – Parte 2

Modelo	**Ano**	**Características**	**Corpus**	**Limitações**
BusinessBERT	2024	Foco empresarial; pré-treinamento com setores industriais.	2,23B tokens de negócios	Baixa eficácia fora de contexto empresarial.
GPT-1	2018	Autoregressivo com previsão sequencial.	BooksCorpus	Corpus pequeno e desempenho limitado.
GPT-2	2019	Expansão do GPT-1; 1,5B parâmetros; 1024 tokens.	WebText	Pode gerar incoerências.
GPT-3	2020	175B parâmetros; zero-shot e few-shot.	CommonCrawl e outros	Alto custo e tendência à alucinação.
GPT-3.5	2022	Refinamento com RLHF para respostas mais naturais.	CommonCrawl e outros	Resultados instáveis entre versões.
GPT-4	2023	Multimodal e mais preciso.	Não especificado	Alto custo, privacidade e desinformação.

O método proposto neste trabalho se diferencia dos modelos financeiros já publicados principalmente por sua abordagem de pré-treinamento em domínios mistos, especificamente voltada para o português e adaptada ao contexto financeiro. Enquanto muitos modelos existentes, como o SEC-BERT e o BusinessBERT, são projetados para operar em inglês e com foco em dados puramente relacionados ao domínio financeiro, o modelo desenvolvido neste estudo incorpora dados de diversos setores relacionados, incluindo política, gestão de negócios e contabilidade. Essa estratégia permite que o modelo absorva um espectro mais amplo de conhecimentos relevantes ao contexto financeiro, o que é particularmente vantajoso para contextos com menos dados anotados, como o português. Além disso, ao evitar informações de domínios muito gerais que poderiam causar transferência negativa, o modelo busca manter seu foco nos jargões e nuances do setor financeiro em português, oferecendo uma solução mais precisa e eficiente para as necessidades de instituições financeiras de países lusófonos.

3.4 Considerações Finais

Neste capítulo, foram revisados diversos trabalhos na literatura sobre os modelos de linguagem mais proximamente relacionados a esta tese, com foco especial em suas características e limitações. Uma tabela comparativa foi apresentada, delineando as principais características dos modelos analisados em relação ao trabalho atual. Além

disso, foram discutidas as conclusões e limitações apontadas pelos autores, que oferecem uma visão crítica sobre os avanços e desafios enfrentados na área.

Essas análises não apenas evidenciam a relevância dos modelos discutidos, mas também fornecem um importante insumo para a tese, permitindo uma compreensão mais profunda do contexto em que o trabalho se insere.

4 Proposta de Um Modelo de Linguagem para Processamento de Linguagem Financeira em Português

4.1 Formulação do Problema

Neste estudo, é abordado um conjunto de problemas de classificação de texto, incluindo classificação de discurso de ódio, detecção de notícias falsas, análise de sentimentos e classificação de risco regulatório. Cada uma dessas tarefas pode ser formalmente definida como um problema de aprendizado supervisionado, em que um modelo aprende a atribuir um rótulo y a um determinado texto de entrada x.

4.1.1 Definição do Problema

Seja $\mathcal{X}$ o espaço de textos de entrada e $\mathcal{Y}$ o conjunto de rótulos possíveis. O objetivo é aprender uma função de classificação:

$$f : \mathcal{X} \to \mathcal{Y}, \tag{4.1}$$

onde f é parametrizada por um modelo baseado em Transformer treinado para minimizar uma função de perda baseada em dados rotulados de treinamento $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, contendo N exemplos.

Cada tarefa de classificação é definida da seguinte forma:

- Classificação de Discurso de Ódio: $\mathcal{Y} = \{\text{discurso de ódio}, \text{não discurso de ódio}\}$
- Detecção de Notícias Falsas: $\mathcal{Y} = \{\text{falsa}, \text{legítima}\}$
- Análise de Sentimento: $\mathcal{Y} = \{\text{positivo}, \text{negativo}\}$
- Classificação de Risco Regulatório: $\mathcal{Y} = \{\text{relevante}, \text{não relevante}\}$

4.1.2 Representação da Entrada

Cada texto de entrada x_i é transformado em uma sequência de tokens:

$$x_i = (w_1, w_2, ..., w_T), \tag{4.2}$$

onde T é o número de tokens no texto, e cada token w_t é mapeado para uma representação vetorial densa usando um modelo *Transformer* pré-treinado:

$$h_t = \text{Embedding}(w_t). \tag{4.3}$$

A representação completa da sequência é obtida por meio de um codificador baseado em *Transformer*, que gera uma representação vetorial contextualizada para cada *token*:

$$H = \text{Transformer}(x_i) \in \mathbb{R}^{T \times d}, \tag{4.4}$$

onde d é a dimensão do modelo.

Para obter uma única representação do documento, aplicamos uma função de *pooling* (por exemplo, a representação do token `[CLS]` ou a média dos *tokens*):

$$z = \text{Pooling}(H) \in \mathbb{R}^d. \tag{4.5}$$

4.1.3 Função de Classificação

A classificação final é realizada usando uma camada totalmente conectada seguida de uma ativação softmax:

$$\hat{y} = \text{softmax}(Wz + b), \tag{4.6}$$

onde $W \in \mathbb{R}^{|\mathcal{Y}| \times d}$ e $b \in \mathbb{R}^{|\mathcal{Y}|}$ são parâmetros treináveis.

4.1.4 Objetivo de Otimização

O treinamento é conduzido minimizando a perda de entropia cruzada categórica *(categorical cross-entropy loss)*:

$$\mathcal{L} = -\sum_{i=1}^{N} \sum_{c=1}^{|\mathcal{Y}|} y_{i,c} \log \hat{y}_{i,c}, \tag{4.7}$$

onde $y_{i,c}$ é um rótulo real codificado em *one-hot*, e $\hat{y}_{i,c}$ é a probabilidade predita para a classe c.

Para melhorar a generalização e a estabilidade, foram incorporadas técnicas de regularização, incluindo *Mixout*, média estocástica dos pesos e decaimento da taxa de aprendizado por camada, conforme descrito na Seção 4.2.6.

4.2 Visão Geral do Fluxo de Trabalho da Pesquisa

Este estudo apresenta uma abordagem completa para avaliar vários modelos de linguagem para tarefas específicas de processamento de linguagem natural. O fluxo de trabalho da pesquisa, conforme ilustrado na Figura 4.1, envolve seis tarefas principais: seleção do modelo e configuração, definição do *corpus*, definição da *baseline*, definição dos *datasets*, pré-treinamento, *fine-tuning* de modelos e avaliação.

Na fase de seleção do modelo, apresentamos duas versões do modelo proposto, uma nova abordagem chamada DeB3RTa (B3 é uma referência à “Bolsa, Brasil, Balcão”, a

principal bolsa de valores do Brasil e a 20ª do mundo em termos de capitalização total do mercado), e sua configuração para otimizar o equilíbrio entre desempenho e custo computacional.

Na fase de definição do *corpus*, é apresentado o *corpus* usado no pré-treinamento do DeB3RTa, que incorpora dados de várias fontes externas, como fatos relevantes, patentes, Scielo, Wikipedia e notícias.

O estágio de definição da *baseline* inclui quatro modelos de propósito geral baseados em *Transformers* (Multilingual BERT, BERTimbau, XLM-RoBERTa e DistilBERT), dois modelos de domínio específico (SEC-BERT e BusinessBERT), e um grupo separado de cinco modelos GPT (GPT-3.5-turbo, GPT-4o-mini, GPT-4o, GPT-4-turbo e GPT-4).

Durante a fase de definição dos *datasets*, quatro deles são incorporados: OFFCOMBR-3, FAKE.BR, CAROSIA e BBRC (*Brazilian Banking Regulation Corpora*). Esses *datasets* servem de base para treinar e testar o DeB3RTa e os modelos da *baseline*.

Além disso, na fase de *fine-tuning* do modelo, foram realizadas tarefas *downstream* no DeB3RTa e nos modelos de *baseline*; nessas tarefas, foram aplicadas técnicas para melhorar o desempenho do DeB3RTa em comparação com os modelos de *baseline*.

Por fim, a fase de avaliação avalia o desempenho de todos esses modelos por meio do F1 *score*, do *recall*, da precisão e da PR-AUC (*Area Under Precision-Recall Curve*), fornecendo uma medida padronizada de eficácia entre as diferentes abordagens.

A metodologia proposta nesta tese foi estruturada como um *pipeline* sequencial de seis etapas, refletindo práticas consolidadas em Ciência de Dados e Inteligência Artificial aplicadas ao Processamento de Linguagem Natural. Inicialmente, realizou-se a seleção e configuração do modelo base, etapa em que foi definida a arquitetura DeBERTa como fundação do DeB3RTa. Em seguida, foi construída uma base textual especializada a partir de múltiplas fontes, compondo o *corpus* para pré-treinamento, de modo a captar as especificidades linguísticas do domínio financeiro. A terceira e a quarta etapas consistiram na definição dos modelos *baseline* (para fins comparativos) e na seleção dos *datasets* de avaliação, cobrindo diferentes tarefas e desafios do PLN em português. A quinta etapa envolveu o pré-treinamento do modelo DeB3RTa sobre o *corpus* curado, seguido pelo *fine-tuning* supervisionado utilizando técnicas avançadas como *Mixout*, média estocástica dos pesos e decaimento da taxa de aprendizado por camada, garantindo robustez, adaptabilidade e generalização em contextos variados.

Para facilitar a compreensão do encadeamento e da natureza das etapas, foi elaborada uma descrição gráfica da metodologia na forma de fluxograma vertical, apresentada na Figura 4.2. Nela, as caixas são coloridas de acordo com a categoria predominante de cada etapa: Modelagem (em verde claro), Dados (em azul claro) e Treinamento (em laranja claro). Essa categorização tem por objetivo evidenciar os diferentes focos ao longo do

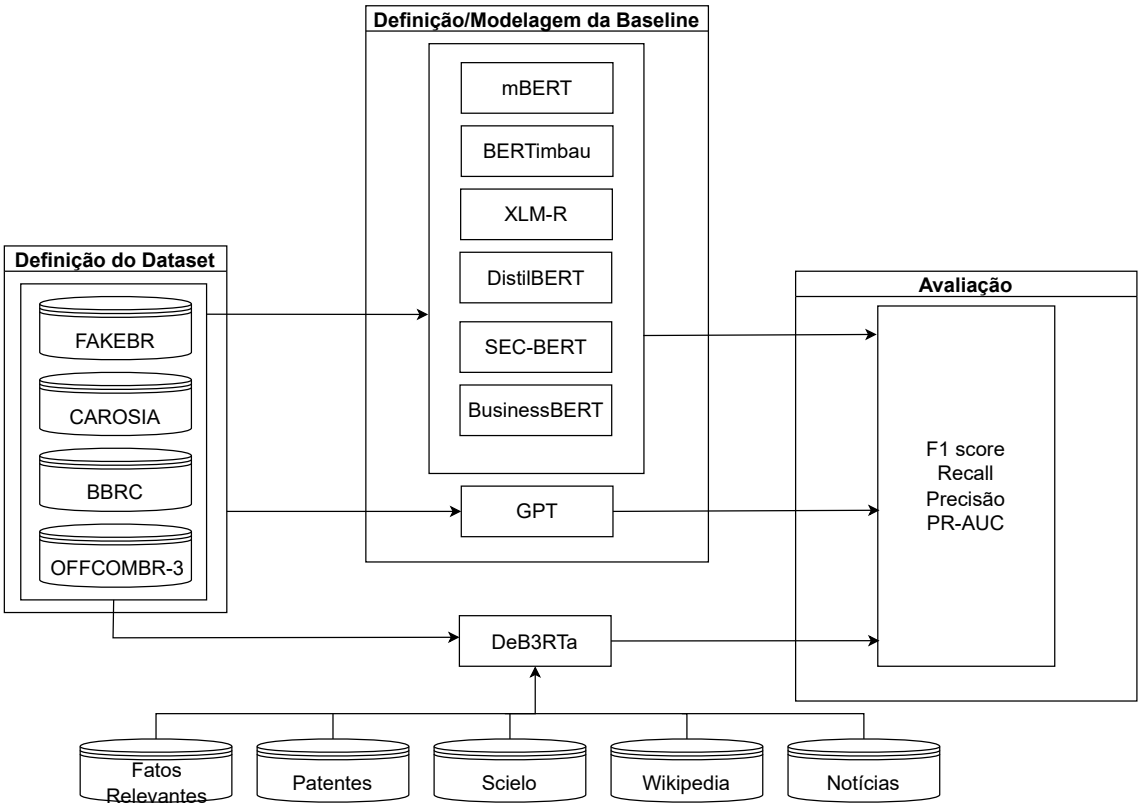

Figura 4.1 – Fluxo de trabalho para desenvolvimento e avaliação do DeB3RTa e dos modelos de *baseline*. Autoria própria.

pipeline — conceitual, informacional e computacional —, permitindo ao leitor visualizar com clareza o fluxo metodológico e a interação entre seus componentes. A legenda associada à figura reforça esse agrupamento conceitual, contribuindo para a organização visual e lógica do processo descrito.

4.2.1 Etapa 1: Seleção do Modelo e Configuração

O modelo escolhido para o estudo foi o DeBERTa-v2 [295], no qual foram feitas configurações personalizadas. Em sua configuração básica, o DeBERTa-v2 tem 24 cabeças de atenção, 24 camadas, dimensão da camada intermediária igual a 6144 e dimensão das camadas ocultas igual a 1536, com aproximadamente 887 milhões de parâmetros treináveis. Criamos duas versões do nosso modelo: uma versão base com 12 cabeças de atenção, 12 camadas, dimensão da camada intermediária igual a 3.072 e dimensão das camadas ocultas igual a 768, totalizando aproximadamente 426 milhões de parâmetros treináveis, e uma versão menor com 6 cabeças de atenção, 12 camadas, dimensão da camada intermediária igual a 1.536 e dimensão das camadas ocultas igual a 384, totalizando por volta de 70 milhões de parâmetros treináveis. Esses valores de hiperparâmetros foram escolhidos por meio de avaliações preliminares, que buscaram verificar valores que proporcionassem um equilíbrio entre generalização e capacidade de representação e custo computacional.

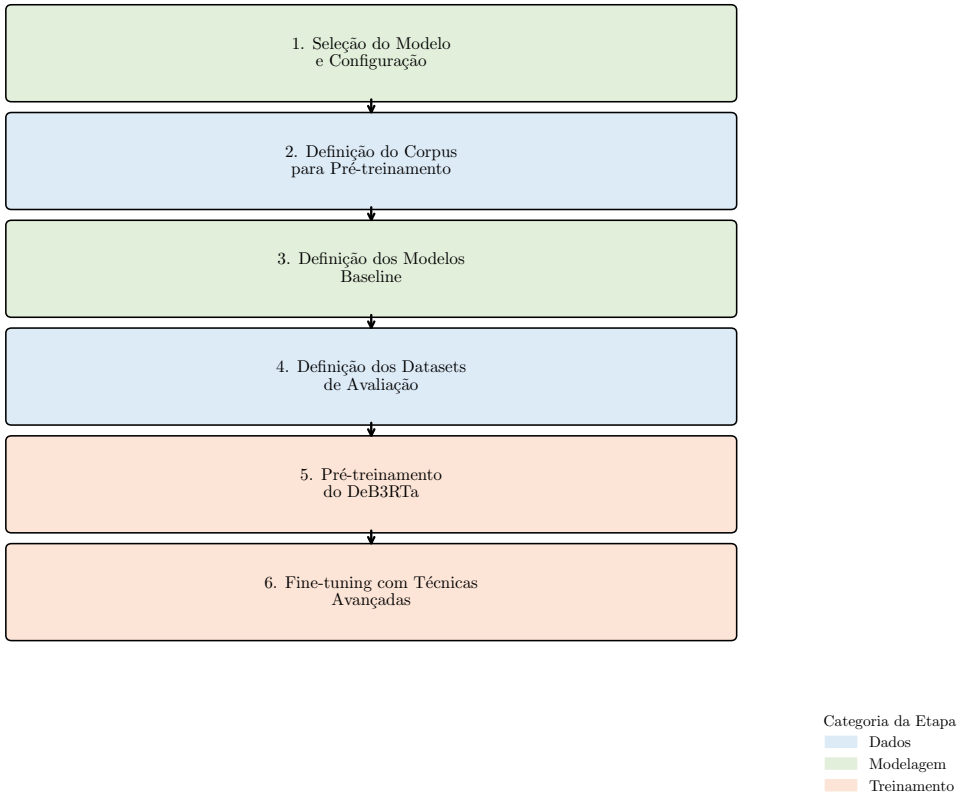

Figura 4.2 – Descrição Gráfica da Metodologia. Autoria própria.

Escolhemos o DeBERTa com base em sua superioridade comprovada em relação a outros codificadores, levando a melhorias documentadas de +0,9% no MNLI, +2,3% no SQuAD v2.0 e +3,6% no RACE em comparação com o RoBERTa-large. Além disso, ele demonstrou um desempenho que supera o nível humano no *benchmark* SuperGLUE. Essas conquistas podem ser atribuídas à incorporação de dois recursos inovadores: atenção desemaranhada, que permite que o modelo se concentre de forma independente em diferentes aspectos da sequência de entrada, e um decodificador de máscara aprimorado.

4.2.2 Etapa 2: Definição do *Corpus* para Pré-treinamento do DeB3RTa

O *corpus* de pré-treinamento integra dados extraídos por meio de *scraping* da Web de várias fontes, incluindo artigos de notícias, patentes e relatórios financeiros, garantindo a relevância do *dataset* para o domínio financeiro. Especificamente, o *corpus* consiste no seguinte:

- **Fatos Relevantes**: Os dados referem-se aos fatos relevantes das empresas apresentadas na carteira de maio-agosto de 2023 do IbrX 100[1], que indica o

[1] https://www.b3.com.br/pt_br/market-data-e-indices/indices/indices-amplos/indice-brasil-100-ibrx-100.htm

desempenho dos 100 ativos mais significativos do mercado acionário brasileiro. Essas informações foram obtidas do banco de dados da Comissão de Valores Mobiliários[2] e abrangem o período de 2003 a 2023;

- ***Google Patents***: O *dataset* compreende as categorias G06Q, G07C, G07F e G07G do Sistema Internacional de Classificação de Patentes (IPC)[3], abrangendo patentes registradas no período de 2006 a 2021;

- **Scielo**: O *dataset* inclui artigos de pesquisa em português no Scielo[4] brasileiro sobre finanças, política, economia, gestão de negócios e contabilidade, abrangendo o período de 1961 a 2023;

- **Wikipedia**: Em 19 de maio de 2023, usamos a biblioteca Python Wikipedia-API[5] para extrair artigos em português. Começamos explorando a categoria "Economia" e incluímos até cinco subcategorias. Descartamos *links* para artigos idênticos para garantir a qualidade dos dados e evitar a repetição de conteúdo;

- **Notícias**: O *dataset* contém artigos de jornais eletrônicos especializados e de grande circulação no Brasil, Portugal e Angola. Esses artigos abrangem finanças, economia, política e tópicos relacionados e foram selecionados entre 1999 e 2023.

Cada corpus passou por um processo de limpeza, antes da consolidação em um *corpus* só; esse processo envolveu a segregação dos dados em frases individuais, a remoção de frases com ruído ou mal construídas e a aplicação do algoritmo MinHash [319], conforme descrito por [320], para remover entradas duplicadas. Depois de concluir essa etapa, mesclamos todo o *corpus* e realizamos um processo de deduplicação adicional para remover o conteúdo redundante, resultando em um total aproximado de 1,05 bilhão de *tokens*. Estatísticas detalhadas de cada *corpus* podem ser encontradas na Tabela 4.1.

Tabela 4.1 – Resumo do *corpus* financeiro por fonte, incluindo contagem de *tokens*, sentenças e documentos.

Corpus	***Tokens***	**Sentenças**	**Documentos**
Fatos Relevantes	8.983.628	200.396	9.581
Google Patents	40.306.323	1.055.227	7.973
Scielo	98.167.792	4.779.149	10.498
Wikipedia	73.925.719	2.548.436	102.140
News	828.975.676	27.107.736	2.347.217
Total	**1.050.359.095**	**35.690.944**	**2.477.409**

[2] https://www.gov.br/cvm/pt-br
[3] https://ipcpub.wipo.int/
[4] https://www.scielo.br/
[5] https://github.com/martin-majlis/Wikipedia-API

4.2.3 Etapa 3: Definição da *Baseline*

Na fase inicial de nosso estudo, realizamos uma avaliação sistemática para identificar os modelos mais eficazes para nossas tarefas e otimizar seu desempenho. Selecionamos quatro modelos de linha de base: mBERT, BERTimbau, XLM-RoBERTa e DistilBERT. Esses modelos foram escolhidos com base em sua eficácia comprovada em várias tarefas de PLN, conforme relatado na literatura, e em sua capacidade de lidar com o idioma português.

O mBERT é reconhecido por sua versatilidade em vários idiomas, o que o torna adequado para contextos linguísticos variados; o BERTimbau foi projetado especificamente para o português do Brasil, garantindo que ele capture as características linguísticas exclusivas desse idioma; o XLM-RoBERTa oferece recursos multilíngues robustos, que são essenciais para tarefas que envolvem *datasets* multilíngues, enquanto o DistilBERT oferece uma alternativa mais eficiente com tamanho de modelo reduzido e tempos de inferência mais rápidos.

Além desses modelos de *baseline*, foram selecionados dois modelos de domínio específico para os experimentos: SEC-BERT e BusinessBERT, ambos treinados em *corpora* em inglês. Esses modelos foram usados por meio de *cross-lingual transfer*, o que permite aproveitar os recursos de aprendizagem de modelos desenvolvidos em um idioma, onde existem recursos abundantes, para realizar tarefas em outro idioma.

Além desses modelos de linha de base, os LLMs de última geração foram incorporados ao experimento. Especificamente, o GPT-3.5-turbo, o GPT-4o-mini, o GPT-4o, o GPT-4-turbo e o GPT-4 foram utilizados para as tarefas. Esses modelos foram selecionados devido a seus recursos avançados e ao sucesso demonstrado em várias aplicações do PLN [321].

4.2.4 Etapa 4: Definição dos *Datasets*

Nossa pesquisa incluiu quatro tarefas de classificação de texto para avaliar a eficácia do nosso modelo: detecção de discurso de ódio, detecção de notícias falsas, análise de sentimentos, e análise de documentos.

Foi demonstrado que as condições econômicas, como o desemprego e a desigualdade de renda, influenciam a incidência de crimes de ódio. Por exemplo, taxas de desemprego mais altas estão associadas ao aumento de crimes de ódio violentos, o que sugere que o estresse econômico pode exacerbar as tensões sociais e levar a crimes motivados por preconceito [322], [323]. Além disso, a estrutura econômica dos crimes de ódio sugere que os indivíduos podem pesar os benefícios intrínsecos de cometer tais crimes em relação aos possíveis custos, incluindo a estima social e as repercussões legais, que podem ser influenciadas pela prevalência do discurso de ódio na sociedade. O discurso de ódio pode

atuar como um sinal de atitudes sociais, potencialmente normalizando preconceitos e incentivando crimes de ódio quando os indivíduos percebem um ambiente de apoio para seus preconceitos [324]. Além disso, o discurso de ódio *on-line* tem sido associado a crimes de ódio *off-line*, indicando que as expressões digitais de preconceito podem se traduzir em violência no mundo real, complicando ainda mais o cenário econômico e social [325].

Quando as informações são disseminadas, é essencial considerar cuidadosamente o impacto das notícias falsas. Identificar e erradicar a desinformação é fundamental, principalmente devido ao seu potencial de influenciar indivíduos e economias inteiras. Rumores falsos e notícias enganosas podem afetar significativamente os preços das ações e a disposição de se envolver em investimentos de grande escala. Portanto, é fundamental abordar essa questão com o máximo de cuidado [326].

A análise de sentimentos é vital em finanças, pois pode prever tendências, identificar possíveis crises e orientar decisões de investimento. Dada a grande quantidade de dados no setor financeiro, a análise de sentimentos tornou-se crucial para analistas e investidores. Ao examinar grandes *datasets* e reconhecer padrões de sentimento do mercado, a análise de sentimento pode oferecer percepções valiosas sobre o comportamento do mercado financeiro e ajudar os investidores a tomar decisões bem informadas [327].

A análise de documentos no setor financeiro e bancário é fundamental por vários motivos, principalmente no que diz respeito à eficiência, à precisão, à conformidade e à tomada de decisões. A classificação eficaz é crucial para o gerenciamento dessas informações, pois o setor financeiro gera grandes quantidades de dados e documentação [328].

Os testes nos quais o *fine-tuning* foi implementado se deram por meio de quatro *datasets*:

- **OFFCOMBR-3** [329]: O OFFCOMBR-2 foi originalmente compilado com 1.250 comentários rotulados por três anotadores, com níveis variados de concordância. Com base nisso, os autores criaram o OFFCOMBR-3, um *dataset* mais refinado de 1.033 comentários, incluindo apenas aqueles para os quais todos os anotadores chegaram a um acordo unânime. Desses, 202 comentários (19,5%) foram rotulados como ofensivos, enquanto os 831 restantes foram considerados não ofensivos, tornando o *dataset* desequilibrado;
- **FAKE.BR** [330]: Os autores selecionaram um *dataset* de 7.200 artigos de notícias rotulados manualmente como legítimos ou falsos. O *dataset* incluiu um número igual de 3.600 artigos de notícias falsos e legítimos, com cada artigo falso emparelhado com um artigo preciso de tamanho semelhante. A maioria dos artigos foi publicada entre janeiro de 2016 e janeiro de 2018. Cada artigo foi submetido a uma verificação manual para garantir que contivesse apenas informações falsas, evitando a inclusão de meias-verdades. Os artigos foram então categorizados em seis tópicos: economia;

ciência e tecnologia; sociedade e notícias diárias; política; religião; e TV e celebridades. No entanto, somente os artigos categorizados em economia e política foram utilizados para as tarefas desempenhadas pelos modelos, pois esses tópicos estão intimamente ligados ao domínio financeiro, resultando em 4.224 artigos de notícias;

- **CAROSIA** [331]: O autor compilou atualizações de notícias sobre o mercado financeiro brasileiro, incluindo 717 relatórios positivos e 553 negativos de fontes confiáveis, como G1, Estadão e Folha de São Paulo. Essas notícias abrangem o índice do mercado acionário brasileiro, Ibovespa[6], e o desempenho de empresas importantes listadas no mercado acionário brasileiro, como Banco do Brasil, Itaú, Gerdau e Ambev;

- **BBRC** [332]: O *dataset* BBRC consiste em 25 *corpora* contendo dados de risco regulatório bancário de várias divisões do Banco do Brasil. Esses *corpora* abrangem uma ampla gama de tópicos, incluindo investimentos, seguros, recursos humanos, segurança, tecnologia, tesouraria, empréstimos, contabilidade, fraude, cartões de crédito, métodos de pagamento, agronegócios e gerenciamento de riscos. O *dataset* inclui 61.650 documentos anotados, a maioria dos quais tem de meia a três páginas. O artigo original detalha dois experimentos sobre análise de documentos e, em nossos testes, seguimos a metodologia do segundo experimento, com a única modificação sendo a eliminação de entradas duplicadas. Após essa etapa, nosso *dataset* consistiu em 337 documentos classificados como relevantes e 295 classificados como irrelevantes.

Neste estudo, nenhuma técnica de balanceamento de classe, como superamostragem (*oversampling*) ou subamostragem (*undersampling*), foi aplicada a nenhum dos *datasets*. Cada *dataset* foi usado para refletir as distribuições naturais das classes. Para a fase de *fine-tuning*, cada *dataset* foi dividido em subconjuntos de treinamento, validação e teste, seguindo uma proporção de 80/10/10, com amostragem estratificada para preservar as distribuições de classe originais em todos os subconjuntos. Estatísticas detalhadas sobre cada divisão de cada *dataset* podem ser encontradas nas Tabelas 4.2 a 4.5.

Tabela 4.2 – Estatísticas descritivas do *dataset* OFFCOMBR-3.

Subconjunto	**Med. do núm. de palavras**	**Mín. do núm. de palavras**	**Máx. do núm. de palavras**	**Não Disc. Ódio**	**Disc. Ódio**
Treino	11	1	91	664	162
Validação	12	1	94	83	20
Teste	10	1	65	84	20

[6] https://www.b3.com.br/pt_br/market-data-e-indices/indices/indices-amplos/ibovespa.htm

Tabela 4.3 – Estatísticas descritivas do *dataset* FAKE.BR.

Subconjunto	**Med. do núm. de palavras**	**Mín. do núm. de palavras**	**Máx. do núm. de palavras**	**Falsas**	**Legítimas**
Treino	348	10	7.517	1.689	1.690
Validação	358,5	10	4.050	211	211
Teste	354	17	4.891	211	212

Tabela 4.4 – Estatísticas descritivas do *dataset* CAROSIA.

Subconjunto	**Med. do núm. de palavras**	**Mín. do núm. de palavras**	**Máx. do núm. de palavras**	**Negativas**	**Positivas**
Treino	13	2	26	442	574
Validação	13	6	24	55	72
Teste	13	5	24	56	71

Tabela 4.5 – Estatísticas descritivas do *dataset* BBRC.

Subconjunto	**Med. do núm. de palavras**	**Mín. do núm. de palavras**	**Máx. do núm. de palavras**	**Irrelevantes**	**Relevantes**
Treino	526	52	219.610	236	269
Validação	480	71	18.849	29	34
Teste	700	127	92.041	30	34

4.2.5 Etapa 5: Pré-treinamento do DeB3RTa

Em ambas as versões do modelo, utilizamos o tokenizador DeBERTa-v2 xlarge para realizar a tokenização, baseado no SentencePiece [333] e que emprega unidades de subpalavra [284] e modelagem de linguagem com unigramas [334]. A sequência de *tokens* foi truncada para 128 *tokens* com preenchimento dinâmico, e o vocabulário foi definido com um tamanho de 128.100. Durante o treinamento, realizado em uma NVIDIA A100 (com duração de 103 horas para o modelo completo e 83 horas para o modelo menor), aplicamos o mascaramento BERT padrão, com uma probabilidade de 15% de mascaramento para cada exemplo.

Foi utilizado o otimizador AdamW com uma taxa de aprendizado de 1×10^{-4} e decaimento linear, reservando o primeiro 1% das 80.650 etapas para aquecimento. O tamanho total do lote foi de 1.536 frases, composto por 192 amostras e com acúmulo de gradiente. Os modelos foram treinados por 50 épocas, e os hiperparâmetros específicos foram cuidadosamente selecionados com base em testes exploratórios extensivos e em pesquisas de pesquisadores com recursos limitados [164]. A Figura 4.4 apresenta a progressão da perda durante a convergência do modelo base.

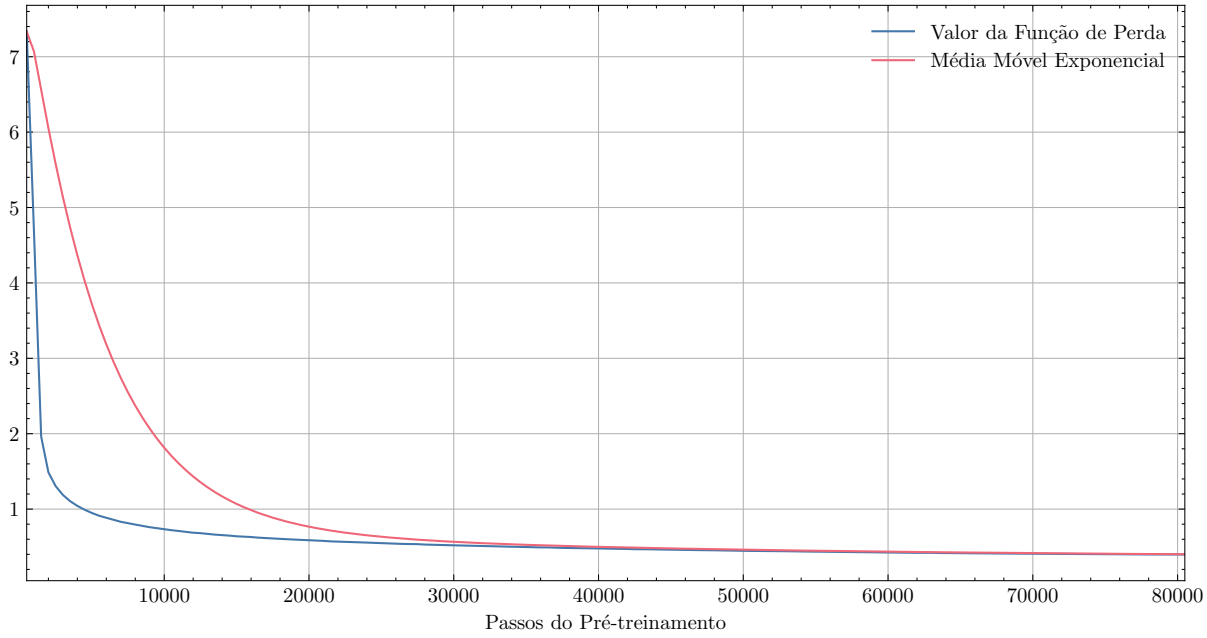

Figura 4.3 – Convergência do modelo DeB3RTa base: valor da função de perda do pré-treinamento com média móvel exponencial (fator de suavização: 0,9). Autoria própria.

Para alcançar os benefícios significativos de redução de uso de memória e computação mais rápida, o que é particularmente vantajoso para modelos de linguagem pré-treinados com altos requisitos computacionais e grande demanda de memória, realizamos o treinamento do nosso modelo utilizando FP16 (também conhecido como formato de ponto flutuante de precisão reduzida) [335]. Esse método utiliza 16 bits para representar um número de ponto flutuante, oferecendo um intervalo menor de valores representáveis em comparação com o formato padrão FP32. A diminuição da precisão permite uma computação mais rápida e um menor uso de memória, tornando-o adequado para tarefas de treinamento e inferência, especialmente quando são usados hardwares especializados otimizados para menor precisão. No entanto, a precisão reduzida inerente ao FP16 pode impactar a precisão numérica, particularmente em cenários de treinamento mais complexos.

4.2.6 Etapa 6: *Fine-tuning* do Modelo

Nos esforços de otimização para o DeB3RTa durante as tarefas de aplicação prática, foram aplicadas várias técnicas, incluindo os otimizadores AdamW, AdamP, RAdam e MADGRAD, a reinicialização de grupos de camadas, a aplicação de decaimento da taxa de aprendizado por camada e o uso da regularização *Mixout*. Nossos experimentos preliminares mostraram que o melhor desempenho foi alcançado ao usar o agendador (*scheduler*) de taxa de aprendizado cosseno com aquecimento (*warmup*) em configurações que utilizavam implementações específicas de *schedulers* (AdamP, RAdam, MADGRAD e LLRD), em vez

do *scheduler* linear com *warmup* usado na configuração padrão da biblioteca Huggingface[7].

Extensas buscas de hiperparâmetros foram realizadas para estabelecer a configuração ideal para cada experimento mencionado. Esse processo envolveu o uso de busca em grade (*grid search*) para explorar sistematicamente uma variedade de configurações de hiperparâmetros, como taxas de aprendizado e fatores de decaimento, a fim de identificar o melhor desempenho nos *datasets*. O desempenho do modelo foi avaliado usando várias métricas: F1 *score*, precisão, *recall* e PR-AUC. Durante a otimização dos hiperparâmetros, a pontuação F1 *macro* foi escolhida como o único critério para selecionar a configuração ideal. Essa métrica fornece uma avaliação equilibrada ao ponderar igualmente cada classe, mitigando o risco de viés em direção à classe majoritária em *datasets* desequilibrados.

Os subconjuntos de treinamento e validação foram empregados durante o procedimento de busca em grade para o desenvolvimento do modelo e a avaliação subsequente de desempenho. O subconjunto de treinamento foi usada para ajustar os modelos, enquanto o de validação permitiu avaliar diferentes configurações de hiperparâmetros. Posteriormente, o subconjunto de teste foi utilizado para avaliar o desempenho dos modelos com as configurações de hiperparâmetros ideais identificadas durante o processo de busca em grade.

O *fine-tuning* dos modelos foi feito com os seguintes hiperparâmetros: sentenças de comprimento máximo de 128 *tokens* com preenchimento (*padding*, estratégia que adiciona um *token* de preenchimento especial para garantir que as sequências mais curtas tenham o mesmo comprimento que o comprimento máximo aceito pelo modelo) e truncamento (*trunking*, estratégia que funciona truncando sequências maiores que o comprimento máximo); quatro épocas (conforme recomendado por [1], é um número de épocas que funciona bem em todas as tarefas); período de *warmup* nos 10% dos passos iniciais de treinamento, já que modelos *Transformers* geralmente apresentam dificuldades para estabilizar o aprendizado se não incorporarem uma taxa de aprendizado gradual no início do treinamento [336].

Devido a variações nos tamanhos dos *datasets*, as tarefas no *dataset* FAKE.BR foram realizadas com tamanhos de lote de treinamento de {32, 64}, enquanto as tarefas nos *datasets* OFFCOMBR-3, CAROSIA e BBRC utilizaram tamanhos de lote de {16, 32}. Além disso, todas as configurações e modelos para o DeB3RTa seguiram os hiperparâmetros especificados nas Tabelas 4.6 a 4.9 para o conjunto completo de hiperparâmetros de cada configuração do DeB3RTa.

Para a configuração de reinicialização de camadas, foram reinicializadas as camadas 10, 11 ou 12. Para configurar o SWA, além das taxas de aprendizagem padrão, foram usadas algumas taxas de aprendizagem a serem aplicadas a partir de um determinado

[7] https://huggingface.co/

ponto do treinamento, no qual o SWA é aplicado: 1×10^{-6}, 2×10^{-6}, 3×10^{-6}, 4×10^{-6}, 5×10^{-6}. De acordo com o método proposto por [337], seguido neste trabalho, a fase SWA é iniciada logo após a conclusão de 50% das etapas de treinamento no *fine-tuning*; a partir deste momento, é usada uma taxa de aprendizagem mais baixa e constante.

A probabilidade p do *Mixout* foi testada em cinco valores discretos: 0,1, 0,3, 0,5, 0,7 e 0,9. No LLRD, diferentes taxas de aprendizado são aplicadas a diferentes camadas da rede. A taxa de aprendizado base é determinada inicialmente e, em seguida, geralmente multiplicada por um fator maior que 1 para definir a taxa de aprendizado da camada mais alta (geralmente a camada específica da tarefa, também chamada de camada de *pooling*). Para cada camada anterior, a taxa de aprendizado é reduzida por um fator de decaimento, resultando em taxas de aprendizado progressivamente mais baixas para as camadas mais próximas da entrada. Adicionalmente, um conjunto de valores para o decaimento de peso foi aplicado para evitar *overfitting*, adicionando um termo de regularização à função de perda que penaliza pesos altos e incentiva o modelo a manter pesos menores e mais generalizáveis. A Figura 4.4 mostra um exemplo dos valores da taxa de aprendizado durante os passos de treinamento em uma configuração de LLRD.

Tabela 4.6 – Configurações do *grid search* baseado nos otimizadores.

Modelo	**Taxa de aprendizado**	**Otimizador**
DeB3RTa base/menor	$\{1 \times 10^{-5}; 2 \times 10^{-5}; 3 \times 10^{-5}; 4 \times 10^{-5}; 5 \times 10^{-5}\}$	{AdamW; AdamP; RAdam; MADGRAD}

Tabela 4.7 – Configurações do *grid search* baseado na reinicialização de camadas.

Modelo	**Taxa de aprendizado**	**Camada reinicializada**
DeB3RTa base	$\{1 \times 10^{-5}; 2 \times 10^{-5}; 3 \times 10^{-5}; 4 \times 10^{-5}; 5 \times 10^{-5}\}$	{10; 11; 12}

Tabela 4.8 – Configurações do *grid search* baseado na média estocástica de pesos.

Modelo	**Taxa de aprendizado**	**Taxa de aprendizado SWA**
DeB3RTa base (SWA)	$\{1 \times 10^{-5}; 2 \times 10^{-5}; 3 \times 10^{-5}; 4 \times 10^{-5}; 5 \times 10^{-5}\}$	$\{1 \times 10^{-6}; 2 \times 10^{-6}; 3 \times 10^{-6}; 4 \times 10^{-6}; 5 \times 10^{-6}\}$

Os modelos da *baseline* foram ajustados com suas configurações padrão, enquanto os modelos GPT foram empregados em uma configuração *zero-shot*, com a temperatura do modelo ajustada para zero para produzir saídas mais determinísticas. Os modelos foram acessados por meio de chamadas à API da OpenAI[8], onde foram fornecidos *prompts* para a execução das tarefas de classificação. Apesar dos *datasets* serem compostos por textos na língua portuguesa, os *prompts* foram usados em língua inglesa, devido a estudos que

[8] https://platform.openai.com/docs/api-reference

Tabela 4.9 – Configurações do *grid search* baseado na regularização *Mixout* e LLRD.

Modelo	**Taxa de aprendizado**	**Hiperparâmetros**
DeB3RTa base (*Mixout*)	$\{1 \times 10^{-5}; 2 \times 10^{-5}; 3 \times 10^{-5}; 4 \times 10^{-5}; 5 \times 10^{-5}\}$	Probabilidade p: $\{0.1; 0.3; 0.5; 0.7; 0.9\}$
DeB3RTa base (LLRD)	$\{1 \times 10^{-4}; 2 \times 10^{-4}; 3 \times 10^{-4}; 4 \times 10^{-4}; 5 \times 10^{-4}\}$	Taxa de decaimento: $\{0.9; 0.95\}$ Multiplicador da camada de *pooling*: $\{1.02; 1.03\}$ Decaimento de peso: $\{1 \times 10^{-4}; 1 \times 10^{-3}; 1 \times 10^{-2}; 1 \times 10^{-1}\}$

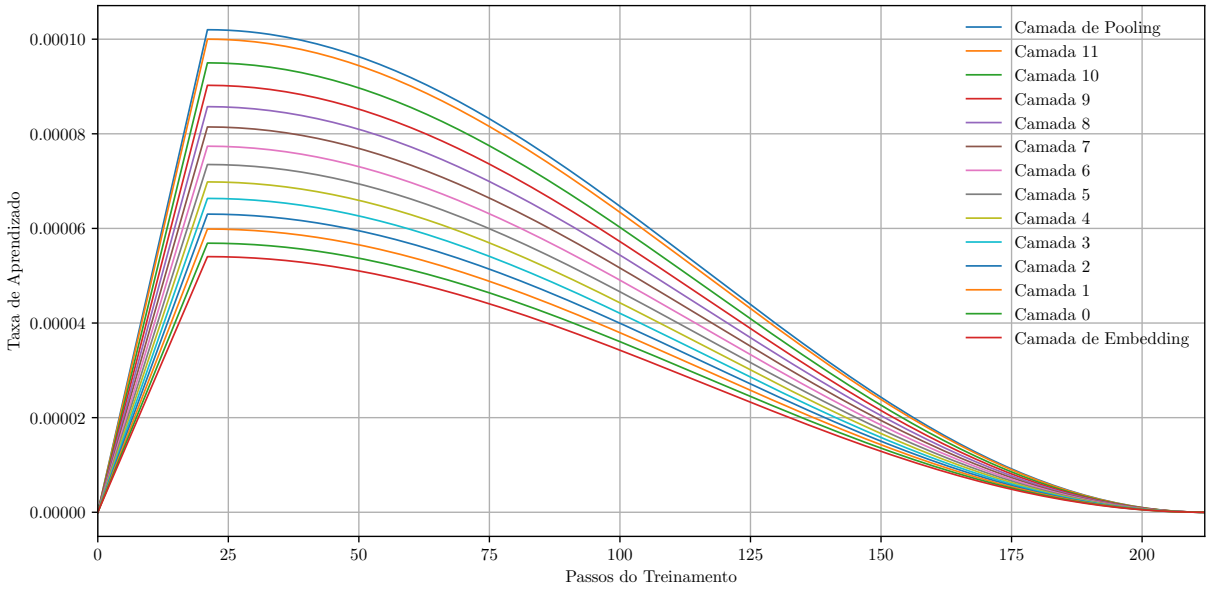

Figura 4.4 – Progressão do decaimento da taxa de aprendizado por camada: taxa de aprendizado inicial = 1e-4, taxa de decaimento = 0.95, multiplicador da camada de *pooling* = 1.02; etapas iniciais de *warmup* e progressão cossenoidal com taxas de aprendizado para as camadas de *embedding*, 0-11 e *pooling*. Autoria própria.

concluíram que modelos GPT têm desempenhos melhores quando os *prompts* são escritos nessa língua [338], [339]. Os *prompts* estão especificados na Tabela 4.10.

4.3 Considerações Finais

O estudo apresenta o desenvolvimento do DeB3RTa, um modelo de linguagem especializado em processamento de linguagem financeira em português, com duas versões (base e menor) baseadas na arquitetura DeBERTa-v2. O modelo foi pré-treinado com um corpus abrangente de aproximadamente 1,05 bilhão de *tokens*, incorporando dados de diversas fontes como fatos relevantes empresariais, patentes, artigos científicos, Wikipedia

Tabela 4.10 – *Prompts* usados pelos modelos GPT para tarefas de classificação nos *datasets* OFFCOMBR-3, FAKE.BR, CAROSIA e BBRC.

Dataset	***Prompt***
OFFCOMBR-3	Classify the following web comment as either hate speech or not hate speech. Message: '`texto do dataset`'. The output should only contain two words: hate or not hate.
FAKE.BR	Classify the following text of news articles as either legitimate (true information) or fake (false information). Message: '`texto do dataset`'. The output should only contain two words: legitimate or fake.
CAROSIA	Classify the following text of news updates about the Brazilian financial market as either positive (indicating favorable market conditions) or negative (indicating unfavorable conditions). Message: '`texto do dataset`'. The output should only contain two words: positive or negative.
BBRC	Classify the following text of banking regulatory risk from different departments of Banco do Brasil as either relevant (impacting departmental compliance and operations) or not relevant (not impacting departmental compliance). Message: '`texto do dataset`'. The output should only contain two words: relevant or not relevant.

e notícias.

A avaliação do modelo foi conduzida através de comparação com *baselines* estabelecidas, incluindo modelos como mBERT, BERTimbau e variantes do GPT, utilizando quatro *datasets* distintos focados em detecção de discurso de ódio, detecção de notícias falsas, análise de sentimentos em notícias financeiras e análise de documentos bancários regulatórios. O processo de *fine-tuning* incorporou técnicas avançadas de otimização, como diferentes otimizadores, reinicialização de camadas e regularização *Mixout*, sendo o treinamento realizado com precisão reduzida (FP16) para otimizar recursos computacionais mantendo o desempenho.

5 Testes e Resultados

A avaliação do modelo foi conduzida por meio de uma comparação com *baselines* estabelecidas, que incluíram mBERT, BERTimbau, XLM-RoBERTa, DistilBERT, SEC-BERT e BusinessBERT, e diversos modelos GPT. O desempenho foi testado em quatro *datasets* distintos: OFFCOMBR-3, utilizado para detecção de discurso de ódio, FAKE.BR, utilizado para detecção de notícias falsas; CAROSIA, voltado para análise de sentimentos em notícias financeiras; e BBRC, empregado na análise de documentos bancários regulatórios. A pesquisa envolveu uma comparação minuciosa do modelo com o BERTimbau, um modelo de domínio geral em português, com modelos multilíngues de domínio geral que também suportam o português, como o mBERT, XLM-RoBERTa e o DistilBERT, modelos de domínio específico, como o SEC-BERT e o BusinessBERT, e modelos da família GPT. Os resultados detalhados dessa comparação podem ser consultados nas Tabelas 5.1 e 5.2.

O processo de *fine-tuning* do modelo foi realizado utilizando técnicas avançadas de otimização, como o emprego de diferentes otimizadores (AdamW, AdamP, RAdam, MADGRAD), a reinicialização de camadas, a média estocástica de pesos, a regularização *Mixout* e o decaimento da taxa de aprendizado por camada. Além disso, o treinamento foi conduzido com precisão reduzida (FP16) para otimizar o uso de recursos computacionais, garantindo, assim, um equilíbrio entre eficiência e desempenho.

5.1 Prototipagem e Ferramentas Computacionais Utilizadas

Para a realização dos experimentos descritos neste capítulo, todas as etapas foram desenvolvidas utilizando a linguagem de programação Python 3, amplamente adotada nas áreas de Ciência de Dados e Aprendizado de Máquina. A prototipagem do *pipeline* foi conduzida de forma modular e reprodutível, com foco em flexibilidade para testes, análise de resultados e controle de experimentos.

O modelo DeB3RTa, bem como os modelos de *baseline*, foram implementados com o suporte da biblioteca `transformers`, mantida pela Huggingface. Essa biblioteca oferece uma interface padronizada para modelos baseados na arquitetura Transformer, além de ferramentas para tokenização, treinamento supervisionado, avaliação e salvamento de *checkpoints*. Para o carregamento de métricas padronizadas e *datasets*, foi utilizada a biblioteca `datasets`, também da Huggingface, que fornece integração direta com métricas como F1 *macro*, amplamente utilizada nas avaliações desta tese.

A busca pelos hiperparâmetros ideais foi conduzida com o uso da biblioteca `optuna`, uma ferramenta robusta de otimização baseada em técnicas de busca automática, que

permite definir espaços de busca customizados e realizar experimentos com múltiplos critérios. Nesta tese, utilizou-se o *sampler* `BruteForceSampler`, garantindo exaustividade nos valores pré-definidos de hiperparâmetros.

O processo de *fine-tuning*, tanto do DeB3RTa quanto das baselines, foi realizado em computadores equipados com placas de vídeo NVIDIA GeForce GTX TITAN X (12 GB de memória), o que possibilitou a execução eficiente de experimentos com grande volume de dados e modelos de alta complexidade.

Outras bibliotecas acessórias, porém fundamentais para a implementação, incluem:

- `numpy` e `pandas`, para manipulação de dados;
- `torch` (PyTorch), como *backend* para o treinamento dos modelos;
- `argparse`, para parametrização via linha de comando;
- `os`, para controle do ambiente de execução e organização de diretórios.

A fim de garantir a reprodutibilidade dos resultados, foi adotada uma configuração determinística de execução, com a fixação explícita das sementes pseudoaleatórias em todas as bibliotecas envolvidas (PyTorch, NumPy e Python).

5.2 Resultados

O dataset OFFCOMBR-3 apresentou desafios únicos devido ao seu desequilíbrio significativo de classe (19,5% de instâncias de discurso de ódio). Embora o gpt-3.5-turbo tenha obtido a pontuação F1 mais alta, de 0,8157, o desempenho do DeB3RTa merece uma análise mais detalhada pelas lentes do PR-AUC, que reflete melhor o desempenho em datasets desequilibrados. O modelo básico DeB3RTa com a configuração AdamP obteve um PR-AUC de 0,8081, superando significativamente outros modelos e demonstrando um desempenho robusto e independente de limitações. Isso é fundamental para aplicações do mundo real, nas quais certos valores, como as proporções de distribuição de classes das instâncias, podem variar.

Na tarefa de classificação de notícias financeiras FAKE.BR, embora o XLM-RoBERTa grande tenha alcançado a pontuação F1 mais alta de 0,9953, o desempenho do DeB3RTa mostrou-se notavelmente competitivo. O modelo básico do DeB3RTa com otimização MADGRAD obteve uma pontuação F1 de 0,9906, ficando aquém em apenas 0,47%. Esse desempenho é particularmente notável, considerando a contagem de parâmetros significativamente menor do DeB3RTa. O modelo também demonstrou estabilidade excepcional em diferentes configurações, com várias variantes (MADGRAD, LLRD,

Tabela 5.1 – F1 *scores* e PR-AUC nos *datasets* (valores mais altos em negrito sublinhado).

F1 *Score*				
Modelo	**OFFCOMBR-3**	**FAKE.BR**	**CAROSIA**	**BBRC**
DeB3RTa menor (AdamW)	0,5460	0,9598	0,8722	0,6712
DeB3RTa base (AdamW)	0,6836	0,9858	0,9120	0,7460
DeB3RTa base (AdamP)	0,7424	0,9858	0,8795	**0,7609**
DeB3RTa base (RAdam)	0,7206	0,9835	0,9038	0,7490
DeB3RTa base (MADGRAD)	0,7539	0,9906	0,9207	0,7176
DeB3RTa base (SWA)	0,6737	0,9716	0,8722	0,7290
DeB3RTa base (Reinic. Camadas)	0,7080	0,9811	0,9201	0,7143
DeB3RTa base (Mixout)	0,7102	0,9811	0,9038	0,7013
DeB3RTa base (LLRD)	0,7424	0,9882	0,8627	0,6995
mBERT base	0,6172	0,9835	0,8789	0,6995
BERTimbau base	0,6877	0,9858	**0,9363**	0,7117
BERTimbau large	0,6780	0,9906	0,9280	0,6797
XLM-RoBERTa base	0,6737	0,9811	0,8326	0,6085
XLM-RoBERTa large	0,4468	**0,9929**	0,9123	0,6246
DistilBERT base	0,6810	0,9740	0,8550	0,6297
BusinessBERT	0,6494	0,9693	0,7233	0,7143
SEC-BERT	0,6905	0,9693	0,8326	0,7478
gpt-3.5-turbo	**0,8157**	0,6407	0,6600	0,5205
gpt-4o-mini	0,7984	0,7285	0,9280	0,4921
gpt-4o	0,6590	0,7754	0,8424	0,5873
gpt-4-turbo	0,6590	0,8345	0,9207	0,5377
gpt-4	0,6586	0,6698	0,8892	0,5536

PR-AUC				
Modelo	**OFFCOMBR-3**	**FAKE.BR**	**CAROSIA**	**BBRC**
DeB3RTa menor (AdamW)	0,6813	0,9925	0,8957	0,7812
DeB3RTa base (AdamW)	0,7534	0,9948	0,9605	0,8315
DeB3RTa base (AdamP)	**0,8081**	0,9943	0,9740	0,8290
DeB3RTa base (RAdam)	0,7424	0,9943	0,9523	0,7861
DeB3RTa base (MADGRAD)	0,7911	0,9960	0,9725	0,8203
DeB3RTa base (SWA)	0,7424	0,9952	0,9356	**0,8402**
DeB3RTa base (Reinic. Camadas)	0,7840	0,9937	0,9711	0,8125
DeB3RTa base (Mixout)	0,7774	0,9947	0,9605	0,8105
DeB3RTa base (LLRD)	0,7913	0,9964	0,9362	0,7527
mBERT base	0,6295	0,9931	0,9726	0,7702
BERTimbau base	0,7180	0,9972	**0,9776**	0,8157
BERTimbau large	0,7534	0,9991	0,9714	0,7367
XLM-RoBERTa base	0,6622	0,9946	0,8689	0,6508
XLM-RoBERTa large	0,5607	**0,9993**	0,9705	0,6738
DistilBERT base	0,6418	0,9970	0,9164	0,7233
BusinessBERT	0,6962	0,9952	0,8250	0,8046
SEC-BERT	0,7352	0,9966	0,8971	0,7927
gpt-3.5-turbo	0,7392	0,7531	0,8360	0,7157
gpt-4o-mini	0,7135	0,8214	0,9529	0,6667
gpt-4o	0,6471	0,8442	0,9164	0,7268
gpt-4-turbo	0,6971	0,8759	0,9559	0,6693
gpt-4	0,6978	0,7959	0,9382	0,7337

Tabela 5.2 – *Recall* e precisão nos *datasets* (valores mais altos em negrito sublinhado).

Recall				
Modelo	**OFFCOMBR-3**	**FAKE.BR**	**CAROSIA**	**BBRC**
DeB3RTa menor (AdamW)	0,5452	0,9598	0,8722	0,6716
DeB3RTa base (AdamW)	0,6643	0,9858	0,9112	0,7451
DeB3RTa base (AdamP)	0,7202	0,9858	0,8774	**0,7598**
DeB3RTa base (RAdam)	0,7083	0,9835	0,9023	0,7490
DeB3RTa base (MADGRAD)	0,7262	0,9906	0,9239	0,7176
DeB3RTa base (SWA)	0,6583	0,9716	0,8722	0,7284
DeB3RTa base (Reinic. Camadas)	0,6893	0,9811	0,9201	0,7137
DeB3RTa base (Mixout)	0,7024	0,9811	0,9023	0,7010
DeB3RTa base (LLRD)	0,7333	0,9882	0,8595	0,6990
mBERT base	0,6298	0,9835	0,8755	0,6990
BERTimbau base	0,6774	0,9858	**0,9380**	0,7118
BERTimbau large	0,6714	0,9906	0,9272	0,6804
XLM-RoBERTa base	0,6583	0,9811	0,8332	0,6088
XLM-RoBERTa large	0,5000	**0,9929**	0,9131	0,6294
DistilBERT base	0,6845	0,9740	0,8525	0,6324
BusinessBERT	0,6262	0,9692	0,7271	0,7137
SEC-BERT	0,6905	0,9692	0,8332	0,7471
gpt-3.5-turbo	**0,8464**	0,6590	0,6986	0,5343
gpt-4o-mini	0,7762	0,7404	0,9272	0,4941
gpt-4o	0,6250	0,7829	0,8516	0,5882
gpt-4-turbo	0,6250	0,8345	0,9239	0,5402
gpt-4	0,6250	0,6929	0,8939	0,5657
Precisão				
Modelo	**OFFCOMBR-3**	**FAKE.BR**	**CAROSIA**	**BBRC**
DeB3RTa menor (AdamW)	0,5990	0,9598	0,8722	0,6711
DeB3RTa base (AdamW)	0,7190	0,9858	0,9129	0,7530
DeB3RTa base (AdamP)	0,7772	0,9858	0,8826	**0,7718**
DeB3RTa base (RAdam)	0,7366	0,9835	0,9055	0,7490
DeB3RTa base (MADGRAD)	0,8016	0,9906	0,9193	0,7176
DeB3RTa base (SWA)	0,6993	0,9716	0,8722	0,7390
DeB3RTa base (Reinic. Camadas)	0,7382	0,9811	0,9201	0,7206
DeB3RTa base (Mixout)	0,7196	0,9811	0,9058	0,7020
DeB3RTa base (LLRD)	0,7532	0,9882	0,8688	0,7032
mBERT base	0,6104	0,9835	0,8852	0,7032
BERTimbau base	0,7015	0,9858	**0,9352**	0,7250
BERTimbau large	0,6860	0,9906	0,9289	0,6917
XLM-RoBERTa base	0,6993	0,9811	0,8321	0,6085
XLM-RoBERTa large	0,4038	**0,9929**	0,9117	0,6310
DistilBERT base	0,6779	0,9740	0,8594	0,6432
BusinessBERT	0,7255	0,9696	0,7240	0,7206
SEC-BERT	0,6905	0,9696	0,8321	0,7500
gpt-3.5-turbo	0,7947	0,7008	0,7452	0,5409
gpt-4o-mini	0,8295	0,7901	0,9289	0,4939
gpt-4o	**0,9242**	0,8242	0,8485	0,5911
gpt-4-turbo	0,9242	0,8345	0,9193	0,5421
gpt-4	0,9235	0,7554	0,8886	0,5784

reinicialização de camadas) atingindo consistentemente pontuações de F1 *score* acima de 0,98.

No dataset CAROSIA para análise de sentimentos financeiros, o BERTimbau base obteve o maior F1 *score* de 0,9363. O DeB3RTa demonstrou um bom desempenho, com sua configuração de base (MADGRAD) atingindo 0,9207 e mantendo pontuações de precisão (0,9193) e *recall* (0,9239) notavelmente equilibradas. Esse equilíbrio entre precisão e recuperação é particularmente valioso para a análise de sentimentos financeiros, em que tanto os falsos positivos quanto os falsos negativos podem ter implicações significativas. A variante pequena do DeB3RTa obteve uma pontuação F1 de 0,8722, superando vários modelos maiores, como o XLM-RoBERTa base (0,8326).

Para a tarefa de classificação de documentos regulatórios do BBRC, o DeB3RTa base com o otimizador AdamP obteve a maior pontuação geral de F1 de 0,7609, superando todos os modelos de *baseline*, incluindo modelos financeiros especializados como o SEC-BERT (0,7478) e o BusinessBERT (0,7143). Esse resultado é particularmente significativo, pois representa uma melhoria substancial de 8,12% em relação ao próximo melhor modelo (BERTimbau large com 0,6797). Ainda mais notável é o fato de que o modelo menor do DeB3RTa, com sua contagem reduzida de parâmetros, obteve uma pontuação F1 de 0,6712, superando modelos maiores, como o XLM-RoBERTa large (0,6246).

A análise das pontuações PR-AUC em todos os datasets revela a capacidade consistente do DeB3RTa de manter um bom desempenho em diferentes pontos de operação. Isso é particularmente evidente no dataset BBRC, em que a base DeB3RTa com a configuração SWA alcançou um PR-AUC de 0,8402, e no FAKE.BR, em que várias configurações mantiveram pontuações PR-AUC acima de 0,99. Esses resultados indicam um desempenho robusto, independentemente do limite de classificação escolhido.

Os resultados experimentais também destacam o desempenho consistente do DeB3RTa em diferentes comprimentos de texto e complexidades de domínio. De pequenos trechos de notícias financeiras no CAROSIA a longos documentos regulatórios no BBRC, o modelo manteve um desempenho competitivo usando menos parâmetros do que alguns de seus concorrentes. Essa compensação entre eficiência e desempenho é particularmente notável na variante menor, que superou consistentemente os modelos de *baseline* maiores em várias tarefas.

Em suma, a avaliação revela cinco pontos fortes principais do DeB3RTa:

1. Desempenho robusto em *datasets* desequilibrados, evidenciado por pontuações PR-AUC superiores;

2. Desempenho competitivo com significativamente menos parâmetros do que os modelos atuais de última geração;

3. desempenho consistente em diferentes comprimentos de texto;

4. equilíbrio entre precisão e *recall* em todas as tarefas.

Essas descobertas demonstram a eficácia do DeB3RTa como um modelo versátil e eficiente em termos de recursos para tarefas de processamento de textos financeiros em português.

5.3 Discussão

Os resultados experimentais demonstram a eficácia do DeB3RTa em diversas tarefas do domínio financeiro, com desempenho particularmente forte na tarefa de classificação de documentos regulatórios (BBRC) e resultados competitivos na tarefa de detecção de notícias falsas (FAKE.BR), na tarefa de análise de sentimentos (CAROSIA) e na tarefa de detecção de discurso de ódio (OFFCOMBR-3). Embora não atinja consistentemente os F1 *scores* mais altos, o DeB3RTa demonstra eficiência e estabilidade notáveis em diferentes características de tarefas e distribuições de dados.

A análise do desempenho dos otimizadores revela *insights* importantes sobre o *fine-tuning* do modelo. Entre os otimizadores testados (AdamW, AdamP, RAdam e MADGRAD), o MADGRAD foi o que apresentou o desempenho mais consistente, destacando-se principalmente nas tarefas FAKE.BR (F1 = 0,9906) e CAROSIA (F1 = 0,9207). No entanto, o AdamP mostrou-se superior para o BBRC (F1 = 0,7881) e apresentou pontos fortes no OFFCOMBR-3 (PR-AUC = 0,8081). Essa variabilidade ressalta a importância da seleção de otimizadores específicos para cada tarefa, em vez de adotar uma abordagem única para todos.

As técnicas avançadas de fine-tuning demonstraram graus variados de eficácia nos datasets. Embora esses métodos tenham o objetivo de evitar o *overfitting*, seu impacto foi notavelmente dependente do *dataset*. No *dataset* do BBRC, as configurações mais simples geralmente superaram as mais complexas, com a configuração básica do AdamW obtendo melhores resultados do que as versões que usam reinicialização de camada, *Mixout* ou LLRD. Isso sugere que a regularização excessiva pode ter impedido o modelo de capturar padrões importantes em textos regulatórios complexos.

O desempenho das variantes GPT, especialmente nos datasets FAKE.BR e BBRC, revelou limitações interessantes. Apesar de sua arquitetura sofisticada e do maior número de parâmetros, esses modelos apresentaram desempenho notavelmente inferior em comparação com as abordagens baseadas na arquitetura *Transformer*. No FAKE.BR, o gpt-3.5-turbo alcançou uma pontuação F1 de apenas 0,6407, enquanto no BBRC, alcançou apenas 0,5205, substancialmente abaixo do desempenho do DeB3RTa. Essa disparidade foi, portanto,

particularmente evidente em tarefas que exigiam conhecimento especializado no domínio financeiro.

O baixo desempenho dos modelos GPT nessas tarefas de classificação provavelmente decorre de sua arquitetura generativa e da abordagem *zero-shot*. Ao contrário do que ocorre durante o *fine-tuning* dos modelos baseados na arquitetura *Transformer*, esses modelos dependem de engenharia imediata e não têm otimização específica para a tarefa. Isso fica evidente em um estudo em que o GPT-3 foi usado para classificar perguntas relacionadas à ciência de dados. Os pesquisadores relataram que o aumento do conjunto de treinamento com exemplos adicionais gerados pelo próprio GPT-3 melhorou significativamente a precisão da classificação, mas ainda ficou aquém da precisão humana [5], [340].

Em comparação com estudos anteriores, que mostram que modelos como o FinBERT e o SEC-BERT alcançam melhorias significativas em tarefas financeiras, os resultados aqui ressaltam a importância do *fine-tuning* específico da tarefa. Por exemplo, os modelos GPT, como o GPT-3.5 e o GPT-4, superam o *fine-tuning* dos modelos não-gerativos em tarefas de classificação de texto *few-shot*, ou seja, uma tarefa em que o objetivo é classificar o texto em diferentes categorias usando apenas um pequeno número de exemplos rotulados. No entanto, o desempenho é significativamente melhor quando os modelos são fornecidos com amostras representativas selecionadas por especialistas humanos [10].

A presente pesquisa ressalta a importância das camadas superiores para acelerar o aprendizado e melhorar o desempenho do modelo. De fato, os resultados da busca em grade nos *datasets* FAKE.BR, CAROSIA e BBRC revelam *insights* importantes sobre a relação entre a estratégia de incorporação posicional do DeB3RTa e seu comportamento de fine-tuning, conforme mostrado na Tabela 5.3. Como o DeBERTa (e, por extensão, o DeB3RTa) incorpora *embeddings* posicionais relativos das camadas 1 a 10 e *embeddings* posicionais absolutas nas camadas 11 e 12 [341], a reinicialização ideal de diferentes camadas nos datasets sugere necessidades específicas da tarefa para lidar com informações posicionais. Isso difere da arquitetura do BERT, que aplica *embeddings* posicionais absolutas em todas as 12 camadas. A abordagem híbrida do DeB3RTa parece oferecer uma vantagem na captura de relacionamentos matizados em diferentes níveis de representação de texto, permitindo que o modelo se adapte de forma eficaz entre as tarefas.

Para o *dataset* OFFCOMBR-3 (tarefa de detecção de discurso de ódio), a reinicialização da camada 10 obteve a maioria das pontuações F1 mais altas. Isso sugere que a tarefa de detecção de discurso de ódio se beneficiou da preservação dos pesos originalmente treinados nas camadas iniciais (1-9) que lidam com as *embeddings* posicionais relativas, o que faz sentido porque o discurso de ódio geralmente se baseia em relações contextuais entre palavras e frases que podem ser mais bem melhor capturadas por *embeddings* posicionais relativos.

Para o *dataset* FAKE.BR (tarefa de detecção de notícias falsas), reinicializar as

Tabela 5.3 – Cinco melhores resultados de busca em grade para os *datasets* OFFCOMBR-3, FAKE.BR, CAROSIA e BBRC com camadas reinicializadas e hiperparâmetros variados (valores mais altos de cada dataset em negrito).

Dataset	**F1 *Score***	**Camada Reinic.**	**Taxa de Aprendizado**	**Tamanho de Lote**
OFFCOMBR-3	**0,8260**	10	4×10^{-5}	32
	0,8239	11	5×10^{-5}	32
	0,8139	10	3×10^{-5}	32
	0,7980	10	5×10^{-5}	16
	0,7944	10	3×10^{-5}	16
FAKE.BR	**0,9905**	12	1×10^{-5}	64
	0,9905	11	1×10^{-5}	64
	0,9905	10	5×10^{-5}	64
	0,9905	12	4×10^{-5}	32
	0,9905	11	2×10^{-5}	32
CAROSIA	**0,9120**	11	3×10^{-5}	16
	0,9038	10	4×10^{-5}	32
	0,9033	12	4×10^{-5}	16
	0,8960	11	2×10^{-5}	16
	0,8872	12	5×10^{-5}	16
BBRC	**0,7597**	10	5×10^{-5}	16
	0,7597	11	4×10^{-5}	16
	0,7597	11	2×10^{-5}	16
	0,7580	12	5×10^{-5}	16
	0,7580	12	4×10^{-5}	16

camadas 10, 11 ou 12 resultou em um desempenho quase idêntico (F1 = 0,9905), indicando que as *embeddings* posicionais absolutas e relativas desempenharam papéis igualmente importantes na detecção de desinformação. De fato, notícias falsas geralmente seguem padrões estilísticos enganosos e inconsistências estruturais que exigem que as relações contextuais locais e a estrutura global do documento sejam identificadas com eficácia. O desempenho uniforme entre as camadas sugere que a detecção de notícias falsas se beneficiou de uma abordagem híbrida que equilibrou essas duas formas de percepção posicional.

Para o *dataset* CAROSIA (tarefa de análise de sentimentos), a maior parte do desempenho ideal foi observada ao reinicializar a camada 11 ou 12. Isso indica que a tarefa se beneficiou do uso de *embeddings* posicionais absolutas, que são introduzidas nas duas camadas finais do DeB3RTa. Nas notícias financeiras em português, a análise de sentimentos pode ser fortemente influenciada pela posição de termos ou frases-chave em uma frase. Por exemplo, a palavra “queda” pode mudar de um significado negativo para um significado positivo, dependendo de sua posição na frase e das frases ao redor (por exemplo, “queda nos juros” ou “declínio nas taxas de juros”). A posição das palavras de negação ou intensificação (por exemplo, “não” ou “muito”) também é fundamental,

pois pode alterar significativamente o sentimento, assim como as formas comparativas e superlativas (por exemplo, “maior lucro” ou “menor lucro”). As *embeddings* posicionais absolutas permitem que o modelo capture essas nuances estruturais, compreendendo não apenas as relações entre as palavras, mas também suas posições específicas na frase. Ao reinicializar as camadas 11 ou 12, o modelo aproveita essas informações posicionais para fazer previsões de sentimento mais precisas com base em como as frases carregadas de sentimento estão posicionadas no texto.

Para o *dataset* BBRC (tarefa de classificação de documentos regulatórios), a reinicialização da camada 10 ou 11 produziu a maior pontuação F1 (0,7597), com a camada 12 logo em seguida (F1 = 0,7580). Esse padrão sugere que a classificação de documentos regulatórios exigiu uma integração equilibrada de *embeddings* posicionais absolutas e relativas. Os textos regulatórios seguem formatos estruturados com cláusulas legais, referências cruzadas e organização hierárquica, em que as relações locais entre os termos e seu posicionamento no documento mais amplo são igualmente importantes. A eficácia quase igual das camadas 10, 11 e 12 destacou a necessidade de um modelo que possa se adaptar tanto às dependências em nível de seção quanto à estrutura de todo o documento ao classificar estes documentos acerca de riscos regulatórios.

Além disso, os resultados da busca em grade sobre a regularização do *Mixout* destacam seu impacto, conforme mostrado na Tabela 5.4. A capacidade do *Mixout* de substituir parâmetros por seus valores pré-treinados parece apoiar a generalização, especialmente em *datasets* menores, como o CAROSIA, atenuando o ajuste excessivo. No entanto, ao contrário das descobertas anteriores [256], os melhores resultados foram frequentemente obtidos com probabilidades de *Mixout* mais baixas (por exemplo, 0,1-0,3) em vez dos 0,7-0,9 sugeridos. Isso sugere que, embora o *Mixout* seja benéfico, sua configuração ideal pode depender das características do *dataset*, destacando a necessidade de *fine-tuning* com base na avaliação empírica. Os F1 *scores* observados reforçam a função do *Mixout* como um método de regularização eficaz, embora com um intervalo de probabilidade diferente do inicialmente esperado.

Observando a Tabela 5.5, é possível observar vários padrões importantes nas configurações de LLRD nos *datasets*.

O *dataset* OFFCOMBR-3 obteve a maior pontuação F1 (0,8577) com uma taxa de aprendizado de 2×10^{-4}, taxa de decaimento de 0,90 e decaimento de peso de 1×10^{-1}. O modelo mostrou consistência em diferentes taxas de decaimento (0,90 e 0,95), mantendo um bom desempenho. O *dataset* FAKE.BR teve o melhor desempenho com uma taxa de aprendizado de 1×10^{-4}, sugerindo que as atualizações conservadoras foram benéficas. A configuração superior (F1 = 0,9953) usou uma taxa de decaimento de 0,95 e decaimento de peso de 1×10^{-4}, reforçando a importância da regularização em tarefas de alta precisão. O *dataset* CAROSIA atingiu sua pontuação F1 mais alta (0,9033) com uma taxa de

Tabela 5.4 – Cinco melhores resultados de busca em grade para os *datasets* OFFCOMBR-3, FAKE.BR, CAROSIA e BBRC com *Mixout* e hiperparâmetros variados (valores mais altos de cada dataset em negrito).

Dataset	**F1 *Score***	***Mixout***	**Taxa de Aprendizado**	**Tamanho de Lote**
OFFCOMBR-3	**0,8449**	0,1	5×10^{-5}	16
	0,826	0,7	4×10^{-5}	16
	0,826	0,1	2×10^{-5}	32
	0,8188	0,3	3×10^{-5}	32
	0,8188	0,5	5×10^{-5}	16
FAKE.BR	**0,9953**	0,1	1×10^{-5}	64
	0,9953	0,1	4×10^{-5}	32
	0,9929	0,3	1×10^{-5}	64
	0,9929	0,9	3×10^{-5}	64
	0,9929	0,3	3×10^{-5}	64
CAROSIA	**0,8867**	0,7	2×10^{-5}	32
	0,8795	0,5	5×10^{-5}	16
	0,8789	0,9	3×10^{-5}	16
	0,8789	0,1	1×10^{-5}	16
	0,8782	0,9	4×10^{-5}	16
BBRC	**0,7444**	0,1	2×10^{-5}	16
	0,7429	0,3	5×10^{-5}	16
	0,7291	0,9	5×10^{-5}	32
	0,7277	0,3	4×10^{-5}	32
	0,7277	0,3	4×10^{-5}	16

aprendizado de 2×10^{-4}, taxa de decaimento de 0,90 e decaimento de peso de 1×10^{-3}, apoiando a ideia de que a tarefa se beneficiou de taxas de aprendizado moderadas. O *dataset* BBRC exigiu taxas de aprendizado mais altas (3×10^{-4}) para obter o desempenho ideal, com sua melhor pontuação F1 (0,7580) observada usando uma taxa de decaimento de 0,95 e decaimento de peso de 1×10^{-1} ou 1×10^{-2}, sugerindo que a tarefa se beneficiou de atualizações maiores.

Em todos os datasets, as taxas de aprendizagem entre 1×10^{-4} e 3×10^{-4} produziram consistentemente os melhores resultados, enquanto as taxas de aprendizagem mais altas não produziram os melhores F1 *scores* em todos os *datasets*. Isso corrobora pesquisas anteriores sobre os riscos de ultrapassar as soluções ideais em modelos de transformadores ao usar taxas de aprendizagem excessivas [342]. O efeito das taxas de decaimento (0,90 e 0,95) foi geralmente mínimo, com o desempenho sendo mais fortemente influenciado pela taxa de aprendizado e pelo decaimento do peso. Entretanto, os valores de decaimento do peso (1×10^{-4} a 1×10^{-1}) tiveram um impacto moderado, mas dependente do *dataset*, indicando que sua influência variava com base nas características específicas da tarefa. Além disso, o multiplicador do *pooler* comportou-se de maneira consistente em ambos os valores possíveis, sugerindo um comportamento de *fine-tuning* estável em todos os *datasets*.

Tabela 5.5 – Cinco melhores resultados de busca em grade para os *datasets* OFFCOMBR-3, FAKE.BR, CAROSIA e BBRC com LLRD e hiperparâmetros variados (valores mais altos de cada dataset em negrito).

Dataset	**F1 *Score***	**Taxa de Aprendizado**	**Dec. Taxa**	**Dec. Peso**	**Multipl. Pooler**	**Tamanho Lote**
OFFCOMBR-3	**0.8577**	2×10^{-4}	0.90	1×10^{-1}	1.03	16
	0.8449	2×10^{-4}	0.95	1×10^{-3}	1.02	16
	0.8387	2×10^{-4}	0.90	1×10^{-4}	1.03	16
	0.8387	2×10^{-4}	0.90	1×10^{-3}	1.03	16
	0.8387	2×10^{-4}	0.95	1×10^{-1}	1.03	16
FAKE.BR	**0.9953**	1×10^{-4}	0.95	1×10^{-4}	1.03	32
	0.9953	1×10^{-4}	0.90	1×10^{-1}	1.03	64
	0.9929	2×10^{-4}	0.90	1×10^{-2}	1.03	64
	0.9929	1×10^{-4}	0.95	1×10^{-1}	1.03	64
	0.9929	2×10^{-4}	0.90	1×10^{-4}	1.02	64
CAROSIA	**0.9033**	2×10^{-4}	0.90	1×10^{-3}	1.02	16
	0.8950	3×10^{-4}	0.90	1×10^{-4}	1.03	16
	0.8872	3×10^{-4}	0.90	1×10^{-4}	1.02	32
	0.8853	1×10^{-4}	0.90	1×10^{-4}	1.03	32
	0.8853	1×10^{-4}	0.90	1×10^{-2}	1.03	32
BBRC	**0.7580**	3×10^{-4}	0.95	1×10^{-1}	1.02	32
	0.7580	3×10^{-4}	0.95	1×10^{-2}	1.02	32
	0.7277	2×10^{-4}	0.90	1×10^{-1}	1.02	16
	0.7232	3×10^{-4}	0.95	1×10^{-4}	1.02	32
	0.7232	2×10^{-4}	0.90	1×10^{-4}	1.03	16

O desempenho do DeB3RTa em todos os quatro *datasets*, incluindo sua variante menor, demonstra consistência, adaptabilidade e eficiência notáveis. No FAKE.BR, a variante básica alcança resultados competitivos, aproximando-se dos modelos de melhor desempenho, enquanto no CAROSIA, ela oferece um desempenho sólido, superando vários modelos baseados em transformadores. Esses resultados enfatizam que a arquitetura DeB3RTa, combinada com o pré-treinamento de domínio misto, oferece uma abordagem equilibrada para lidar com textos de tamanhos variados e jargões financeiros complexos, o que a torna altamente adequada para aplicações financeiras em que os recursos computacionais podem ser limitados. Além das técnicas de *fine-tuning*, os resultados sugerem que a versão menor do DeB3RTa mantém um desempenho competitivo em relação a modelos muito maiores, como o XLM-RoBERTa e o GPT-4, o que o torna uma opção mais prática para implantação em instituições financeiras que exigem análise em tempo real com recursos computacionais limitados.

5.4 Análise de Falha

Na tarefa de detecção de discurso de ódio do OFFCOMBR-3, a melhor configuração do DeB3RTa obteve uma pontuação F1 de 0,7539 (com o otimizador MADGRAD), com desempenho significativamente inferior ao 0,8157 do gpt-3.5-turbo. Essa diferença ficou ainda mais acentuada com a variante menor, que atingiu apenas 0,5460. A precisão de 0,8016 indica que, quando o modelo identifica algo como discurso de ódio, ele está correto em 80,16% das vezes. No entanto, a recuperação de 0,7262 mostra que o modelo identifica com sucesso apenas 72,62% de todas as instâncias de discurso de ódio no *dataset*.

No *dataset* de análise de sentimentos CAROSIA, embora o DeB3RTa tenha demonstrado um desempenho sólido e equilibrado (F1 = 0,9207, precisão = 0,9193, *recall* = 0,9239), ainda há uma lacuna de desempenho em comparação com o BERTimbau base (F1 = 0,9363), indicando que há espaço para melhorias mesmo em tarefas em que o modelo apresenta um desempenho sólido de *baseline*.

A limitação mais significativa aparece na redução do tamanho do modelo. A variante menor do DeB3RTa apresentou uma degradação considerável do desempenho em todas as tarefas:

- OFFCOMBR-3: Queda de 0.7539 para 0.5460 (F1 *score*);
- CAROSIA: apesar de superar os modelos maiores, como o XLM-RoBERTa base (0,8326), a pontuação F1 da variante menor de 0,8722 ainda representa uma queda notável em relação aos 0,9207 do modelo base;
- BBRC: Queda de 0.7609 para 0.6712 (F1 *score*).

Esse padrão consistente de degradação com a arquitetura menor indica um desafio significativo na manutenção do desempenho ao reduzir o tamanho do modelo. A tarefa de classificação de documentos regulatórios do BBRC destaca especialmente essa limitação, em que a queda de quase 9 pontos percentuais sugere que a compreensão de documentos complexos é altamente sensível à capacidade do modelo.

5.5 Considerações Finais

Este capítulo apresentou uma análise abrangente da eficácia do DeB3RTa em tarefas do domínio financeiro em português, comparando seu desempenho com diversas *baselines* estabelecidas. Os resultados obtidos demonstraram que o DeB3RTa alcançou um desempenho competitivo e, em alguns casos, superior em relação a modelos maiores e mais complexos, destacando-se particularmente na classificação de documentos regulatórios e mantendo resultados robustos nas demais tarefas.

A investigação comparativa entre diferentes técnicas de otimização revelou *insights* valiosos sobre o comportamento do modelo em diversos cenários. O otimizador MADGRAD demonstrou consistência notável em várias tarefas, enquanto o AdamP se destacou em contextos específicos, evidenciando que a seleção do otimizador deve ser adaptada às características de cada tarefa. A avaliação de técnicas avançadas de *fine-tuning*, como reinicialização de camadas, regularização *Mixout* e LLRD, revelou a importância dessas abordagens para melhorar o desempenho do modelo, embora com impactos variáveis dependendo do *dataset* e da tarefa.

A análise detalhada das métricas de desempenho, incluindo F1 *score*, PR-AUC, precisão e *recall*, proporcionou uma visão holística da eficácia do modelo. O DeB3RTa demonstrou robustez particular em *datasets* desequilibrados, como evidenciado pelos valores de PR-AUC superiores no OFFCOMBR-3, e apresentou um equilíbrio notável entre precisão e *recall* em tarefas críticas como a análise de sentimentos em notícias financeiras.

Um aspecto particularmente relevante dos resultados foi o desempenho competitivo do DeB3RTa com significativamente menos parâmetros que modelos como XLM-RoBERTa large e GPT-4. Esta eficiência paramétrica representa uma vantagem considerável para implementações em ambientes com recursos computacionais limitados, sem comprometer substancialmente a qualidade das predições.

A análise comparativa com modelos GPT revelou limitações interessantes desta família de modelos em tarefas específicas de classificação financeira, particularmente nos *datasets* FAKE.BR e BBRC. Este resultado destaca a importância do *fine-tuning* específico para domínios especializados e reforça o valor de modelos como o DeB3RTa, que podem ser otimizados para necessidades específicas.

Apesar dos resultados promissores, a análise de falhas identificou oportunidades de melhoria, especialmente na variante menor do DeB3RTa, que apresentou degradação de desempenho em várias tarefas. Esta limitação sugere a necessidade de técnicas adicionais de otimização para redução de parâmetros sem comprometer significativamente o desempenho.

Por fim, este estudo contribui para o campo do processamento de linguagem natural em português ao estabelecer *benchmarks* rigorosos em diversas tarefas financeiras, demonstrando a viabilidade de modelos eficientes como o DeB3RTa e fornecendo *insights* metodológicos sobre técnicas de *fine-tuning*. Os resultados obtidos oferecem uma base sólida para futuros desenvolvimentos de modelos linguísticos para o domínio financeiro em português, equilibrando eficiência computacional e desempenho robusto em aplicações práticas.

6 Conclusão

6.1 Objetivos Alcançados

Ao término deste trabalho, pode-se concluir, com base nos resultados obtidos, que a pesquisa confirmou a hipótese levantada e atendeu plenamente aos objetivos gerais e específicos estabelecidos no início. O uso de modelos baseados em *Transformers*, exemplificados pelo DeB3RTa, mostrou-se altamente adequado para o desenvolvimento de soluções no contexto de processamento de linguagem natural aplicada ao domínio financeiro de língua portuguesa. Além disso, a metodologia adotada foi considerada eficaz para alcançar os objetivos propostos e realizar os procedimentos necessários para validar a hipótese. O embasamento bibliográfico coletado também desempenhou um papel crucial, permitindo a fundamentação teórica que norteou o desenvolvimento do modelo final.

Foi observado que, embora o modelo DeB3RTa tenha se destacado em termos de desempenho e adaptabilidade, outras abordagens, como uso de outras técnicas de regularização e otimização, uso de modelos mais leves sem que haja perda significativa de desempenho e ampliação do suporte linguístico, podem contribuir de maneira igualmente significativa para o avanço de soluções no PLN financeiro. A integração de diferentes métodos e técnicas, combinados com abordagens modernas de *fine-tuning*, como regularização de *Mixout* e reinicialização de camadas, pode potencialmente gerar resultados ainda mais robustos.

Por fim, a execução dos testes destacou a relevância de projetar soluções eficientes em termos computacionais, especialmente em domínios onde o equilíbrio entre desempenho e consumo de recursos é fundamental. Esse aspecto reforçou a necessidade de desenvolver modelos otimizados, como o DeB3RTa, que conciliam alta performance com uso eficiente de recursos, uma questão previamente enfatizada no referencial bibliográfico deste trabalho.

6.2 Limitações

A validade interna do estudo é, em geral, forte, dado o uso de *datasets* confiáveis. Entretanto, como em qualquer análise de dados, existe a possibilidade de pequenas inconsistências no pré-processamento e na anotação dos dados. O tamanho menor desses conjuntos de dados, de fato, pode facilitar a verificação completa, embora ainda possam surgir pequenos problemas.

O tamanho menor desses conjuntos de dados, embora permita uma verificação mais completa, também pode representar uma possível ameaça à validade dos resultados. Para

resolver esse problema, técnicas de regularização, como a redução de peso, são empregadas neste trabalho para ajudar a atenuar o excesso de ajuste, um desafio comum em conjuntos de dados menores. No entanto, com divisões de treinamento variando de 505 a 3.379 instâncias e divisões de teste de 64 a 423 instâncias, o tamanho limitado dos dados ainda pode influenciar os resultados do estudo.

O desequilíbrio de classes em conjuntos de dados como o CAROSIA e o BBRC representa um desafio comum a muitos problemas de aprendizado de máquina do mundo real. Embora isso reflita cenários realistas, é importante considerar sua possível influência no desempenho do modelo em diferentes classes.

A escolha da métrica de avaliação a ser usada durante a busca em grade (F1 *score*) é apropriada para as tarefas em questão, especialmente devido à natureza desequilibrada de alguns conjuntos de dados. Entretanto, como acontece com qualquer métrica única, ela pode não captar todas as nuances do desempenho do modelo.

O desempenho superior consistente do DeB3RTa em todas as tarefas fornece fortes evidências de sua eficácia. Entretanto, como é padrão na pesquisa de aprendizado de máquina, é importante reconhecer que pequenas diferenças nas métricas de desempenho podem ser influenciadas por fatores como seleção aleatória de sementes e escolhas de hiperparâmetros. O uso pelo estudo de técnicas avançadas de *fine-tuning*, como LLRD e *Mixout*, contribui para sua metodologia robusta e, ao mesmo tempo, introduz otimizações específicas da tarefa que podem influenciar a generalização.

O processo de *fine-tuning*, incluindo a escolha de otimizadores e técnicas de programação, foi crucial para o sucesso do modelo. Embora esse nível de otimização seja um ponto forte do estudo, ele também apresenta um desafio comum na pesquisa de aprendizagem profunda: a sensibilidade dos resultados a configurações específicas de hiperparâmetros.

6.3 Trabalhos Futuros

Como trabalhos futuros, destaca-se a possibilidade de aplicar o modelo DeB3RTa a tarefas mais complexas e de maior impacto prático no setor financeiro, como a previsão de tendências e variações em índices de mercado, a exemplo do Ibovespa. Nesse contexto, o modelo poderia ser integrado a pipelines preditivos que combinem texto de notícias com séries temporais e outros indicadores estruturados, ampliando seu escopo de aplicação para análises quantitativas assistidas por linguagem natural. Além disso, propõe-se o desenvolvimento de mecanismos de monitoramento e registro do uso de recursos computacionais durante o treinamento e inferência dos modelos, incluindo consumo de CPU, GPU, memória e operações de entrada e saída, de modo a fomentar experimentações mais conscientes e alinhadas a boas práticas de eficiência energética e

sustentabilidade computacional. No plano técnico, sugere-se também a investigação de técnicas de compressão de modelos, como poda (*pruning*) [343] e quantização [344], para reduzir os custos computacionais sem perda significativa de precisão. Outra estratégia promissora seria a exploração de técnicas adicionais de aprendizagem por transferência, como o pré-treinamento contínuo, para refinar ainda mais a capacidade do DeB3RTa de lidar com textos financeiros especializados. Além disso, a ampliação da capacidade do modelo para suportar ambientes financeiros multilíngues beneficiaria muito as instituições que operam em mercados globais. Nota-se também a importância de comparar o modelo proposto com outros modelos mais recentemente desenvolvidos, bem como o uso de outros *datasets* relacionados ao domínio financeiro, sobretudo aplicáveis a outros tipos de tarefas, como *question-answering* e reconhecimento de entidades nomeadas.

6.4 Publicações

O trabalho desta tese deu origem a uma publicação em periódico, cujas informações seguem na Tabela 6.1.

Tabela 6.1 – Produção científica referente à tese.

Fator de Impacto	**Artigo**	**Tipo**	**Status**
3,7	PIRES, H. F. S.; PAUCAR, V. L.; CARVALHO, J. P. B. DeB3RTa: A Transformer-Based Model for the Portuguese Financial Domain. **Big Data and Cognitive Computing**, p. 1-30, Mar. 2025. DOI: 10.3390/bdcc9030051	Periódico	Publicado

6.5 Considerações Finais

Como visto ao longo desta pesquisa, a aplicação de modelos baseados em *Transformers*, em especial o DeB3RTa, no processamento de linguagem natural no contexto financeiro de língua portuguesa representa uma evolução significativa nas soluções de análise e interpretação de dados textuais especializados. Os resultados obtidos demonstraram que a abordagem adotada foi eficaz para lidar com a complexidade linguística e a especificidade do domínio financeiro, superando limitações anteriores em termos de adaptabilidade e precisão. A integração de técnicas avançadas de *fine-tuning* e a análise detalhada das métricas de desempenho destacaram o potencial do DeB3RTa em tarefas desafiadoras, como classificação de texto e análise de sentimentos, além de evidenciar a importância de uma metodologia robusta que combine rigor técnico e otimização de recursos computacionais. Os *insights* gerados por este trabalho não apenas corroboram a eficácia do modelo proposto,

mas também abrem portas para novas abordagens que poderão ampliar ainda mais a aplicabilidade do PLN em domínios financeiros especializados.

Entretanto, como em qualquer estudo, as limitações do trabalho também merecem ser destacadas, especialmente em relação ao tamanho dos conjuntos de dados utilizados e ao desequilíbrio de classes, fatores que podem afetar a generalização dos resultados. Embora o estudo tenha aplicado técnicas de regularização para mitigar o sobreajuste, questões relacionadas à representatividade dos dados e à sensibilidade a configurações de hiperparâmetros continuam sendo desafios a serem superados em investigações futuras. O caminho para o aprimoramento do modelo passa por explorar alternativas como a poda e a quantização de redes neurais, além de avaliar a eficácia de técnicas adicionais de aprendizagem por transferência, como o pré-treinamento contínuo. A ampliação da capacidade de suportar múltiplos idiomas, bem como a aplicação do modelo em outras tarefas complexas do PLN, como *question answering* e reconhecimento de entidades, são estratégias promissoras que poderão incrementar ainda mais a robustez e a versatilidade do modelo, expandindo seu impacto no campo das finanças.

Para facilitar a reprodutibilidade dos experimentos, foram disponibilizados o código-fonte e as instruções detalhadas em repositório público: https://github.com/higopires/DeB3RTa-Paper-Scripts. Além disso, os modelos treinados estão disponíveis publicamente por meio do *Hugging Face Model Hub* em https://huggingface.co/higopires/DeB3RTa-base e https://huggingface.co/higopires/DeB3RTa-small.

Bibliografia

[1] J. Devlin, M.-W. Chang, K. Lee e K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", em *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran e T. Solorio, ed., Minneapolis, Minnesota: Association for Computational Linguistics, jun. de 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. endereço: https://aclanthology.org/N19-1423.

[2] Ç. Aksoy, A. AhmetoĞlu e T. GÜngÖr, *Hierarchical Multitask Learning Approach for BERT*, 2020. arXiv: 2011.04451 [cs.CL]. endereço: https://arxiv.org/abs/2011.04451.

[3] Z. Zhang et al., *Semantics-aware BERT for Language Understanding*, 2020. arXiv: 1909.02209 [cs.CL]. endereço: https://arxiv.org/abs/1909.02209.

[4] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever et al., "Language models are unsupervised multitask learners", *OpenAI blog*, v. 1, n. 8, p. 9, 2019.

[5] T. Brown et al., "Language Models are Few-Shot Learners", em *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan e H. Lin, ed., vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901. endereço: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[6] OpenAI et al., *GPT-4 Technical Report*, 2024. arXiv: 2303.08774 [cs.CL]. endereço: https://arxiv.org/abs/2303.08774.

[7] D. Araci, *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*, 2019. arXiv: 1908.10063 [cs.CL]. endereço: https://arxiv.org/abs/1908.10063.

[8] Y. Yang, M. C. S. Uy e A. Huang, *FinBERT: A Pretrained Language Model for Financial Communications*, 2020. arXiv: 2006.08097 [cs.CL]. endereço: https://arxiv.org/abs/2006.08097.

[9] Z. Liu, D. Huang, K. Huang, Z. Li e J. Zhao, "FinBERT: a pre-trained financial language representation model for financial text mining", em *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, sér. IJCAI'20, Yokohama, Yokohama, Japan, 2021, ISBN: 9780999241165.

[10] L. Loukas et al., "FiNER: Financial Numeric Entity Recognition for XBRL Tagging", em *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov e A. Villavicencio, ed., Dublin, Ireland: Association for Computational Linguistics, mai. de 2022, pp. 4419–4431. DOI: 10.18653/v1/2022.acl-long.303. endereço: https://aclanthology.org/2022.acl-long.303.

[11] R. Shah et al., "When FLUE Meets FLANG: Benchmarks and Large Pretrained Language Model for Financial Domain", em *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva e Y. Zhang, ed., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, dez. de 2022, pp. 2322–2335. DOI: 10.18653/v1/2022.emnlp-main.148. endereço: https://aclanthology.org/2022.emnlp-main.148.

[12] D. Nguyen, N. Cao, S. Nguyen, S. Ta e C. Dinh, "MFinBERT: Multilingual Pretrained Language Model For Financial Domain", em *2022 14th International Conference on Knowledge and Systems Engineering (KSE)*, 2022, pp. 1–6. DOI: 10.1109/KSE56063.2022.9953749.

[13] J. Delgadillo, J. Kinyua e C. Mutigwe, "FinSoSent: Advancing Financial Market Sentiment Analysis through Pretrained Large Language Models", *Big Data and Cognitive Computing*, v. 8, n. 8, 2024, ISSN: 2504-2289. DOI: 10.3390/bdcc8080087. endereço: https://www.mdpi.com/2504-2289/8/8/87.

[14] Y. Cao, L. Yang, C. Wei e H. Wang, "Financial Text Sentiment Classification Based on Baichuan2 Instruction Finetuning Model", em *2023 5th International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, 2023, pp. 403–406. DOI: 10.1109/ICFTIC59930.2023.10454145.

[15] A. W. Lo, M. Singh e ChatGPT, "From ELIZA to ChatGPT: The evolution of natural language processing and financial applications", en, *J. Portf. Manag.*, v. 49, n. 7, pp. 201–235, jun. de 2023.

[16] P. R. Inserte, M. NakhlÉ, R. Qader, G. Caillaut e J. Liu, *Large Language Model Adaptation for Financial Sentiment Analysis*, 2024. arXiv: 2401.14777 [cs.CL]. endereço: https://arxiv.org/abs/2401.14777.

[17] A. Conneau et al., "Unsupervised Cross-lingual Representation Learning at Scale", em *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter e J. Tetreault, ed., Online: Association for Computational Linguistics, jul. de 2020, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. endereço: https://aclanthology.org/2020.acl-main.747.

[18] F. Souza, R. Nogueira e R. Lotufo, "BERT models for Brazilian Portuguese: Pretraining, evaluation and tokenization analysis", *Applied Soft Computing*, v. 149, p. 110 901, 2023, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2023.110901. endereço: https://www.sciencedirect.com/science/article/pii/S1568494623009195.

[19] S. Russell e P. Norvig, *Artificial intelligence*, en, 4ª ed. Upper Saddle River, NJ: Pearson, nov. de 2020.

[20] A. Ashta e H. Herrmann, "Artificial intelligence and fintech: an overview of opportunities and risks for banking, investments, and microfinance", *Strategic Change*, v. 30, pp. 211–222, 3 2021. DOI: 10.1002/jsc.2404.

[21] K. Frankish e W. M. Ramsey, ed., *The Cambridge handbook of artificial intelligence*. Cambridge, England: Cambridge University Press, jul. de 2014.

[22] A. Newell e H. Simon, "The logic theory machine–A complex information processing system", *IRE Transactions on Information Theory*, v. 2, n. 3, pp. 61–79, 1956. DOI: 10.1109/TIT.1956.1056797.

[23] A. Newell, J. C. Shaw e H. A. Simon, "Report on a general problem solving program", em *IFIP congress*, Pittsburgh, PA, vol. 256, 1959, p. 64.

[24] L. Gugerty, "Newell and Simon's Logic Theorist: Historical Background and Impact on Cognitive Modeling", *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, v. 50, n. 9, pp. 880–884, 2006. DOI: 10.1177/154193120605000904. eprint: https://doi.org/10.1177/154193120605000904. endereço: https://doi.org/10.1177/154193120605000904.

[25] R. D. Rupert, "Embodied Functionalism and Inner Complexity: Simon's Twenty-first Century Mind", em *Minds, Models and Milieux: Commemorating the Centennial of the Birth of Herbert Simon*, R. Frantz e L. Marsh, ed. London: Palgrave Macmillan UK, 2016, pp. 7–33, ISBN: 978-1-137-44250-5. DOI: 10.1057/9781137442505_2. endereço: https://doi.org/10.1057/9781137442505_2.

[26] J. Cao et al., "Artificial intelligence in gastroenterology and hepatology: status and challenges", *World Journal of Gastroenterology*, v. 27, pp. 1664–1690, 16 2021. DOI: 10.3748/wjg.v27.i16.1664.

[27] X. Ma, Y. Zhang e Y. Li, "Artificial intelligence applications in pediatric oncology diagnosis", *Exploration of Targeted Anti-Tumor Therapy*, pp. 157–169, 2023. DOI: 10.37349/etat.2023.00127.

[28] A. T. Rizvi, A. Haleem, S. Bahl e M. Javaid, "Artificial Intelligence (AI) and Its Applications in Indian Manufacturing: A Review", *Current Advances in Mechanical Engineering*, pp. 825–835, 2021. DOI: 10.1007/978-981-33-4795-3_76.

[29] B. Lal, V. S. N, M. A. Kumar, N. Chinthamu e S. Pokhriyal, "Development of Product Quality with Enhanced Productivity in Industry 4.0 with AI Driven Automation and Robotic Technology", *2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, pp. 184–189, 2023. DOI: 10.1109/ICAISS58487.2023.10250736.

[30] S. Wan, Z. Gu e Q. Ni, "Cognitive computing and wireless communications on the edge for healthcare service robots", *Comput. Commun.*, v. 149, pp. 99–106, 2020. DOI: 10.1016/j.comcom.2019.10.012.

[31] J. Holland et al., "Service Robots in the Healthcare Sector", *Robotics*, v. 10, p. 47, 2021. DOI: 10.3390/ROBOTICS10010047.

[32] M. B. Teli, M. S. Totad e M. S. Desai, "Use of AI (Artificial Intelligence) in Robotics", *International Journal for Research in Applied Science and Engineering Technology*, 2023. DOI: 10.22214/ijraset.2023.55235.

[33] N. Sousa, N. Oliveira e I. PraÇa, "A Multi-Agent System for Autonomous Mobile Robot Coordination", *ArXiv*, v. abs/2109.12386, 2021.

[34] S. Bhattamisra, P. Banerjee, P. Gupta, J. Mayuren, S. Patra e M. Candasamy, "Artificial Intelligence in Pharmaceutical and Healthcare Research", *Big Data Cogn. Comput.*, v. 7, p. 10, 2023. DOI: 10.3390/bdcc7010010.

[35] V. Mehta, "Artificial Intelligence in Medicine: Revolutionizing Healthcare for Improved Patient Outcomes", *Journal of Medical Research and Innovation*, 2023. DOI: 10.32892/jmri.292.

[36] W. Jiao, X. Zhang e F. D'Souza, "The Economic Value and Clinical Impact of Artificial Intelligence in Healthcare: A Scoping Literature Review", *IEEE Access*, v. 11, pp. 123 445–123 457, 2023. DOI: 10.1109/ACCESS.2023.3327905.

[37] A. Seyhan e C. Carini, "Are innovation and new technologies in precision medicine paving a new era in patients centric care?", *Journal of Translational Medicine*, v. 17, 2019. DOI: 10.1186/s12967-019-1864-9.

[38] K. B. Johnson et al., "Precision Medicine, AI, and the Future of Personalized Health Care", *Clinical and Translational Science*, v. 14, pp. 86–93, 2020. DOI: 10.1111/cts.12884.

[39] S. Rezayi, S. R. N. Kalhori e S. Saeedi, "Effectiveness of Artificial Intelligence for Personalized Medicine in Neoplasms: A Systematic Review", *BioMed Research International*, v. 2022, 2022. DOI: 10.1155/2022/7842566.

[40] S. G. Fritz-Morgenthal, B. Hein e J. Papenbrock, "Financial Risk Management and Explainable, Trustworthy, Responsible AI", *Frontiers in Artificial Intelligence*, v. 5, 2021. DOI: 10.3389/frai.2022.779799.

[41] L. Al-Blooshi e H. Nobanee, "Applications of Artificial Intelligence in Financial Management Decisions: A Mini-Review", *Consumer Financial Fraud eJournal*, 2020. DOI: 10.2139/ssrn.3540140.

[42] Y. Han, J. Chen, M. Dou, J. Wang e K. Feng, "The Impact of Artificial Intelligence on the Financial Services Industry", *Academic Journal of Management and Social Sciences*, 2023. DOI: 10.54097/ajmss.v2i3.8741.

[43] F. Ahmed, "ETHICAL ASPECTS OF ARTIFICIAL INTELLIGENCE IN BANKING", *Journal of Research in Economics and Finance Management*, 2022. DOI: 10.56596/jrefm.v1i2.7.

[44] H. Aldboush e M. Ferdous, "Building Trust in Fintech: An Analysis of Ethical and Privacy Considerations in the Intersection of Big Data, AI, and Customer Trust", *International Journal of Financial Studies*, 2023. DOI: 10.3390/ijfs11030090.

[45] A. Singh e N. Ahlawat, "A review article: -the Growing Role of Data Science and AI in Banking and Finance", *International Research Journal of Modernization in Engineering Technology and Science*, 2023. DOI: 10.56726/irjmets44000.

[46] J. Lee, "Access to Finance for Artificial Intelligence Regulation in the Financial Services Industry", *European Business Organization Law Review*, v. 21, pp. 731–757, 2019. DOI: 10.1007/s40804-020-00200-0.

[47] C. Maple et al., "The AI Revolution: Opportunities and Challenges for the Finance Sector", *ArXiv*, v. abs/2308.16538, 2023. DOI: 10.48550/arXiv.2308.16538.

[48] A. Akavova, Z. Temirkhanova e Z. Lorsanova, "Adaptive learning and artificial intelligence in the educational space", *E3S Web of Conferences*, 2023. DOI: 10.1051/e3sconf/202345106011.

[49] M. Chhatwal, V. Garg e N. Rajput, "Role of AI in the Education Sector", *Lloyd Business Review*, 2023. DOI: 10.56595/lbr.v2i1.11.

[50] A. Harry, "Role of AI in Education", *Interdiciplinary Journal and Hummanity (INJURITY)*, 2023. DOI: 10.58631/injurity.v2i3.52.

[51] C. Fang e A. Tse, "Case Study: Postgraduate Students' Class Engagement in Various Online Learning Contexts When Taking Privacy Issues to Incorporate with Artificial Intelligence Applications", *International Journal of Learning and Teaching*, 2023. DOI: 10.18178/ijlt.9.2.90-95.

[52] Y. Ma, Z. Wang, H. Yang e L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: a survey", *IEEE/CAA Journal of Automatica Sinica*, v. 7, pp. 315–329, 2020. DOI: 10.1109/JAS.2020.1003021.

[53] T. Zhang, T. Zhao, Y. Qin e S. Liu, "Artificial intelligence in intelligent vehicles: recent advances and future directions", *Journal of the Chinese Institute of Engineers*, v. 46, pp. 905–911, 2023. DOI: 10.1080/02533839.2023.2262759.

[54] L. S. Iyer, "AI enabled applications towards intelligent transportation", *Transportation Engineering*, v. 5, p. 100 083, 2021, ISSN: 2666-691X. DOI: https://doi.org/10.1016/j.treng.2021.100083. endereço: https://www.sciencedirect.com/science/article/pii/S2666691X21000397.

[55] S. Khan, A. Adnan e N. Iqbal, "Applications of Artificial Intelligence in Transportation", *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pp. 1–6, 2022. DOI: 10.1109/ICECET55527.2022.9872928.

[56] S. Chavhan et al., "Edge Computing AI-IoT Integrated Energy-efficient Intelligent Transportation System for Smart Cities", *ACM Transactions on Internet Technology*, v. 22, pp. 1–18, 2022. DOI: 10.1145/3507906.

[57] F. Yuli, Y. Jin, Y. Shi, Y. Wang, K. Wang e L. Yuqiing, "The Application and Challenges of Artificial Intelligence in the Transportation Field", *Cambridge Explorations in Arts and Sciences*, 2023. DOI: 10.61603/ceas.v1i2.21.

[58] W. Tong, A. Hussain, W. Bo e S. Maharjan, "Artificial Intelligence for Vehicle-to-Everything: A Survey", *IEEE Access*, v. 7, pp. 10 823–10 843, 2019. DOI: 10.1109/ACCESS.2019.2891073.

[59] J. Na, H. Karimi, C. Hu, Y. Qin e H. Du, "Guest Editorial: AI Applications to Intelligent Vehicles for Advancing Intelligent Transport Systems", *IET Intelligent Transport Systems*, 2020. DOI: 10.1049/iet-its.2020.0189.

[60] J. Burstein, "Opportunities for Natural Language Processing Research in Education", em *Computational Linguistics and Intelligent Text Processing*, A. Gelbukh, ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 6–27, ISBN: 978-3-642-00382-0.

[61] M. Mashaabi, A. Alotaibi, H. Qudaih, R. Alnashwan e H. Al-Khalifa, "Natural Language Processing in Customer Service: A Systematic Review", *ArXiv*, v. abs/2212.09523, 2022. DOI: 10.48550/arXiv.2212.09523.

[62] K. Juglan, B. Sharma, A. Gehlot, S. P. Singh, A. Hussein e M. Alazzam, "Exploring the Effectiveness of Natural Language Processing in Customer Service", *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 814–818, 2023. DOI: 10.1109/ICACITE57410.2023.10183107.

[63] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi e G. Neubig, "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing", *ACM Comput. Surv.*, v. 55, n. 9, jan. de 2023, ISSN: 0360-0300. DOI: 10.1145/3560815. endereço: https://doi.org/10.1145/3560815.

[64] X. Ze, "Research on deep learning in natural language processing", *Advances in Computer and Communication*, v. 4, pp. 196–200, 3 2023. DOI: 10.26855/acc.2023.06.018.

[65] C. P. Chai, "Comparison of text preprocessing methods", en, *Nat. Lang. Eng.*, v. 29, n. 3, pp. 509–553, mai. de 2023.

[66] M. F. Porter, "An algorithm for suffix stripping", en, *Program*, v. 14, n. 3, pp. 130–137, mar. de 1980.

[67] V. Balakrishnan e L.-Y. Ethel, "Stemming and lemmatization: A comparison of retrieval performances", *Lect. Notes Softw. Eng.*, v. 2, n. 3, pp. 262–267, 2014.

[68] D. J. Ladani e N. P. Desai, "Stopword Identification and Removal Techniques on TC and IR applications: A Survey", em *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020, pp. 466–472. DOI: 10.1109/ICACCS48705.2020.9074166.

[69] V. Rolim, R. Ferreira e E. Costa, "Utilização de Técnicas de Aprendizado de Máquina para Acompanhamento de Fóruns Educacionais", *Rev. Bras. Inform. Na Educ.*, v. 25, n. 03, p. 112, out. de 2017.

[70] V. A. Kozhevnikov, Peter the Great St. Petersburg Polytechnic University, E. S. Pankratova e Peter the Great St. Petersburg Polytechnic University, "Research of text pre-processing methods for preparing data in Russian for machine learning", *Theor. Appl. Sci.*, v. 84, n. 04, pp. 313–320, abr. de 2020.

[71] K. Juluru, H.-H. Shih, K. N. Keshava Murthy e P. Elnajjar, "Bag-of-words technique in natural language processing: A primer for radiologists", en, *Radiographics*, v. 41, n. 5, pp. 1420–1426, set. de 2021.

[72] J. Camacho-Collados e M. T. Pilehvar, "On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis", em *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, T. Linzen, G. Chrupała e A. Alishahi, ed., Brussels, Belgium: Association for Computational Linguistics, nov. de 2018, pp. 40–46. DOI: 10.18653/v1/W18-5406. endereço: https://aclanthology.org/W18-5406.

[73] A. ÇayIr, I. YenidoĞan e H. DaĞ, "Feature Extraction Based on Deep Learning for Some Traditional Machine Learning Methods", *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pp. 494–497, 2018. DOI: 10.1109/UBMK.2018.8566383.

[74] C. Hung, E. Song e Y. Lan, "Foundation of Deep Machine Learning in Neural Networks", *Image Texture Analysis*, 2019. DOI: 10.1007/978-3-030-13773-1_9.

[75] D. Wang, J. Su e H. Yu, "Feature Extraction and Analysis of Natural Language Processing for Deep Learning English Language", *IEEE Access*, v. 8, pp. 46 335–46 345, 2020. DOI: 10.1109/ACCESS.2020.2974101.

[76] C. Affonso, A. L. Rossi, F. H. A. Vieira e A. Carvalho, "Deep learning for biological image classification", *Expert Syst. Appl.*, v. 85, pp. 114–122, 2017. DOI: 10.1016/J.ESWA.2017.05.039.

[77] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi e J. Benediktsson, "Deep Learning for Hyperspectral Image Classification: An Overview", *IEEE Transactions on Geoscience and Remote Sensing*, v. 57, pp. 6690–6709, 2019. DOI: 10.1109/TGRS.2019.2907932.

[78] S. Purushotham, C. Meng, Z. Che e Y. Liu, "Benchmarking deep learning models on large healthcare datasets", *Journal of biomedical informatics*, v. 83, pp. 112–134, 2018. DOI: 10.1016/j.jbi.2018.04.007.

[79] Q. Zhang, L. Yang, Z. Chen e P. Li, "A survey on deep learning for big data", *Inf. Fusion*, v. 42, pp. 146–157, 2018. DOI: 10.1016/j.inffus.2017.10.006.

[80] C. Sun, A. Shrivastava, S. Singh e A. Gupta, "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era", *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 843–852, 2017. DOI: 10.1109/ICCV.2017.97.

[81] M. Capra, B. Bussolino, A. Marchisio, G. Masera, M. Martina e M. Shafique, "Hardware and Software Optimizations for Accelerating Deep Neural Networks: Survey of Current Trends, Challenges, and the Road Ahead", *IEEE Access*, v. 8, pp. 225 134–225 180, 2020. DOI: 10.1109/ACCESS.2020.3039858.

[82] N. C. Thompson, K. H. Greenewald, K. Lee e G. F. Manso, "The Computational Limits of Deep Learning", *ArXiv*, v. abs/2007.05558, 2020. DOI: 10.21428/bf6fb269.1f033948.

[83] P. Ren et al., "A Survey of Deep Active Learning", *ACM Computing Surveys (CSUR)*, v. 54, pp. 1–40, 2020. DOI: 10.1145/3472291.

[84] Y. LeCun et al., "Handwritten Digit Recognition with a Back-Propagation Network", em *Advances in Neural Information Processing Systems*, D. Touretzky, ed., vol. 2, Morgan-Kaufmann, 1989. endereço: https://proceedings.neurips.cc/paper_files/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf.

[85] R. Yamashita, M. Nishio, R. K. G. Do e K. Togashi, "Convolutional neural networks: an overview and application in radiology", en, *Insights Imaging*, v. 9, n. 4, pp. 611–629, ago. de 2018.

[86] A. Derry, M. Krzywinski e N. Altman, "Convolutional neural networks", *Nature Methods*, v. 20, pp. 1269–1270, 2023. DOI: 10.1038/s41592-023-01973-1.

[87] J. Gu et al., "Recent advances in convolutional neural networks", en, *Pattern Recognit.*, v. 77, pp. 354–377, mai. de 2018.

[88] Z. Li, F. Liu, W. Yang, S. Peng e J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects", *IEEE Transactions on Neural Networks and Learning Systems*, v. 33, n. 12, pp. 6999–7019, 2022. DOI: 10.1109/TNNLS.2021.3084827.

[89] M. Ferreira, P. Portella, J. Souza, B. Dias, L. Assunção e L. Oliveira, "Avaliação do Uso Redes Neurais Convolucionais para Identificação de Lesões Cariosas Dentárias", em *Anais do XXIII Simpósio Brasileiro de Computação Aplicada à Saúde*, São Paulo/SP: SBC, 2023, pp. 473–478. DOI: 10.5753/sbcas.2023.229626. endereço: https://sol.sbc.org.br/index.php/sbcas/article/view/25313.

[90] S. Hochreiter e J. Schmidhuber, "Long short-term memory", en, *Neural Comput.*, v. 9, n. 8, pp. 1735–1780, nov. de 1997.

[91] K. Cho et al., "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation", em *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang e W. Daelemans, ed., Doha, Qatar: Association for Computational Linguistics, out. de 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179. endereço: https://aclanthology.org/D14-1179.

[92] Y. Yu, X. Si, C. Hu e J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures", *Neural Computation*, v. 31, n. 7, pp. 1235–1270, 2019. DOI: 10.1162/neco_a_01199.

[93] S. A. Marhon, C. J. F. Cameron e S. C. Kremer, "Recurrent Neural Networks", em *Intelligent Systems Reference Library*, sér. Intelligent systems reference library, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 29–65.

[94] V. S. Lalapura, J. Amudha e H. S. Satheesh, "Recurrent Neural Networks for Edge Intelligence: A Survey", *ACM Comput. Surv.*, v. 54, n. 4, mai. de 2021, ISSN: 0360-0300. DOI: 10.1145/3448974. endereço: https://doi.org/10.1145/3448974.

[95] O. Barak, "Recurrent neural networks as versatile tools of neuroscience research", *Curr. Opin. Neurobiol.*, v. 46, pp. 1–6, out. de 2017.

[96] G. Barbosa, G. M. Bezerra, D. S. de Medeiros, M. Andreoni Lopez e D. Mattos, "Segurança em Redes 5G: Oportunidades e Desafios em Detecção de Anomalias e Predição de Tráfego Baseadas em Aprendizado de Máquina", em *Minicursos do XXI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, SBC, out. de 2021, pp. 145–189.

[97] J. Zhou, J. Ye, Y. Ouyang, M. Tong, X. Pan e J. Gao, "On Building Real Time Intelligent Agricultural Commodity Trading Models", em *2022 IEEE Eighth International Conference on Big Data Computing Service and Applications (BigDataService)*, 2022, pp. 89–95. DOI: 10.1109/BigDataService55688.2022.00021.

[98] G. E. Hinton, S. Osindero e Y.-W. Teh, "A fast learning algorithm for deep belief nets", en, *Neural Comput.*, v. 18, n. 7, pp. 1527–1554, jul. de 2006.

[99] N. Le Roux e Y. Bengio, "Representational Power of Restricted Boltzmann Machines and Deep Belief Networks", *Neural Computation*, v. 20, n. 6, pp. 1631–1649, 2008. DOI: 10.1162/neco.2008.04-07-510.

[100] M. Zambra, A. Testolin e M. Zorzi, "A developmental approach for training deep belief networks", en, *Cognit. Comput.*, v. 15, n. 1, pp. 103–120, jan. de 2023.

[101] R. Sarikaya, G. E. Hinton e A. Deoras, "Application of Deep Belief Networks for Natural Language Understanding", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, v. 22, n. 4, pp. 778–784, 2014. DOI: 10.1109/TASLP.2014.2303296.

[102] S. N. Tran e A. S. d'Avila Garcez, "Deep Logic Networks: Inserting and Extracting Knowledge From Deep Belief Networks", *IEEE Transactions on Neural Networks and Learning Systems*, v. 29, n. 2, pp. 246–258, 2018. DOI: 10.1109/TNNLS.2016.2603784.

[103] Y. Hua, J. Guo e H. Zhao, "Deep Belief Networks and deep learning", em *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things*, 2015, pp. 1–4. DOI: 10.1109/ICAIOT.2015.7111524.

[104] A. Elhassouny e F. Smarandache, "Trends in deep convolutional neural Networks architectures: a review", em *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, 2019, pp. 1–8. DOI: 10.1109/ICCSRE.2019.8807741.

[105] P. Baldi e Z. Lu, "Complex-valued autoencoders", *Neural Networks*, v. 33, pp. 136–147, set. de 2012, ISSN: 0893-6080. DOI: 10.1016/j.neunet.2012.04.011. endereço: http://dx.doi.org/10.1016/j.neunet.2012.04.011.

[106] R. Al-Hmouz, W. Pedrycz, A. Balamash e A. Morfeq, "Logic-driven autoencoders", en, *Knowl. Based Syst.*, v. 183, n. 104874, p. 104 874, nov. de 2019.

[107] J. Romero, J. P. Olson e A. Aspuru-Guzik, "Quantum autoencoders for efficient compression of quantum data", *Quantum Sci. Technol.*, v. 2, n. 4, p. 045 001, dez. de 2017.

[108] Z. Wang, K. Li, S. Q. Xia e H. Liu, "Economic Recession Prediction Using Deep Neural Network", arXiv.org, Papers 2107.10980, jul. de 2021. endereço: https://ideas.repec.org/p/arx/papers/2107.10980.html.

[109] C. do Nascimento, V. Garcia e R. Araujo, "A Word Sense Disambiguation Method Applied to Natural Language Processing for the Portuguese Language", *IEEE Open Journal of the Computer Society*, v. 5, n. 01, pp. 268–277, jan. de 2024, ISSN: 2644-1268. DOI: 10.1109/OJCS.2024.3396518.

[110] N. N. Nunes, "Lexical and semantic variation in contemporary spoken Portuguese in urban Funchal and rural areas of Madeira Island", *J. Port. Linguist.*, v. 20, n. 1, p. 4, abr. de 2021.

[111] L. F. de Lima e R. de Araujo, "A call for a research agenda on fair NLP for Portuguese", em *Anais do XIV Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, Belo Horizonte/MG: SBC, 2023, pp. 187–192. DOI: 10.5753/stil.2023.233763. endereço: https://sol.sbc.org.br/index.php/stil/article/view/25450.

[112] J. da Rocha Junqueira et al., "BERTimbau in Action: An Investigation of its Abilities in Sentiment Analysis, Aspect Extraction, Hate Speech Detection, and Irony Detection.", em *The International FLAIRS Conference Proceedings*, vol. 36, 2023.

[113] R. de Oliveira, B. Menezes, J. Ortiz e E. Nascimento, "Automatic BI-RADS Classification of Breast Magnetic Resonance Medical Records Using Transformer-Based Models for Brazilian Portuguese", em *Machine Learning and Data Mining Annual Volume 2023*, M. A. Aceves-Fernández, ed., Rijeka: IntechOpen, 2023, cap. 6. DOI: 10.5772/intechopen.113886. endereço: https://doi.org/10.5772/intechopen.113886.

[114] J. A. Wagner Filho, R. Wilkens, M. Idiart e A. Villavicencio, "The brWaC Corpus: A New Open Resource for Brazilian Portuguese", em *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, N. Calzolari et al., ed., Miyazaki, Japan: European Language Resources Association (ELRA), mai. de 2018. endereço: https://aclanthology.org/L18-1686.

[115] E. Fonseca, L. Santos, M. Criscuolo e S. Aluisio, "ASSIN: Avaliacao de similaridade semantica e inferencia textual", em *Computational Processing of the Portuguese Language-12th International Conference, Tomar, Portugal*, 2016, pp. 13–15.

[116] L. Real, E. Fonseca e H. Gonçalo Oliveira, "The ASSIN 2 Shared Task: A Quick Overview", em *Computational Processing of the Portuguese Language*, P. Quaresma, R. Vieira, S. Aluísio, H. Moniz, F. Batista e T. Gonçalves, ed., Cham: Springer International Publishing, 2020, pp. 406–412, ISBN: 978-3-030-41505-1.

[117] D. Carmo, M. Piau, I. Campiotti, R. Nogueira e R. Lotufo, *PTT5: Pretraining and validating the T5 model on Brazilian Portuguese data*, 2020. arXiv: 2008.09144 [cs.CL]. endereço: https://arxiv.org/abs/2008.09144.

[118] P. Fialho, L. Coheur e P. Quaresma, "Benchmarking Natural Language Inference and Semantic Textual Similarity for Portuguese", *Information*, v. 11, n. 10, 2020, ISSN: 2078-2489. DOI: 10.3390/info11100484. endereço: https://www.mdpi.com/2078-2489/11/10/484.

[119] J. E. Andrade Junior, J. Cardoso-Silva e L. C. T. Bezerra, "Comparing Contextual Embeddings for Semantic Textual Similarity in Portuguese", em *Intelligent Systems*, A. Britto e K. Valdivia Delgado, ed., Cham: Springer International Publishing, 2021, pp. 389–404, ISBN: 978-3-030-91699-2.

[120] T. Mikolov, K. Chen, G. Corrado e J. Dean, *Efficient Estimation of Word Representations in Vector Space*, 2013. arXiv: 1301.3781 [cs.CL]. endereço: https://arxiv.org/abs/1301.3781.

[121] J. Pennington, R. Socher e C. Manning, "GloVe: Global Vectors for Word Representation", em *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang e W. Daelemans, ed., Doha, Qatar: Association for Computational Linguistics, out. de 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. endereço: https://aclanthology.org/D14-1162/.

[122] P. Bojanowski, E. Grave, A. Joulin e T. Mikolov, "Enriching Word Vectors with Subword Information", *Transactions of the Association for Computational Linguistics*, v. 5, L. Lee, M. Johnson e K. Toutanova, ed., pp. 135–146, 2017. DOI: 10.1162/tacl_a_00051. endereço: https://aclanthology.org/Q17-1010/.

[123] N. Lazzari, A. Poltronieri e V. Presutti, *Pitchclass2vec: Symbolic Music Structure Segmentation with Chord Embeddings*, 2023. arXiv: 2303.15306 [cs.SD]. endereço: https://arxiv.org/abs/2303.15306.

[124] M. Antoniak e D. Mimno, "Evaluating the Stability of Embedding-based Word Similarities", *Transactions of the Association for Computational Linguistics*, v. 6, L. Lee, M. Johnson, K. Toutanova e B. Roark, ed., pp. 107–119, 2018. DOI: 10.1162/tacl_a_00008. endereço: https://aclanthology.org/Q18-1008/.

[125] E. Chersoni, E. Santus, C.-R. Huang e A. Lenci, "Decoding Word Embeddings with Brain-Based Semantic Features", *Computational Linguistics*, v. 47, n. 3, pp. 663–698, nov. de 2021. DOI: 10.1162/coli_a_00412. endereço: https://aclanthology.org/2021.cl-3.20/.

[126] O. Papakyriakopoulos, S. Hegelich, J. C. M. Serrano e F. Marco, "Bias in word embeddings", em *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, sér. FAT* '20, Barcelona, Spain: Association for Computing Machinery, 2020, pp. 446–457, ISBN: 9781450369367. DOI: 10.1145/3351095.3372843. endereço: https://doi.org/10.1145/3351095.3372843.

[127] L. Wendlandt, J. K. Kummerfeld e R. Mihalcea, "Factors Influencing the Surprising Instability of Word Embeddings", em *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, M. Walker, H. Ji e A. Stent, ed., New Orleans, Louisiana: Association for Computational Linguistics, jun. de 2018, pp. 2092–2102. DOI: 10.18653/v1/N18-1190. endereço: https://aclanthology.org/N18-1190/.

[128] A. Silva e C. Amarathunga, "On Learning Word Embeddings From Linguistically Augmented Text Corpora", em *Proceedings of the 13th International Conference on Computational Semantics - Short Papers*, S. Dobnik, S. Chatzikyriakidis e V. Demberg, ed., Gothenburg, Sweden: Association for Computational Linguistics, mai. de 2019, pp. 52–58. DOI: 10.18653/v1/W19-0508. endereço: https://aclanthology.org/W19-0508/.

[129] A. Vaswani et al., *Attention Is All You Need*, 2023. arXiv: 1706.03762 [cs.CL]. endereço: https://arxiv.org/abs/1706.03762.

[130] G. Qin, Y. Feng e B. Van Durme, "The NLP Task Effectiveness of Long-Range Transformers", em *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, A. Vlachos e I. Augenstein, ed., Dubrovnik, Croatia: Association for Computational Linguistics, mai. de 2023, pp. 3774–3790. DOI: 10.18653/v1/2023.eacl-main.273. endereço: https://aclanthology.org/2023.eacl-main.273.

[131] S. Islam et al., "A comprehensive survey on applications of transformers for deep learning tasks", en, *Expert Syst. Appl.*, v. 241, n. 122666, p. 122 666, mai. de 2024.

[132] B. Peng, Y. Ding e W. Kang, "Metaformer: A Transformer That Tends to Mine Metaphorical-Level Information", *Sensors*, v. 23, n. 11, 2023, ISSN: 1424-8220. DOI: 10.3390/s23115093. endereço: https://www.mdpi.com/1424-8220/23/11/5093.

[133] K. Murray, J. Kinnison, T. Q. Nguyen, W. Scheirer e D. Chiang, "Auto-Sizing the Transformer Network: Improving Speed, Efficiency, and Performance for Low-Resource Machine Translation", em *Proceedings of the 3rd Workshop on Neural Generation and Translation*, A. Birch et al., ed., Hong Kong: Association for Computational Linguistics, nov. de 2019, pp. 231–240. DOI: 10.18653/v1/D19-5625. endereço: https://aclanthology.org/D19-5625/.

[134] K. He, X. Zhang, S. Ren e J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", em *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123.

[135] S. Ruder, M. E. Peters, S. Swayamdipta e T. Wolf, "Transfer Learning in Natural Language Processing", em *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, A. Sarkar e M. Strube, ed., Minneapolis, Minnesota: Association for Computational Linguistics, jun. de 2019, pp. 15–18. DOI: 10.18653/v1/N19-5004. endereço: https://aclanthology.org/N19-5004.

[136] Z. Alyafeai, M. S. AlShaibani e I. Ahmad, *A Survey on Transfer Learning in Natural Language Processing*, 2020. arXiv: 2007.04239 [cs.CL]. endereço: https://arxiv.org/abs/2007.04239.

[137] J. Wang e Y. Chen, "Transfer learning for natural language processing", en, em *Machine Learning: Foundations, Methodologies, and Applications*, Singapore: Springer Nature Singapore, 2023, pp. 275–279.

[138] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li e L. Fei-Fei, "ImageNet: A large-scale hierarchical image database", em *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[139] L. Vogado, R. Veras e K. Aires, ""LeukNet" - Um Modelo de Rede Neural Convolucional para o Diagnóstico de Leucemia", em *Anais Estendidos do XXI Simpósio Brasileiro de Computação Aplicada à Saúde*, Evento Online: SBC, 2021, pp. 85–90. DOI: 10.5753/sbcas.2021.16106. endereço: https://sol.sbc.org.br/index.php/sbcas_estendido/article/view/16106.

[140] Z. Zhu, K. Lin, A. K. Jain e J. Zhou, "Transfer Learning in Deep Reinforcement Learning: A Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 45, pp. 13 344–13 362, 2020. DOI: 10.1109/TPAMI.2023.3292075.

[141] J. Nance e P. Baumgartner, "gobbli: A uniform interface to deep learning for text in Python", *J. Open Source Softw.*, v. 6, n. 62, p. 2395, jun. de 2021.

[142] S. Meftah, N. Semmar, Y. Tamaazousti, H. Essafi e F. Sadat, *Neural Supervised Domain Adaptation by Augmenting Pre-trained Models with Random Units*, 2021. arXiv: 2106.04935 [cs.CL]. endereço: https://arxiv.org/abs/2106.04935.

[143] W. Zhang, L. Deng, L. Zhang e D. Wu, "A Survey on Negative Transfer", *IEEE/CAA Journal of Automatica Sinica*, v. 10, n. 2, pp. 305–329, 2023. DOI: 10.1109/JAS.2022.106004.

[144] E. Strubell, A. Ganesh e A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP", em *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum e L. Màrquez, ed., Florence, Italy: Association for Computational Linguistics, jul. de 2019, pp. 3645–3650. DOI: 10.18653/v1/P19-1355. endereço: https://aclanthology.org/P19-1355.

[145] V. Sanh, L. Debut, J. Chaumond e T. Wolf, *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*, 2020. arXiv: 1910.01108 [cs.CL]. endereço: https://arxiv.org/abs/1910.01108.

[146] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick e G. Neubig, *Towards a Unified View of Parameter-Efficient Transfer Learning*, 2022. arXiv: 2110.04366 [cs.CL]. endereço: https://arxiv.org/abs/2110.04366.

[147] E. Laparra, A. Mascio, S. Velupillai e T. Miller, "A review of recent work in transfer learning and domain adaptation for natural language processing of electronic health records", *Yearbook of medical informatics*, v. 30, n. 01, pp. 239–244, 2021.

[148] W. Wang, V. W. Zheng, H. Yu e C. Miao, "A Survey of Zero-Shot Learning: Settings, Methods, and Applications", *ACM Trans. Intell. Syst. Technol.*, v. 10, n. 2, jan. de 2019, ISSN: 2157-6904. DOI: 10.1145/3293318. endereço: https://doi.org/10.1145/3293318.

[149] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr e T. M. Hospedales, "Learning to Compare: Relation Network for Few-Shot Learning", em *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208. DOI: 10.1109/CVPR.2018.00131.

[150] Y. Song, T. Wang, P. Cai, S. K. Mondal e J. P. Sahoo, "A Comprehensive Survey of Few-shot Learning: Evolution, Applications, Challenges, and Opportunities", *ACM Comput. Surv.*, v. 55, n. 13s, jul. de 2023, ISSN: 0360-0300. DOI: 10.1145/3582688. endereço: https://doi.org/10.1145/3582688.

[151] A. T. Oyewole, O. B. Adeoye, W. A. Addy, C. C. Okoye, O. C. Ofodile e C. E. Ugochukwu, "Automating financial reporting with natural language processing: A review and case analysis", *World J. Adv. Res. Rev.*, v. 21, n. 3, pp. 575–589, mar. de 2024.

[152] H. Kang, "Text mining and financial analysis modeling for financial statement disclosure", *J. Electr. Syst.*, v. 20, n. 6s, pp. 2230–2240, abr. de 2024.

[153] P. Jawale, S. Jawale, D. Ingale e M. Shetty, "Sentiment analysis for financial markets", *Int. J. Res. Appl. Sci. Eng. Technol.*, v. 11, n. 12, pp. 535–541, dez. de 2023.

[154] F. Z. Xing, E. Cambria e R. E. Welsch, "Natural language based financial forecasting: a survey", en, *Artif. Intell. Rev.*, v. 50, n. 1, pp. 49–73, jun. de 2018.

[155] E. Clarkson, W. Nasir e N. Ford, "FAST-SCAN: Forward-looking Analysis System for Stock Trend Anticipation", *Preprints*, abr. de 2024. DOI: 10.20944/preprints202404.0780.v1. endereço: https://doi.org/10.20944/preprints202404.0780.v1.

[156] R. Sawhney, A. Aggarwal e R. R. Shah, "An Empirical Investigation of Bias in the Multimodal Analysis of Financial Earnings Calls", em *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova et al., ed., Online: Association for Computational Linguistics, jun. de 2021, pp. 3751–3757. DOI: 10.18653/v1/2021.naacl-main.294. endereço: https://aclanthology.org/2021.naacl-main.294.

[157] R. Jørgensen, O. Brandt, M. Hartmann, X. Dai, C. Igel e D. Elliott, "MultiFin: A Dataset for Multilingual Financial NLP", em *Findings of the Association for Computational Linguistics: EACL 2023*, A. Vlachos e I. Augenstein, ed., Dubrovnik, Croatia: Association for Computational Linguistics, mai. de 2023, pp. 894–909. DOI: 10.18653/v1/2023.findings-eacl.66. endereço: https://aclanthology.org/2023.findings-eacl.66.

[158] J. Lin, R. Nogueira e A. Yates, *Pretrained Transformers for Text Ranking: BERT and Beyond*, 2021. arXiv: 2010.06467 [cs.IR]. endereço: https://arxiv.org/abs/2010.06467.

[159] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai e X. Huang, "Pre-trained models for natural language processing: A survey", en, *Sci. China Technol. Sci.*, v. 63, n. 10, pp. 1872–1897, out. de 2020.

[160] B. Min et al., "Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey", *ACM Comput. Surv.*, v. 56, n. 2, set. de 2023, ISSN: 0360-0300. DOI: 10.1145/3605943. endereço: https://doi.org/10.1145/3605943.

[161] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer", *J. Mach. Learn. Res.*, v. 21, n. 1, jan. de 2020, ISSN: 1532-4435.

[162] M. Mars, "From Word Embeddings to Pre-Trained Language Models: A State-of-the-Art Walkthrough", *Applied Sciences*, v. 12, n. 17, 2022, ISSN: 2076-3417. DOI: 10.3390/app12178805. endereço: https://www.mdpi.com/2076-3417/12/17/8805.

[163] Y. Li, J. Li, Y. Suhara, A. Doan e W.-C. Tan, "Deep entity matching with pre-trained language models", *Proc. VLDB Endow.*, v. 14, n. 1, pp. 50–60, set. de 2020, ISSN: 2150-8097. DOI: 10.14778/3421424.3421431. endereço: https://doi.org/10.14778/3421424.3421431.

[164] P. Izsak, M. Berchansky e O. Levy, "How to Train BERT with an Academic Budget", em *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia e S. W.-t. Yih, ed., Online e Punta Cana, Dominican Republic: Association for Computational Linguistics, nov. de

2021, pp. 10 644–10 652. DOI: 10.18653/v1/2021.emnlp-main.831. endereço: https://aclanthology.org/2021.emnlp-main.831.

[165] A. Lauscher, T. Lueken e G. Glavaš, "Sustainable Modular Debiasing of Language Models", em *Findings of the Association for Computational Linguistics: EMNLP 2021*, M.-F. Moens, X. Huang, L. Specia e S. W.-t. Yih, ed., Punta Cana, Dominican Republic: Association for Computational Linguistics, nov. de 2021, pp. 4782–4797. DOI: 10.18653/v1/2021.findings-emnlp.411. endereço: https://aclanthology.org/2021.findings-emnlp.411.

[166] T. Bandyopadhyay, S. Saha e D. Pal, "Beyond imitation: Exploring novelty in generative AI", en, *International Journal of Advanced Research in Science, Communication and Technology*, pp. 472–475, set. de 2023.

[167] S. Kar, C. Roy, M. Das, S. Mullick e R. Saha, "AI horizons: Unveiling the future of generative intelligence", en, *International Journal of Advanced Research in Science, Communication and Technology*, pp. 387–391, set. de 2023.

[168] I. J. Goodfellow et al., *Generative Adversarial Networks*, 2014. arXiv: 1406.2661 [stat.ML]. endereço: https://arxiv.org/abs/1406.2661.

[169] M. V. Zuba, R. M. Gomes e B. A. Santos, "Análise de redes neurais adversariais generativas para a geração de imagens sintéticas", *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, v. 8, n. 1, 2021.

[170] L. V. Resende, R. P. Finotti, F. S. Barbosa e A. A. Cury, "Structural damage detection with autoencoding neural networks", em *XLIII Ibero-Latin American Congress on Computational Methods in Engineering*, vol. 4, 2022.

[171] S. Bengesi, H. El-Sayed, M. K. Sarker, Y. Houkpati, J. Irungu e T. Oladunni, "Advancements in Generative AI: A Comprehensive Review of GANs, GPT, Autoencoders, Diffusion Model, and Transformers", *IEEE Access*, v. 12, pp. 69 812–69 837, 2024. DOI: 10.1109/ACCESS.2024.3397775.

[172] P. J. Cobb, "Large Language Models and generative AI, oh my!", en, *Adv. Archaeol. Pr.*, v. 11, n. 3, pp. 363–369, ago. de 2023.

[173] A. Ramesh et al., *Zero-Shot Text-to-Image Generation*, 2021. arXiv: 2102.12092 [cs.CV]. endereço: https://arxiv.org/abs/2102.12092.

[174] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford e I. Sutskever, *Jukebox: A Generative Model for Music*, 2020. arXiv: 2005.00341 [eess.AS]. endereço: https://arxiv.org/abs/2005.00341.

[175] Y. Cao et al., *A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT*, 2023. arXiv: 2303.04226 [cs.AI]. endereço: https://arxiv.org/abs/2303.04226.

[176] V. Liu, "Beyond Text-to-Image: Multimodal Prompts to Explore Generative AI", em *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, sér. CHI EA '23, Hamburg, Germany: Association for Computing Machinery, 2023, ISBN: 9781450394222. DOI: 10.1145/3544549.3577043. endereço: https://doi.org/10.1145/3544549.3577043.

[177] F. Abdullahu e H. Grabner, *Commonly Interesting Images*, 2024. arXiv: 2409.16736 [cs.CV]. endereço: https://arxiv.org/abs/2409.16736.

[178] R. Rombach, A. Blattmann, D. Lorenz, P. Esser e B. Ommer, *High-Resolution Image Synthesis with Latent Diffusion Models*, 2022. arXiv: 2112.10752 [cs.CV]. endereço: https://arxiv.org/abs/2112.10752.

[179] U. Singer et al., *Make-A-Video: Text-to-Video Generation without Text-Video Data*, 2022. arXiv: 2209.14792 [cs.CV]. endereço: https://arxiv.org/abs/2209.14792.

[180] N. Maus, P. Chao, E. Wong e J. Gardner, *Black Box Adversarial Prompting for Foundation Models*, 2023. arXiv: 2302.04237 [cs.LG]. endereço: https://arxiv.org/abs/2302.04237.

[181] A. R. Babhulgaonkar e S. P. Sonavane, "Experimenting with factored language model and generalized back-off for Hindi", en, *Int. J. Inf. Technol.*, v. 14, n. 4, pp. 2105–2118, jun. de 2022.

[182] J. Ho, N. Kalchbrenner, D. Weissenborn e T. Salimans, *Axial Attention in Multidimensional Transformers*, 2019. arXiv: 1912.12180 [cs.CV]. endereço: https://arxiv.org/abs/1912.12180.

[183] M. Aldosari e J. Miller, "On transformer autoregressive decoding for multivariate time series forecasting", em *ESANN 2023 proceesdings*, Bruges (Belgium) e online: Ciaco - i6doc.com, 2023.

[184] Y. Feng e C. Shao, "Non-Autoregressive Models for Fast Sequence Generation", em *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, S. R. El-Beltagy e X. Qiu, ed., Abu Dubai, UAE: Association for Computational Linguistics, dez. de 2022, pp. 30–35. DOI: 10.18653/v1/2022.emnlp-tutorials.6. endereço: https://aclanthology.org/2022.emnlp-tutorials.6.

[185] X. Ma, C. Zhou, X. Li, G. Neubig e E. Hovy, "FlowSeq: Non-Autoregressive Conditional Sequence Generation with Generative Flow", em *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng e X. Wan, ed., Hong Kong, China: Association

for Computational Linguistics, nov. de 2019, pp. 4282–4292. DOI: 10.18653/v1/D19-1437. endereço: https://aclanthology.org/D19-1437.

[186] Y. Liao, X. Jiang e Q. Liu, "Probabilistically Masked Language Model Capable of Autoregressive Generation in Arbitrary Word Order", em *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter e J. Tetreault, ed., Online: Association for Computational Linguistics, jul. de 2020, pp. 263–274. DOI: 10.18653/v1/2020.acl-main.24. endereço: https://aclanthology.org/2020.acl-main.24.

[187] Y. Bengio, J. Louradour, R. Collobert e J. Weston, "Curriculum learning", em *Proceedings of the 26th Annual International Conference on Machine Learning*, sér. ICML '09, Montreal, Quebec, Canada: Association for Computing Machinery, 2009, pp. 41–48, ISBN: 9781605585161. DOI: 10.1145/1553374.1553380. endereço: https://doi.org/10.1145/1553374.1553380.

[188] S. Bond-Taylor, A. Leach, Y. Long e C. G. Willcocks, "Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 44, n. 11, pp. 7327–7347, 2022. DOI: 10.1109/TPAMI.2021.3116668.

[189] X. Liang, L. Wu, J. Li e M. Zhang, "JANUS: Joint Autoregressive and Non-autoregressive Training with Auxiliary Loss for Sequence Generation", em *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva e Y. Zhang, ed., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, dez. de 2022, pp. 8050–8060. DOI: 10.18653/v1/2022.emnlp-main.550. endereço: https://aclanthology.org/2022.emnlp-main.550.

[190] Y. Su et al., "Non-Autoregressive Text Generation with Pre-trained Language Models", em *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, P. Merlo, J. Tiedemann e R. Tsarfaty, ed., Online: Association for Computational Linguistics, abr. de 2021, pp. 234–243. DOI: 10.18653/v1/2021.eacl-main.18. endereço: https://aclanthology.org/2021.eacl-main.18.

[191] X. V. Lin et al., "Few-shot Learning with Multilingual Generative Language Models", em *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva e Y. Zhang, ed., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, dez. de 2022, pp. 9019–9052. DOI: 10.18653/v1/2022.emnlp-main.616. endereço: https://aclanthology.org/2022.emnlp-main.616.

[192] A. Hendy et al., *How Good Are GPT Models at Machine Translation? A Comprehensive Evaluation*, 2023. arXiv: 2302.09210 [cs.CL]. endereço: https://arxiv.org/abs/2302.09210.

[193] Y. Xiao et al., "A Survey on Non-Autoregressive Generation for Neural Machine Translation and Beyond", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 45, n. 10, pp. 11 407–11 427, 2023. DOI: 10.1109/TPAMI.2023.3277122.

[194] S. Diederich et al., "On the design of and interaction with conversational agents: An organizing and assessing review of human-computer interaction research", en, *J. Assoc. Inf. Syst.*, v. 23, n. 1, pp. 96–138, 2022.

[195] A. Goldstein et al., "Thinking ahead: spontaneous prediction in context as a keystone of language in humans and machines", dez. de 2020.

[196] R. Fan, W. Chu, P. Chang e A. Alwan, "A CTC Alignment-Based Non-Autoregressive Transformer for End-to-End Automatic Speech Recognition", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, v. 31, pp. 1436–1448, 2023. DOI: 10.1109/TASLP.2023.3263789.

[197] F. Huang, P. Ke e M. Huang, "Directed Acyclic Transformer Pre-training for High-quality Non-autoregressive Text Generation", *Transactions of the Association for Computational Linguistics*, v. 11, pp. 941–959, 2023. DOI: 10.1162/tacl_a_00582. endereço: https://aclanthology.org/2023.tacl-1.53.

[198] Z. Tian, J. Yi, J. Tao, S. Zhang e Z. Wen, "Hybrid Autoregressive and Non-Autoregressive Transformer Models for Speech Recognition", *IEEE Signal Processing Letters*, v. 29, pp. 762–766, 2022. DOI: 10.1109/LSP.2022.3152128.

[199] S. K. Routray, A. Javali, K. P. Sharmila, M. K. Jha, M. Pappa e M. Singh, "Large Language Models (LLMs): Hypes and Realities", em *2023 International Conference on Computer Science and Emerging Technologies (CSET)*, 2023, pp. 1–6. DOI: 10.1109/CSET58993.2023.10346621.

[200] M. R. Douglas, *Large Language Models*, 2023. arXiv: 2307.05782 [cs.CL]. endereço: https://arxiv.org/abs/2307.05782.

[201] H. Naveed et al., *A Comprehensive Overview of Large Language Models*, 2024. arXiv: 2307.06435 [cs.CL]. endereço: https://arxiv.org/abs/2307.06435.

[202] R. Thoppilan et al., *LaMDA: Language Models for Dialog Applications*, 2022. arXiv: 2201.08239 [cs.CL]. endereço: https://arxiv.org/abs/2201.08239.

[203] T. J. Sejnowski, "Large language models and the reverse Turing test", en, *Neural Comput.*, v. 35, n. 3, pp. 309–342, fev. de 2023.

[204] B. Xu e M.-M. Poo, "Large language models and brain-inspired general intelligence", en, *Natl. Sci. Rev.*, v. 10, n. 10, nwad267, out. de 2023.

[205] Z. Zhao et al., " Recommender Systems in the Era of Large Language Models (LLMs) ", *IEEE Transactions on Knowledge & Data Engineering*, v. 36, n. 11, pp. 6889–6907, nov. de 2024, ISSN: 1558-2191. DOI: 10.1109/TKDE.2024.3392335. endereço: https://doi.ieeecomputersociety.org/10.1109/TKDE.2024.3392335.

[206] K. S. Kalyan, A. Rajasekharan e S. Sangeetha, "AMMU: A survey of transformer-based biomedical pretrained language models", en, *J. Biomed. Inform.*, v. 126, n. 103982, p. 103 982, fev. de 2022.

[207] G. Yenduri et al., "GPT (Generative Pre-Trained Transformer)— A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions", *IEEE Access*, v. 12, pp. 54 608–54 649, 2024. DOI: 10.1109/ACCESS.2024.3389497.

[208] N. Dehouche, "Plagiarism in the age of massive Generative Pre-trained Transformers (GPT-3)", en, *Ethics Sci. Environ. Polit.*, v. 21, pp. 17–23, mar. de 2021.

[209] D. M. Korngiebel e S. D. Mooney, "Considering the possibilities and pitfalls of Generative Pre-trained Transformer 3 (GPT-3) in healthcare delivery", en, *NPJ Digit. Med.*, v. 4, n. 1, p. 93, jun. de 2021.

[210] E. Sezgin, J. Sirrianni e S. L. Linwood, "Operationalizing and implementing pretrained, large artificial intelligence linguistic models in the US health care system: Outlook of generative pretrained transformer 3 (GPT-3) as a service model", en, *JMIR Med. Inform.*, v. 10, n. 2, e32875, fev. de 2022.

[211] R. Luo et al., "BioGPT: generative pre-trained transformer for biomedical text generation and mining", en, *Brief. Bioinform.*, v. 23, n. 6, nov. de 2022.

[212] J. Zhu et al., *VL-GPT: A Generative Pre-trained Transformer for Vision and Language Understanding and Generation*, 2023. arXiv: 2312.09251 [cs.CV]. endereço: https://arxiv.org/abs/2312.09251.

[213] H. Touvron et al., *LLaMA: Open and Efficient Foundation Language Models*, 2023. arXiv: 2302.13971 [cs.CL]. endereço: https://arxiv.org/abs/2302.13971.

[214] H. Touvron et al., *Llama 2: Open Foundation and Fine-Tuned Chat Models*, 2023. arXiv: 2307.09288 [cs.CL]. endereço: https://arxiv.org/abs/2307.09288.

[215] A. Grattafiori et al., *The Llama 3 Herd of Models*, 2024. arXiv: 2407.21783 [cs.AI]. endereço: https://arxiv.org/abs/2407.21783.

[216] R. Taori et al., *Stanford Alpaca: An Instruction-following LLaMA model*, https://github.com/tatsu-lab/stanford_alpaca, 2023.

[217] W.-L. Chiang et al., *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*, mar. de 2023. endereço: https://lmsys.org/blog/2023-03-30-vicuna/.

[218] X. Geng et al., *Koala: A Dialogue Model for Academic Research*, Blog post, abr. de 2023. acesso em 3 de abr. de 2023. endereço: https://bair.berkeley.edu/blog/2023/04/03/koala/.

[219] A. Q. Jiang et al., *Mistral 7B*, 2023. arXiv: 2310.06825 [cs.CL]. endereço: https://arxiv.org/abs/2310.06825.

[220] A. Q. Jiang et al., *Mixtral of Experts*, 2024. arXiv: 2401.04088 [cs.LG]. endereço: https://arxiv.org/abs/2401.04088.

[221] H. Tran, Z. Yang, Z. Yao e H. Yu, "BioInstruct: instruction tuning of large language models for biomedical natural language processing", en, *J. Am. Med. Inform. Assoc.*, v. 31, n. 9, pp. 1821–1832, set. de 2024.

[222] R. Zhang et al., *LLaMA-Adapter: Efficient Fine-tuning of Language Models with Zero-init Attention*, 2024. arXiv: 2303.16199 [cs.CV]. endereço: https://arxiv.org/abs/2303.16199.

[223] Y. Cui, Z. Yang e X. Yao, *Efficient and Effective Text Encoding for Chinese LLaMA and Alpaca*, 2024. arXiv: 2304.08177 [cs.CL]. endereço: https://arxiv.org/abs/2304.08177.

[224] X. Zhang, C. Tian, X. Yang, L. Chen, Z. Li e L. R. Petzold, *AlpaCare:Instruction-tuned Large Language Models for Medical Application*, 2024. arXiv: 2310.14558 [cs.CL]. endereço: https://arxiv.org/abs/2310.14558.

[225] A. Chowdhery et al., *PaLM: Scaling Language Modeling with Pathways*, 2022. arXiv: 2204.02311 [cs.CL]. endereço: https://arxiv.org/abs/2204.02311.

[226] H. W. Chung et al., "Scaling Instruction-Finetuned Language Models", *Journal of Machine Learning Research*, v. 25, n. 70, pp. 1–53, 2024. endereço: http://jmlr.org/papers/v25/23-0870.html.

[227] D. Vilar, M. Freitag, C. Cherry, J. Luo, V. Ratnakar e G. Foster, "Prompting PaLM for Translation: Assessing Strategies and Performance", em *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber e N. Okazaki, ed., Toronto, Canada: Association for Computational Linguistics, jul. de 2023, pp. 15 406–15 427. DOI: 10.18653/v1/2023.acl-long.859. endereço: https://aclanthology.org/2023.acl-long.859.

[228] E. Briakou, C. Cherry e G. Foster, "Searching for Needles in a Haystack: On the Role of Incidental Bilingualism in PaLM's Translation Capability", em *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber e N. Okazaki, ed., Toronto, Canada: Association for Computational Linguistics, jul. de 2023, pp. 9432–9452. DOI: 10.

18653/v1/2023.acl-long.524. endereço: https://aclanthology.org/2023.acl-long.524.

[229] N. Pourkamali e S. E. Sharifi, *Machine Translation with Large Language Models: Prompt Engineering for Persian, English, and Russian Directions*, 2024. arXiv: 2401.08429 [cs.CL]. endereço: https://arxiv.org/abs/2401.08429.

[230] Y. Tay et al., "Transcending Scaling Laws with 0.1% Extra Compute", em *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino e K. Bali, ed., Singapore: Association for Computational Linguistics, dez. de 2023, pp. 1471–1486. DOI: 10.18653/v1/2023.emnlp-main.91. endereço: https://aclanthology.org/2023.emnlp-main.91.

[231] R. Anil et al., *PaLM 2 Technical Report*, 2023. arXiv: 2305.10403 [cs.CL]. endereço: https://arxiv.org/abs/2305.10403.

[232] G. Team et al., *Gemini: A Family of Highly Capable Multimodal Models*, 2024. arXiv: 2312.11805 [cs.CL]. endereço: https://arxiv.org/abs/2312.11805.

[233] L. J. Labrague, "Utilizing artificial intelligence-based tools for addressing clinical queries: ChatGPT versus Google Gemini", en, *J. Nurs. Educ.*, v. 63, n. 8, pp. 556–559, ago. de 2024.

[234] D. Noever e S. E. M. Noever, *Multimodal Analysis Of Google Bard And GPT-Vision: Experiments In Visual Reasoning*, 2023. arXiv: 2309.16705 [cs.CV]. endereço: https://arxiv.org/abs/2309.16705.

[235] Y. Liang et al., "XGLUE: A New Benchmark Dataset for Cross-lingual Pretraining, Understanding and Generation", em *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He e Y. Liu, ed., Online: Association for Computational Linguistics, nov. de 2020, pp. 6008–6018. DOI: 10.18653/v1/2020.emnlp-main.484. endereço: https://aclanthology.org/2020.emnlp-main.484.

[236] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi e N. Smith, *Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping*, 2020. arXiv: 2002.06305 [cs.CL]. endereço: https://arxiv.org/abs/2002.06305.

[237] F. Stollenwerk, *Adaptive Fine-Tuning of Transformer-Based Language Models for Named Entity Recognition*, 2022. arXiv: 2202.02617 [cs.CL]. endereço: https://arxiv.org/abs/2202.02617.

[238] D. P. Kingma e J. Ba, *Adam: A Method for Stochastic Optimization*, 2017. arXiv: 1412.6980 [cs.LG]. endereço: https://arxiv.org/abs/1412.6980.

[239] I. Loshchilov e F. Hutter, *Decoupled Weight Decay Regularization*, 2019. arXiv: 1711.05101 [cs.LG]. endereço: https://arxiv.org/abs/1711.05101.

[240] R. Llugsi, S. E. Yacoubi, A. Fontaine e P. Lupera, "Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andean city of Quito", em *2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM)*, 2021, pp. 1–6. DOI: 10.1109/ETCM53643.2021.9590681.

[241] Z. Zhuang, M. Liu, A. Cutkosky e F. Orabona, *Understanding AdamW through Proximal Methods and Scale-Freeness*, 2022. arXiv: 2202.00089 [cs.LG]. endereço: https://arxiv.org/abs/2202.00089.

[242] H. Naganuma et al., *Empirical Study on Optimizer Selection for Out-of-Distribution Generalization*, 2023. arXiv: 2211.08583 [cs.LG]. endereço: https://arxiv.org/abs/2211.08583.

[243] L. Liu et al., *On the Variance of the Adaptive Learning Rate and Beyond*, 2021. arXiv: 1908.03265 [cs.LG]. endereço: https://arxiv.org/abs/1908.03265.

[244] D. O. Melinte e L. Vladareanu, "Facial Expressions Recognition for Human–Robot Interaction Using Deep Convolutional Neural Networks with Rectified Adam Optimizer", *Sensors*, v. 20, n. 8, 2020, ISSN: 1424-8220. DOI: 10.3390/s20082393. endereço: https://www.mdpi.com/1424-8220/20/8/2393.

[245] D. Pasechnyuk, A. Prazdnichnykh, M. Evtikhiev e T. Bryksin, "Judging Adam: Studying the Performance of Optimization Methods on ML4SE Tasks", em *Proceedings of the 45th International Conference on Software Engineering: New Ideas and Emerging Results*, sér. ICSE-NIER '23, Melbourne, Australia: IEEE Press, 2023, pp. 117–122, ISBN: 9798350300390. DOI: 10.1109/ICSE-NIER58687.2023.00027. endereço: https://doi.org/10.1109/ICSE-NIER58687.2023.00027.

[246] K. K. Mamidala e S. K. Sanampudi, "Text summarization on Telugu e-news based on long-short term memory with rectified Adam optimizer", *Int. J. Comput. Digit. Syst.*, v. 11, n. 1, pp. 355–368, jan. de 2022.

[247] B. Heo et al., *AdamP: Slowing Down the Slowdown for Momentum Optimizers on Scale-invariant Weights*, 2021. arXiv: 2006.08217 [cs.LG]. endereço: https://arxiv.org/abs/2006.08217.

[248] A. Defazio e S. Jelassi, "Adaptivity without compromise: a momentumized, adaptive, dual averaged gradient method for stochastic optimization", *J. Mach. Learn. Res.*, v. 23, n. 1, jan. de 2022, ISSN: 1532-4435.

[249] J. Duchi, E. Hazan e Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization", *Journal of Machine Learning Research*, v. 12, n. 61, pp. 2121–2159, 2011. endereço: http://jmlr.org/papers/v12/duchi11a.html.

[250] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever e R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting", *J. Mach. Learn. Res.*, v. 15, n. 1, pp. 1929–1958, jan. de 2014, ISSN: 1532-4435.

[251] M. Andriushchenko, F. D'Angelo, A. Varre e N. Flammarion, *Why Do We Need Weight Decay in Modern Deep Learning?*, 2023. arXiv: 2310.04415 [cs.LG]. endereço: https://arxiv.org/abs/2310.04415.

[252] X. Liu et al., *Adversarial Training for Large Neural Language Models*, 2020. arXiv: 2004.08994 [cs.CL]. endereço: https://arxiv.org/abs/2004.08994.

[253] J. Yosinski, J. Clune, Y. Bengio e H. Lipson, "How transferable are features in deep neural networks?", em *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, sér. NIPS'14, Montreal, Canada: MIT Press, 2014, pp. 3320–3328.

[254] T. Zhang, F. Wu, A. Katiyar, K. Q. Weinberger e Y. Artzi, *Revisiting Few-sample BERT Fine-tuning*, 2021. arXiv: 2006.05987 [cs.CL]. endereço: https://arxiv.org/abs/2006.05987.

[255] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun e R. Fergus, "Regularization of Neural Networks using DropConnect", em *Proceedings of the 30th International Conference on Machine Learning*, S. Dasgupta e D. McAllester, ed., sér. Proceedings of Machine Learning Research, vol. 28, Atlanta, Georgia, USA: PMLR, 17–19 Jun de 2013, pp. 1058–1066. endereço: https://proceedings.mlr.press/v28/wan13.html.

[256] C. Lee, K. Cho e W. Kang, *Mixout: Effective Regularization to Finetune Large-scale Pretrained Language Models*, 2020. arXiv: 1909.11299 [cs.LG]. endereço: https://arxiv.org/abs/1909.11299.

[257] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov e A. G. Wilson, *Averaging Weights Leads to Wider Optima and Better Generalization*, 2019. arXiv: 1803.05407 [cs.LG]. endereço: https://arxiv.org/abs/1803.05407.

[258] H. Guo, J. Jin e B. Liu, "Stochastic Weight Averaging Revisited", *Applied Sciences*, v. 13, n. 5, 2023, ISSN: 2076-3417. endereço: https://www.mdpi.com/2076-3417/13/5/2935.

[259] P. Lu, I. Kobyzev, M. Rezagholizadeh, A. Rashid, A. Ghodsi e P. Langlais, *Improving Generalization of Pre-trained Language Models via Stochastic Weight Averaging*, 2022. arXiv: 2212.05956 [cs.CL]. endereço: https://arxiv.org/abs/2212.05956.

[260] A. Talman, H. Celikkanat, S. Virpioja, M. Heinonen e J. Tiedemann, *Uncertainty-Aware Natural Language Inference with Stochastic Weight Averaging*, 2023. arXiv: 2304.04726 [cs.CL]. endereço: https://arxiv.org/abs/2304.04726.

[261] E. Onal, K. Flöge, E. Caldwell, A. Sheverdin e V. Fortuin, *Gaussian Stochastic Weight Averaging for Bayesian Low-Rank Adaptation of Large Language Models*, 2024. arXiv: 2405.03425 [cs.CL]. endereço: https://arxiv.org/abs/2405.03425.

[262] J. Howard e S. Ruder, "Universal Language Model Fine-tuning for Text Classification", em *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych e Y. Miyao, ed., Melbourne, Australia: Association for Computational Linguistics, jul. de 2018, pp. 328–339. DOI: 10.18653/v1/P18-1031. endereço: https://aclanthology.org/P18-1031.

[263] K. Clark, M.-T. Luong, Q. V. Le e C. D. Manning, *ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators*, 2020. arXiv: 2003.10555 [cs.CL]. endereço: https://arxiv.org/abs/2003.10555.

[264] Y. You et al., *Large Batch Optimization for Deep Learning: Training BERT in 76 minutes*, 2020. arXiv: 1904.00962 [cs.LG]. endereço: https://arxiv.org/abs/1904.00962.

[265] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov e Q. V. Le, "XLNet: generalized autoregressive pretraining for language understanding", em *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[266] N. Q. K. Le, Q.-T. Ho, T.-T.-D. Nguyen e Y.-Y. Ou, "A transformer architecture based on BERT and 2D convolutional neural network to identify DNA enhancers from sequence information", en, *Brief. Bioinform.*, v. 22, n. 5, set. de 2021.

[267] F. Meng, J. Feng, D. Yin e M. Hu, "A New Fine-Tuning Architecture Based on Bert for Word Relation Extraction", em *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II*, Dunhuang, China: Springer-Verlag, 2019, pp. 327–337, ISBN: 978-3-030-32235-9. DOI: 10.1007/978-3-030-32236-6_29. endereço: https://doi.org/10.1007/978-3-030-32236-6_29.

[268] Z. Luo et al., "DecBERT: Enhancing the Language Understanding of BERT with Causal Attention Masks", em *Findings of the Association for Computational Linguistics: NAACL 2022*, M. Carpuat, M.-C. de Marneffe e I. V. Meza Ruiz, ed., Seattle, United States: Association for Computational Linguistics, jul. de 2022, pp. 1185–1197. DOI: 10.18653/v1/2022.findings-naacl.89. endereço: https://aclanthology.org/2022.findings-naacl.89.

[269] Z. Gao, A. Feng, X. Song e X. Wu, "Target-Dependent Sentiment Classification With BERT", *IEEE Access*, v. 7, pp. 154 290–154 299, 2019. DOI: 10.1109/ACCESS.2019.2946594.

[270] S. M. Ali Shah, S. W. Taju, Q.-T. Ho, T.-T.-D. Nguyen e Y.-Y. Ou, "GT-Finder: Classify the family of glucose transporters with pre-trained BERT language models", en, *Comput. Biol. Med.*, v. 131, n. 104259, p. 104 259, abr. de 2021.

[271] Z. Fu, W. Zhou, J. Xu, H. Zhou e L. Li, "Contextual Representation Learning beyond Masked Language Modeling", em *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov e A. Villavicencio, ed., Dublin, Ireland: Association for Computational Linguistics, mai. de 2022, pp. 2701–2714. DOI: 10.18653/v1/2022.acl-long.193. endereço: https://aclanthology.org/2022.acl-long.193.

[272] H. Nishimoto, "Effective deep learning through bidirectional reading on masked language model", em *Human Systems Engineering and Design (IHSED2021) Future Trends and Applications*, AHFE International, 2021.

[273] C. Amrhein e R. Sennrich, "How Suitable Are Subword Segmentation Strategies for Translating Non-Concatenative Morphology?", em *Findings of the Association for Computational Linguistics: EMNLP 2021*, M.-F. Moens, X. Huang, L. Specia e S. W.-t. Yih, ed., Punta Cana, Dominican Republic: Association for Computational Linguistics, nov. de 2021, pp. 689–705. DOI: 10.18653/v1/2021.findings-emnlp.60. endereço: https://aclanthology.org/2021.findings-emnlp.60.

[274] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer e O. Levy, "SpanBERT: Improving Pre-training by Representing and Predicting Spans", *Transactions of the Association for Computational Linguistics*, v. 8, M. Johnson, B. Roark e A. Nenkova, ed., pp. 64–77, 2020. DOI: 10.1162/tacl_a_00300. endereço: https://aclanthology.org/2020.tacl-1.5.

[275] Y. Levine et al., *PMI-Masking: Principled masking of correlated spans*, 2020. arXiv: 2010.01825 [cs.LG]. endereço: https://arxiv.org/abs/2010.01825.

[276] Y. Cui, W. Che, T. Liu, B. Qin e Z. Yang, "Pre-Training With Whole Word Masking for Chinese BERT", *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, v. 29, pp. 3504–3514, nov. de 2021, ISSN: 2329-9290. DOI: 10.1109/TASLP.2021.3124365. endereço: https://doi.org/10.1109/TASLP.2021.3124365.

[277] E. Barba, N. Campolungo e R. Navigli, "DMLM: Descriptive Masked Language Modeling", em *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber e N. Okazaki, ed., Toronto, Canada: Association for Computational Linguistics, jul. de 2023, pp. 12 770–12 788. DOI: 10.18653/v1/2023.findings-acl.808. endereço: https://aclanthology.org/2023.findings-acl.808.

[278] L. A. Mullen, K. Benoit, O. Keyes, D. Selivanov e J. Arnold, "Fast, Consistent Tokenization of Natural Language Text", *Journal of Open Source Software*, v. 3, n. 23, p. 655, 2018. DOI: 10.21105/joss.00655. endereço: https://doi.org/10.21105/joss.00655.

[279] C. P. Chai, "Comparison of text preprocessing methods", *Natural Language Engineering*, v. 29, pp. 509–553, 2022. DOI: 10.1017/S1351324922000213.

[280] C. Toraman, E. H. Yilmaz, F. Şahi.nuç e O. Ozcelik, "Impact of Tokenization on Language Models: An Analysis for Turkish", *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, v. 22, n. 4, mar. de 2023, ISSN: 2375-4699. DOI: 10.1145/3578707. endereço: https://doi.org/10.1145/3578707.

[281] A. Nayak, H. Timmapathini, K. Ponnalagu e V. Gopalan Venkoparao, "Domain adaptation challenges of BERT in tokenization and sub-word representations of Out-of-Vocabulary words", em *Proceedings of the First Workshop on Insights from Negative Results in NLP*, A. Rogers, J. Sedoc e A. Rumshisky, ed., Online: Association for Computational Linguistics, nov. de 2020, pp. 1–5. DOI: 10.18653/v1/2020.insights-1.1. endereço: https://aclanthology.org/2020.insights-1.1.

[282] X. Song, A. Salcianu, Y. Song, D. Dopson e D. Zhou, "Fast WordPiece Tokenization", em *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia e S. W.-t. Yih, ed., Online e Punta Cana, Dominican Republic: Association for Computational Linguistics, nov. de 2021, pp. 2089–2103. DOI: 10.18653/v1/2021.emnlp-main.160. endereço: https://aclanthology.org/2021.emnlp-main.160.

[283] A. Özçift, K. Akarsu, F. Yumuk e C. Söylemez, "Advancing natural language processing (NLP) applications of morphologically rich languages with bidirectional encoder representations from transformers (BERT): an empirical case study for Turkish", en, *Automatika*, v. 62, n. 2, pp. 226–238, abr. de 2021.

[284] R. Sennrich, B. Haddow e A. Birch, "Neural Machine Translation of Rare Words with Subword Units", em *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Erk e N. A. Smith, ed., Berlin, Germany: Association for Computational Linguistics, ago. de 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162. endereço: https://aclanthology.org/P16-1162.

[285] Y. Liu et al., *RoBERTa: A Robustly Optimized BERT Pretraining Approach*, 2019. arXiv: 1907.11692 [cs.CL]. endereço: https://arxiv.org/abs/1907.11692.

[286] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy e S. Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding", em *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, T. Linzen, G. Chrupała e A. Alishahi, ed., Brussels, Belgium: Association for Computational Linguistics, nov. de 2018, pp. 353–355. DOI: 10.18653/v1/W18-5446. endereço: https://aclanthology.org/W18-5446.

[287] P. Rajpurkar, J. Zhang, K. Lopyrev e P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text", em *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, J. Su, K. Duh e X. Carreras, ed., Austin, Texas: Association for Computational Linguistics, nov. de 2016, pp. 2383–2392. DOI: 10.18653/v1/D16-1264. endereço: https://aclanthology.org/D16-1264.

[288] P. Rajpurkar, R. Jia e P. Liang, "Know What You Don't Know: Unanswerable Questions for SQuAD", em *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, I. Gurevych e Y. Miyao, ed., Melbourne, Australia: Association for Computational Linguistics, jul. de 2018, pp. 784–789. DOI: 10.18653/v1/P18-2124. endereço: https://aclanthology.org/P18-2124.

[289] A. Conneau et al., "XNLI: Evaluating Cross-lingual Sentence Representations", em *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier e J. Tsujii, ed., Brussels, Belgium: Association for Computational Linguistics, out. de 2018, pp. 2475–2485. DOI: 10.18653/v1/D18-1269. endereço: https://aclanthology.org/D18-1269.

[290] P. Lewis, B. Oguz, R. Rinott, S. Riedel e H. Schwenk, "MLQA: Evaluating Cross-lingual Extractive Question Answering", em *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter e J. Tetreault, ed., Online: Association for Computational Linguistics, jul. de 2020, pp. 7315–7330. DOI: 10.18653/v1/2020.acl-main.653. endereço: https://aclanthology.org/2020.acl-main.653.

[291] S. Kofi Akpatsa et al., "Online news sentiment classification using DistilBERT", en, *Journal of Quantum Computing*, v. 4, n. 1, pp. 1–11, 2022.

[292] M. Abadeer, "Assessment of DistilBERT performance on Named Entity Recognition task for the detection of Protected Health Information and medical concepts", em *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, A. Rumshisky, K. Roberts, S. Bethard e T. Naumann, ed., Online: Association for Computational Linguistics, nov. de 2020, pp. 158–167. DOI: 10.18653/v1/2020.

clinicalnlp-1.18. endereço: https://aclanthology.org/2020.clinicalnlp-1.18.

[293] V. Dogra, A. Singh, S. Verma, Kavita, N. Z. Jhanjhi e M. N. Talib, "Analyzing DistilBERT for sentiment classification of banking financial news", em *Intelligent Computing and Innovation on Data Science*, sér. Lecture notes in networks and systems, Singapore: Springer Singapore, 2021, pp. 501–510.

[294] P. Delobelle, T. Winters e B. Berendt, "RobBERT: a Dutch RoBERTa-based Language Model", em *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, nov. de 2020, pp. 3255–3265. DOI: 10.18653/v1/2020.findings-emnlp.292. endereço: https://www.aclweb.org/anthology/2020.findings-emnlp.292.

[295] P. He, X. Liu, J. Gao e W. Chen, *DeBERTa: Decoding-enhanced BERT with Disentangled Attention*, 2021. arXiv: 2006.03654 [cs.CL]. endereço: https://arxiv.org/abs/2006.03654.

[296] R. K. Singh, M. K. Sachan e R. B. Patel, "Cross-domain sentiment classification using decoding-enhanced bidirectional encoder representations from transformers with disentangled attention", en, *Concurr. Comput.*, v. 35, n. 6, pp. 1–1, mar. de 2023.

[297] W. Liang e Y. Liang, *BPDec: Unveiling the Potential of Masked Language Modeling Decoder in BERT pretraining*, 2024. arXiv: 2401.15861 [cs.CL]. endereço: https://arxiv.org/abs/2401.15861.

[298] P. He, J. Gao e W. Chen, *DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing*, 2023. arXiv: 2111.09543 [cs.CL]. endereço: https://arxiv.org/abs/2111.09543.

[299] D. Santos, N. Seco, N. Cardoso e R. Vilela, "HAREM: An Advanced NER Evaluation Contest for Portuguese", em *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, N. Calzolari et al., ed., Genoa, Italy: European Language Resources Association (ELRA), mai. de 2006. endereço: http://www.lrec-conf.org/proceedings/lrec2006/pdf/59_pdf.pdf.

[300] P. Borchert, K. Coussement, J. De Weerdt e A. De Caigny, "Industry-sensitive language modeling for business", *European Journal of Operational Research*, v. 315, n. 2, pp. 691–702, 2024, ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2024.01.023. endereço: https://www.sciencedirect.com/science/article/pii/S0377221724000444.

[301] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma e R. Soricut, *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*, 2020. arXiv: 1909.11942 [cs.CL]. endereço: https://arxiv.org/abs/1909.11942.

[302] S. Yu, J. Su e D. Luo, "Improving BERT-Based Text Classification With Auxiliary Sentence and Domain Knowledge", *IEEE Access*, v. 7, pp. 176 600–176 612, 2019. DOI: 10.1109/ACCESS.2019.2953990.

[303] M. Kowsher, A. A. Sami, N. J. Prottasha, M. S. Arefin, P. K. Dhar e T. Koshiba, "Bangla-BERT: Transformer-Based Efficient Model for Transfer Learning and Language Understanding", *IEEE Access*, v. 10, pp. 91 855–91 870, 2022. DOI: 10.1109/ACCESS.2022.3197662.

[304] A. Radford, K. Narasimhan, T. Salimans e I. Sutskever, "Improving language understanding by generative pre-training", 2018.

[305] T. Webb, K. J. Holyoak e H. Lu, "Emergent analogical reasoning in large language models", en, *Nat. Hum. Behav.*, v. 7, n. 9, pp. 1526–1541, set. de 2023.

[306] J. Ye et al., *A Comprehensive Capability Analysis of GPT-3 and GPT-3.5 Series Models*, 2023. arXiv: 2303.10420 [cs.CL]. endereço: https://arxiv.org/abs/2303.10420.

[307] B. Meskó, "The impact of multimodal large language models on health care's future", en, *J. Med. Internet Res.*, v. 25, e52865, nov. de 2023.

[308] M. C. Schubert, M. Lasotta, F. Sahm, W. Wick e V. Venkataramani, "Evaluating the multimodal capabilities of generative AI in complex clinical diagnostics", nov. de 2023.

[309] D. Zhu, J. Chen, X. Shen, X. Li e M. Elhoseiny, *MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models*, 2023. arXiv: 2304.10592 [cs.CV]. endereço: https://arxiv.org/abs/2304.10592.

[310] S. Bubeck et al., *Sparks of Artificial General Intelligence: Early experiments with GPT-4*, 2023. arXiv: 2303.12712 [cs.CL]. endereço: https://arxiv.org/abs/2303.12712.

[311] V. Zouhar et al., "A Formal Perspective on Byte-Pair Encoding", em *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber e N. Okazaki, ed., Toronto, Canada: Association for Computational Linguistics, jul. de 2023, pp. 598–614. DOI: 10.18653/v1/2023.findings-acl.38. endereço: https://aclanthology.org/2023.findings-acl.38.

[312] E. Frantar, S. Ashkboos, T. Hoefler e D. Alistarh, *GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers*, 2023. arXiv: 2210.17323 [cs.LG]. endereço: https://arxiv.org/abs/2210.17323.

[313] M. Zong e B. Krishnamachari, *A survey on GPT-3*, 2022. arXiv: 2212.00857 [cs.CL]. endereço: https://arxiv.org/abs/2212.00857.

[314] L. Chen, M. Zaharia e J. Zou, *How is ChatGPT's behavior changing over time?*, 2023. arXiv: 2307.09009 [cs.CL]. endereço: https://arxiv.org/abs/2307.09009.

[315] X. Chen et al., *How Robust is GPT-3.5 to Predecessors? A Comprehensive Study on Language Understanding Tasks*, 2023. arXiv: 2303.00293 [cs.CL]. endereço: https://arxiv.org/abs/2303.00293.

[316] J. A. Baktash e M. Dawodi, *Gpt-4: A Review on Advancements and Opportunities in Natural Language Processing*, 2023. arXiv: 2305.03195 [cs.CL]. endereço: https://arxiv.org/abs/2305.03195.

[317] K. Jafarzade, "The Role of GPT Models in Education: Challenges and Solutions", em *2023 5th International Conference on Problems of Cybernetics and Informatics (PCI)*, 2023, pp. 1–3. DOI: 10.1109/PCI60110.2023.10325940.

[318] K. Gao et al., *Examining User-Friendly and Open-Sourced Large GPT Models: A Survey on Language, Multimodal, and Scientific GPT Models*, 2023. arXiv: 2308.14149 [cs.CL]. endereço: https://arxiv.org/abs/2308.14149.

[319] A. Broder, "On the resemblance and containment of documents", em *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, 1997, pp. 21–29. DOI: 10.1109/SEQUEN.1997.666900.

[320] K. Lee et al., "Deduplicating Training Data Makes Language Models Better", em *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov e A. Villavicencio, ed., Dublin, Ireland: Association for Computational Linguistics, mai. de 2022, pp. 8424–8445. DOI: 10.18653/v1/2022.acl-long.577. endereço: https://aclanthology.org/2022.acl-long.577.

[321] K. S. Kalyan, "A survey of GPT-3 family large language models including ChatGPT and GPT-4", *Natural Language Processing Journal*, v. 6, p. 100 048, 2024, ISSN: 2949-7191. DOI: https://doi.org/10.1016/j.nlp.2023.100048. endereço: https://www.sciencedirect.com/science/article/pii/S2949719123000456.

[322] L. R. Gale, W. C. Heath e R. W. Ressler, "An Economic Analysis of Hate Crime", *Eastern Economic Journal*, v. 28, n. 2, pp. 203–216, 2002, ISSN: 00945056, 19394632. acesso em 2 de fev. de 2025. endereço: http://www.jstor.org/stable/40326097.

[323] A. Curthoys, *Identifying the Effect of Unemployment on Hate Crime*, Renée Crown University Honors Thesis Projects - All. 33, 2013. endereço: https://surface.syr.edu/honors_capstone/33/.

[324] D. Dharmapala e R. H. McAdams, "Words that kill: An economic perspective on hate speech and hate crimes", en, *SSRN Electron. J.*, 2002.

[325] M. L. Williams, P. Burnap, A. Javed, H. Liu e S. Ozalp, "Hate in the Machine: Anti-Black and Anti-Muslim Social Media Posts as Predictors of Offline Racially and Religiously Aggravated Crime", *The British Journal of Criminology*, v. 60, n. 1, pp. 93–117, jul. de 2019, ISSN: 0007-0955. DOI: 10.1093/bjc/azz049. eprint: https://academic.oup.com/bjc/article-pdf/60/1/93/31634412/azz049.pdf. endereço: https://doi.org/10.1093/bjc/azz049.

[326] S. Vosoughi, D. Roy e S. Aral, "The spread of true and false news online", *Science*, v. 359, n. 6380, pp. 1146–1151, 2018. DOI: 10.1126/science.aap9559. eprint: https://www.science.org/doi/pdf/10.1126/science.aap9559. endereço: https://www.science.org/doi/abs/10.1126/science.aap9559.

[327] S. Dong e C. Liu, "Sentiment Classification for Financial Texts Based on Deep Learning", *Computational Intelligence and Neuroscience*, v. 2021, n. 1, p. 9 524 705, 2021. DOI: https://doi.org/10.1155/2021/9524705. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2021/9524705. endereço: https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/9524705.

[328] R. Singh, V. Sharma, R. Kashyap e M. Manwal, "Automated Multi-Page Document Classification and Information Extraction for Insurance Applications using Deep Learning Techniques", em *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2024, pp. 1–7. DOI: 10.1109/ICRITO61523.2024.10522111.

[329] R. de Pelle e V. Moreira, "Offensive Comments in the Brazilian Web: a dataset and baseline results", em *Anais do VI Brazilian Workshop on Social Network Analysis and Mining*, São Paulo: SBC, 2017, pp. 510–519. DOI: 10.5753/brasnam.2017.3260. endereço: https://sol.sbc.org.br/index.php/brasnam/article/view/3260.

[330] R. M. Silva, R. L. Santos, T. A. Almeida e T. A. Pardo, "Towards automatically filtering fake news in Portuguese", *Expert Systems with Applications*, v. 146, p. 113 199, 2020, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.113199. endereço: https://www.sciencedirect.com/science/article/pii/S0957417420300257.

[331] A. E. d. O. Carosia, A. E. A. d. Silva e G. P. Coelho, *Replication data for: Predicting the Brazilian Stock Market using Sentiment Analysis, Technical Indicators, and Stock Prices*, versão V1, 2022. DOI: 10.25824/redu/GFJHFK. endereço: https://doi.org/10.25824/redu/GFJHFK.

[332] R. Faria de Azevedo, T. H. Eduardo Muniz, C. Pimentel, G. Jose de Assis Foureaux, B. Caldeira Macedo e D. d. L. Vasconcelos, "BBRC: Brazilian Banking Regulation Corpora", em *Proceedings of the Joint Workshop of the 7th Financial Technology and Natural Language Processing, the 5th Knowledge Discovery from Unstructured Data in Financial Services, and the 4th Workshop on Economics and Natural*

Language Processing @ LREC-COLING 2024, C.-C. Chen et al., ed., Torino, Italia: ELRA e ICCL, mai. de 2024, pp. 150–166. endereço: https://aclanthology.org/2024.finnlp-1.15.

[333] T. Kudo e J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing", em *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, E. Blanco e W. Lu, ed., Brussels, Belgium: Association for Computational Linguistics, nov. de 2018, pp. 66–71. DOI: 10.18653/v1/D18-2012. endereço: https://aclanthology.org/D18-2012.

[334] T. Kudo, "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates", em *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych e Y. Miyao, ed., Melbourne, Australia: Association for Computational Linguistics, jul. de 2018, pp. 66–75. DOI: 10.18653/v1/P18-1007. endereço: https://aclanthology.org/P18-1007.

[335] P. Micikevicius et al., *Mixed Precision Training*, 2018. arXiv: 1710.03740 [cs.AI]. endereço: https://arxiv.org/abs/1710.03740.

[336] P. Xu et al., "Optimizing Deeper Transformers on Small Datasets", em *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li e R. Navigli, ed., Online: Association for Computational Linguistics, ago. de 2021, pp. 2089–2102. DOI: 10.18653/v1/2021.acl-long.163. endereço: https://aclanthology.org/2021.acl-long.163.

[337] P. Lu, I. Kobyzev, M. Rezagholizadeh, A. Rashid, A. Ghodsi e P. Langlais, "Improving Generalization of Pre-trained Language Models via Stochastic Weight Averaging", em *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva e Y. Zhang, ed., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, dez. de 2022, pp. 4948–4954. DOI: 10.18653/v1/2022.findings-emnlp.363. endereço: https://aclanthology.org/2022.findings-emnlp.363/.

[338] K. Dey, P. Tarannum, M. A. Hasan, I. Razzak e U. Naseem, *Better to Ask in English: Evaluation of Large Language Models on English, Low-resource and Cross-Lingual Settings*, 2024. arXiv: 2410.13153 [cs.CL]. endereço: https://arxiv.org/abs/2410.13153.

[339] Y. Jin, M. Chandra, G. Verma, Y. Hu, M. De Choudhury e S. Kumar, "Better to Ask in English: Cross-Lingual Evaluation of Large Language Models for Healthcare Queries", em *Proceedings of the ACM Web Conference 2024*, sér. WWW '24, Singapore, Singapore: Association for Computing Machinery, 2024, pp. 2627–2638,

ISBN: 9798400701719. DOI: 10.1145/3589334.3645643. endereço: https://doi.org/10.1145/3589334.3645643.

[340] S. V. Balkus e D. Yan, "Improving short text classification with augmented data using GPT-3", *Natural Language Engineering*, pp. 1–30, 2023. DOI: 10.1017/S1351324923000438.

[341] Y. Jeong e E. Kim, "SciDeBERTa: Learning DeBERTa for Science Technology Documents and Fine-Tuning Information Extraction Tasks", *IEEE Access*, v. 10, pp. 60 805–60 813, 2022. DOI: 10.1109/ACCESS.2022.3180830.

[342] M. Wortsman et al., *Small-scale proxies for large-scale Transformer training instabilities*, 2023. arXiv: 2309.14322 [cs.LG]. endereço: https://arxiv.org/abs/2309.14322.

[343] W. Kwon, S. Kim, M. W. Mahoney, J. Hassoun, K. Keutzer e A. Gholami, *A Fast Post-Training Pruning Framework for Transformers*, 2022. arXiv: 2204.09656 [cs.CL]. endereço: https://arxiv.org/abs/2204.09656.

[344] Y. Bondarenko, M. Nagel e T. Blankevoort, "Understanding and Overcoming the Challenges of Efficient Transformer Quantization", em *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia e S. W.-t. Yih, ed., Online e Punta Cana, Dominican Republic: Association for Computational Linguistics, nov. de 2021, pp. 7947–7969. DOI: 10.18653/v1/2021.emnlp-main.627. endereço: https://aclanthology.org/2021.emnlp-main.627.