Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

Processamento de Linguagem Natural e Machine Learning na Categorização de Notícias

José Carlos Lopes Varanda

Dissertação

Mestrado em Engenharia Informática - Sistemas de Informação

Trabalho efetuado sob a orientação de

Professora Doutora Ana Cristina Wanzeller Guedes de Lacerda

Professora Doutora Joana Rita da Silva Fialho

Março de 2026

Politécnico
de Viseu

Escola Superior
de Tecnologia
e Gestão de Viseu

Processamento de Linguagem Natural e Machine Learning na Categorização de Notícias

José Carlos Lopes Varanda

Dissertação

Mestrado em Engenharia Informática – Sistemas de Informação

Trabalho efetuado sob a orientação de
Professora Doutora Ana Cristina Wanzeller Guedes de Lacerda
Professora Doutora Joana Rita da Silva Fialho

Março de 2026

AGRADECIMENTOS

Esta dissertação encerra uma etapa desafiante e enriquecedora. Marca o fim de um percurso de crescimento, não só académico, mas também pessoal. A concretização deste trabalho só foi possível graças ao apoio de figuras fundamentais, a quem manifesto a minha profunda gratidão.

Em primeiro lugar, agradeço sinceramente às minhas orientadoras, Professora Doutora Ana Cristina Wanzeller Guedes de Lacerda e Professora Doutora Joana Rita da Silva Fialho. Agradeço a disponibilidade, a partilha de conhecimento e a orientação segura, presentes em todas as fases críticas deste projeto.

Aos docentes do Mestrado em Engenharia Informática, agradeço o estímulo intelectual e os contributos científicos. As suas lições foram essenciais para ampliar horizontes. Consolidaram as competências necessárias em Processamento de Linguagem Natural e em *Machine Learning*.

Aos meus colegas de curso, agradeço o espírito de entreajuda e as discussões construtivas. O vosso apoio tornou a jornada mais leve e os desafios, em termos de motivação e produtividade, essenciais para a conclusão deste documento.

Por fim, agradeço profundamente à minha família. O vosso apoio incondicional, a paciência e o incentivo foram o alicerce essencial para enfrentar este objetivo com determinação e serenidade.

RESUMO

O crescimento exponencial de conteúdos digitais, especialmente notícias, artigos e publicações nas redes sociais, tem gerado grandes volumes de informação não estruturada, tornando a sua organização e análise progressivamente mais complexas. Neste contexto, o presente trabalho contribui para a simplificação e a otimização da classificação automática de notícias em português, por meio do desenvolvimento e da validação de abordagens de categorização que conciliem o elevado desempenho com a eficiência computacional.

Para a recolha e análise de dados, realizou-se uma revisão da literatura, seguida da seleção de algoritmos e ferramentas adequadas, da análise exploratória dos dados e da implementação de vários modelos de classificação. Estas etapas visaram comparar criticamente metodologias tradicionais de *Machine Learning* e de Mineração de Texto com modelos recentes de Processamento de Linguagem Natural, incluindo arquiteturas baseadas em *Transformers*, nomeadamente o BERT, avaliando simultaneamente o desempenho e o custo computacional associados.

Os resultados obtidos indicam que os modelos mais avançados de Processamento de Linguagem Natural, ao apresentarem uma maior capacidade de compreensão do contexto textual, alcançam um desempenho superior, embora impliquem um maior consumo de recursos computacionais. Em contrapartida, os modelos mais leves, baseados em representações otimizadas do texto, revelam um compromisso mais equilibrado entre a precisão e a eficiência computacional, mostrando-se adequados para cenários com restrições de *hardware*.

O trabalho experimental desenvolvido demonstra, deste modo, a viabilidade da implementação de um modeloe experimental funcional de categorização automática de notícias em língua portuguesa, capaz de processar grandes volumes de texto de forma eficaz, assegurando simultaneamente o controlo da complexidade computacional.

Em suma, este estudo evidencia que é possível conciliar o desempenho elevado com a eficiência computacional, oferecendo contributos relevantes para o desenvolvimento de modelos de classificação automática de notícias cada vez mais eficazes.

Palavras-chave: BERT; *Machine Learning*; Classificação; Categorização de Notícias; Mineração de Texto; Inteligência Artificial.

ABSTRACT

The exponential growth of digital content, particularly news articles, written publications, and social media posts, has led to large volumes of unstructured information, making its organization and analysis increasingly complex. In this context, this work contributes to the simplification and optimization of automatic news classification in the Portuguese language through the development and validation of categorization approaches that reconcile high performance with computational efficiency.

For data collection and analysis, a literature review was conducted, followed by the selection of appropriate algorithms and tools, exploratory data analysis, and the implementation of several classification models. These stages aimed to critically compare traditional Machine Learning and Text Mining methodologies with recent Natural Language Processing models, including Transformer-based architectures, namely BERT, while simultaneously evaluating performance and associated computational costs.

The results indicate that more advanced Natural Language Processing models, due to their greater ability to capture textual context, achieve superior performance, albeit at the expense of increased computational resource consumption. In contrast, lighter models based on optimized text representations exhibit a more balanced trade-off between accuracy and computational efficiency, making them suitable for scenarios with hardware constraints.

The developed prototype thus demonstrates the feasibility of implementing a functional automatic news categorization system in the Portuguese language, capable of efficiently processing large volumes of text while ensuring control over computational complexity.

In summary, this study shows that it is possible to reconcile high performance with computational efficiency, providing relevant contributions to the development of increasingly effective automatic news classification systems.

Keywords: BERT; Machine Learning; Classification; News Categorization; Text Mining; Artificial Intelligence.

ÍNDICE GERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

ÍNDICE DE EQUAÇÕES

LISTA DE SIGLAS

BERT – *Bidirectional Encoder Representations from Transformers*

Bi-LSTM – *Bidirectional Long Short-Term Memory*

BoW– *Bag-of-Words*

CNN – *Convolutional Neural Network*

CRISP-DM – *Cross Industry Standard Process for Data Mining*

DL – *Deep Learning*/Aprendizagem Profunda

GPU – *Graphics Processing Unit*

I&D – *Investigação e Desenvolvimento*

LSTM – *Long Short-Term Memory*

ML – *Machine Learning*

MLM – *Masked Language Modeling*

NLP / PLN – *Natural Language Processing* / Processamento de Linguagem Natural

NSP– *Next Sentence Prediction*

RNN – *Recurrent Neural Network*

SVM – *Support Vector Machines*

TF-IDF – *Term Frequency–Inverse Document Frequency*

TPU – *Tensor Processing Unit*

1. INTRODUÇÃO

Este capítulo apresenta os principais tópicos que orientam o desenvolvimento do presente trabalho, incluindo a contextualização do estudo, a motivação subjacente, a definição do problema, os objetivos propostos e os principais resultados obtidos. Adicionalmente, é apresentada a estrutura do documento, bem como o plano de trabalhos e a respetiva calendarização.

1.1. Enquadramento

A sociedade contemporânea encontra-se profundamente marcada pela expansão contínua da informação digital, em particular no que respeita a dados de natureza textual. No domínio das notícias, este crescimento assume relevância, impulsionado pela proliferação de plataformas digitais, redes sociais, portais informativos e diversos serviços *online*, que produzem grandes volumes de conteúdos de forma contínua. Esta multiplicação de fontes e formatos contribui para o aumento da complexidade associada à organização, gestão e extração de conhecimento a partir de dados predominantemente não estruturados, exigindo a adoção de soluções tecnológicas cada vez mais eficazes (Zhang et al., 2022).

Neste contexto, a classificação automática de notícias assume um papel fundamental, ao permitir a atribuição sistemática de categorias pré-definidas a textos noticiosos. Esta tarefa possibilita a organização eficiente de grandes coleções de documentos, a identificação de tópicos dominantes e o acesso rápido à informação relevante (Li et al., 2022; Mahajan, S., & Ingle, D. R., 2021). Para além destas funcionalidades, a classificação automática de notícias é utilizada em sistemas de recomendação personalizada, na filtragem de conteúdos indesejados e na análise de tendências em tempo real, constituindo um instrumento essencial para a gestão inteligente da informação e para o apoio à tomada de decisões em diversos contextos (Zemlyanskiy et. al., 2021; Kumari et al., 2023).

A extração do conhecimento a partir de conteúdos noticiosos torna-se ainda mais relevante em cenários de *Big Data*, caracterizados pela elevada quantidade, diversidade e natureza não estruturada da informação textual, nos quais as abordagens tradicionais de análise se revelam insuficientes, exigindo a adoção de técnicas avançadas de Mineração de Texto, Processamento de Linguagem Natural (PLN) e métodos de *Machine Learning* (ML) para identificar padrões e extrair informação relevante de grandes repositórios textuais.

Nos últimos anos, o progresso tecnológico tem impulsionado o surgimento de modelos de Aprendizagem Profunda (*Deep Learning* - DL) que transformaram significativamente o panorama do PLN (Tay et al., 2022). Entre estes modelos, destaca-se o *Bidirectional Encoder Representations from Transformers* (BERT), desenvolvido pela *Google*, cuja arquitetura baseada em transformadores

bidirecionais constitui um avanço na modelação das relações semânticas entre palavras, ao considerar simultaneamente o contexto anterior e posterior de cada termo (Minaee et al., 2021; Schick et al., 2021). Comparativamente às abordagens tradicionais, como os *word embeddings* estáticos, o BERT tem demonstrado melhorias significativas em diversas tarefas de classificação textual (Lee et al., 2020). A sua capacidade de adaptação a diferentes línguas e domínios reforça o seu valor como ferramenta versátil para a categorização automática de notícias (Zhang et al., 2022).

Adicionalmente, destacam-se os seguintes modelos: o *RoBERTa* (Minaee et al., 2021), o *DeBERTa* (He et al., 2020) e o GPT-3 (Brown et al., 2020), que permitiram expandir significativamente as capacidades das arquiteturas baseadas em transformadores, por meio da introdução de melhorias nos processos de treino e da obtenção de resultados mais consistentes e precisos em diversas tarefas de PLN.

1.2. Motivação

Apesar dos progressos alcançados no domínio da PLN e da crescente capacidade das técnicas de DL para tratar grandes volumes de dados textuais, ainda persistem limitações relativamente ao contexto da classificação automática de notícias. Em particular, a literatura científica revela uma escassez de estudos que realizem comparações sistemáticas, consistentes e empiricamente fundamentadas entre os métodos tradicionais de PLN e as abordagens mais recentes baseadas em representações distribuídas do texto, como o *Word2Vec*, o *GloVe* e o *FastText*, associadas a arquiteturas de redes neuronais convolucionais (*Convolutional Neural Networks* - CNN). Esta lacuna torna-se evidente no domínio da categorização automática de notícias em língua portuguesa, num contexto ainda pouco explorado em comparação com outros idiomas mais dominantes, como, por exemplo, o inglês ou o espanhol.

A ausência de análises comparativas sólidas dificulta a identificação da arquitetura mais adequada para esta tarefa, limitando a tomada de decisões tanto ao nível da investigação científica como na implementação de soluções práticas. Acresce, ainda, que muitas das abordagens mais avançadas apresentam elevados requisitos computacionais ou dependem de ferramentas pouco acessíveis, o que compromete a sua integração em sistemas operacionais utilizados em contextos jornalísticos e empresariais.

Neste enquadramento, o presente trabalho tem como principal motivação colmatar as limitações identificadas por meio da realização de um estudo comparativo aprofundado das principais abordagens

descritas na literatura para a classificação automática de notícias. A metodologia proposta visa, por um lado, analisar de forma crítica o desempenho das diferentes técnicas, avaliando o compromisso entre a precisão e a eficiência computacional, e, por outro lado, desenvolver uma implementação experimental que permita demonstrar a aplicação prática das soluções estudadas.

A relevância desta investigação reside, deste modo, no seu duplo contributo: no plano científico, pretende fornecer evidência empírica comparativa que apoie e oriente futuras investigações no domínio do PLN aplicado à língua portuguesa; e numa vertente aplicada, propõe uma abordagem e uma solução experimental, capaz de responder às necessidades atuais da sociedade da informação no que respeita à organização, classificação e análise eficiente de grandes volumes de conteúdos noticiosos.

1.3. Definição do Problema

A proliferação exponencial de conteúdos noticiosos no ecossistema digital impôs novos paradigmas à gestão e ao tratamento da informação. Atualmente, a convergência entre o volume massivo, a instantaneidade da difusão e a heterogeneidade das fontes cria barreiras significativas à organização estruturada, comprometendo a capacidade do utilizador de aceder, de forma eficiente e criteriosa, à informação que detém real relevância.

Atualmente, as plataformas jornalísticas, os sistemas de agregação de conteúdos e os mecanismos de recomendação enfrentam dificuldades significativas na categorização e filtragem eficazes de grandes volumes de informação textual. Esta limitação é agravada pelo facto de a maioria dos dados disponíveis *online* se encontrar em formatos não estruturados, com especial predomínio de conteúdos textuais, o que coloca desafios adicionais à sua gestão e análise automática. Investigações recentes corroboram que mais de 80% do ecossistema informativo digital apresenta este tipo de complexidade estrutural, o que acentua a necessidade de desenvolver soluções automatizadas mais robustas e eficientes (Hassani et al., 2020; Tay et al., 2022). Este cenário demonstra que a gestão manual ou baseada em modelos simplistas é insuficiente diante da escala e da natureza dos dados contemporâneos.

Os principais problemas identificados podem ser agrupados em três tópicos centrais:

1. Sobrecarga informacional e dispersão de fontes — O fluxo contínuo de notícias provenientes de múltiplas origens, como *websites*, blogues, plataformas digitais e redes sociais, gera grandes volumes de dados não estruturados. Esta diversidade de fontes e formatos cria dificuldades significativas na organização e na filtragem da informação. Sem mecanismos automáticos de classificação

suficientemente robustos, é fácil perder conteúdos relevantes, tornando difícil identificar rapidamente tópicos ou acontecimentos específicos. A literatura recente evidencia que lidar com a heterogeneidade de dados é um desafio constante e que a automatização da classificação é essencial para garantir o acesso rápido às notícias mais importantes do momento (Hassani et al., 2020; da Silva et al., 2020; Sokolova et al., 2018).

2. Limitações das abordagens tradicionais de categorização — As técnicas clássicas de ML e de Mineração de Texto, frequentemente baseadas em representações estáticas como os modelos de *Bag-of-Words* (BoW) ou o *Term Frequency–Inverse Document Frequency* (TF-IDF), apresentam limitações evidentes na compreensão do contexto e da semântica das palavras. Estas abordagens têm dificuldade em lidar com ambiguidades linguísticas, fenómenos de polissemia e a constante evolução do vocabulário jornalístico, o que compromete a precisão das classificações (da Silva et al., 2020; SANTANA et al., 2022). No que respeita à língua portuguesa, estudos recentes demonstram que métodos como o *Support Vector Machine* (SVM) ou o *Naive Bayes* ainda conseguem produzir resultados aceitáveis em cenários com recursos computacionais limitados. Contudo, estas técnicas revelam-se claramente inferiores às abordagens cujas arquiteturas se baseiam em transformadores (arquitetura *Transformer*), que capturam de forma mais profunda as relações contextuais do texto e proporcionam classificações mais consistentes e fiáveis (SANTANA et al., 2022; Schick et al., 2021).

3. Exigências computacionais dos modelos modernos — As abordagens modernas baseadas em DL e na arquitetura *Transformer*, como o BERT, introduziram melhorias significativas na compreensão do contexto linguístico e na capacidade de generalização dos modelos de classificação (SANTANA et al., 2022; Estevanell-Valladares et al., 2024; Tay et al., 2022). No entanto, estas vantagens implicam um custo computacional elevado, uma vez que estes modelos requerem grandes volumes de dados, tempos de treino prolongados e infraestruturas especializadas, como unidades de processamento gráfico (*Graphics Processing Unit* - GPU) ou unidades de processamento de tensor (*Tensor Processing Unit* - TPU). Por este motivo, a sua aplicação pode ser limitada em cenários com recursos computacionais limitados ou em sistemas que

exigem tempos de resposta reduzidos. Como alternativas viáveis a estas abordagens mais exigentes, destacam-se modelos intermédios que buscam equilibrar o desempenho com os requisitos computacionais. Entre estes modelos, estão incluídas as redes de memória de longo e curto prazo bidirecionais (*Bidirectional Long Short-Term Memory*, ou Bi-LSTM) com mecanismos de atenção, que permitem captar dependências contextuais relevantes no texto, bem como sistemas interpretáveis que combinam diferentes sinais linguísticos. Estas soluções apresentam-se como opções promissoras por oferecerem um melhor compromisso entre a qualidade da classificação e a eficiência computacional, tornando-se particularmente adequadas para cenários com recursos limitados ou necessidades de resposta em tempo útil (Estevanell-Valladares et al., 2024; Pereira et al., 2025).

1.4. Objetivos

O objetivo geral deste trabalho consiste em contribuir para a simplificação e a otimização da classificação de textos em língua portuguesa. Para o efeito, pretende-se desenvolver e validar um modelo de categorização automática de notícias em língua portuguesa que apresente precisão e eficiência computacional, de forma a simplificar a tarefa de classificação e viabilizar a implementação de um modelo experimental funcional em ambientes com recursos limitados.

O presente trabalho envolverá as seguintes atividades específicas, que abordam os problemas que foram identificados na secção anterior:

1. Realizar uma revisão sistemática e crítica da literatura, abrangendo trabalhos sobre a categorização de notícias no âmbito do PLN, com especial foco na língua portuguesa, e analisando os avanços nas técnicas de ML e DL, bem como nas arquiteturas *Transformers*.

2. Comparar criticamente diferentes metodologias de classificação, avaliando métodos tradicionais de ML e de mineração de texto face aos modelos de DL, incluindo as diversas variantes do BERT, de forma a identificar as respetivas vantagens, limitações e o potencial de adaptação à língua portuguesa.

3. Avaliar quantitativamente o desempenho e o custo computacional das abordagens selecionadas, definindo as métricas apropriadas (*Accuracy*, *F1-*

Score, tempo de treino e tempo de inferência) e quantificando o compromisso entre a precisão e os recursos computacionais necessários.

4. Desenvolver e avaliar uma implementação experimental otimizada de um modelo de categorização de notícias, capaz de operar eficientemente em contextos com restrições de *hardware*, recorrendo a modelos mais leves ou a versões otimizadas de *Transformers* que assegurem um desempenho elevado com recursos computacionais reduzidos.

1.5. Resultados Esperados

A presente dissertação pretende alcançar um conjunto de resultados distribuído em duas dimensões complementares: a produção de conhecimento científico e o desenvolvimento e validação experimental de abordagens de classificação automática. No plano científico, espera-se obter um conjunto sólido de evidências empíricas resultantes da comparação sistemática do desempenho de diferentes abordagens de classificação de texto em língua portuguesa. Esta análise será sustentada por meio de tabelas e gráficos que permitam avaliar detalhadamente a eficácia das metodologias estudadas, utilizando métricas amplamente reconhecidas, como a *Accuracy*, a *Precision*, o *Recall* e o *F1-Score*. Este processo deverá culminar na identificação do modelo que apresenta o melhor equilíbrio entre precisão e eficiência computacional no contexto específico da categorização automática de notícias, contribuindo, assim, para o avanço do conhecimento na área e para a disponibilização de dados comparativos que orientem investigações futuras.

Numa vertente mais aplicada, prevê-se o desenvolvimento de um modelo experimental de classificação automática de notícias que permitirá validar empiricamente as técnicas analisadas e demonstrar a viabilidade prática da solução proposta. Este modelo será concebido para processar textos completos de notícias, incluindo o título e o corpo textual, fornecidos como *strings* provenientes de fontes jornalísticas em língua portuguesa. A saída do modelo consistirá na atribuição de uma categoria temática pré-definida, num cenário de classificação multiclasse que inclui, por exemplo, áreas como a Política, Economia, Desporto ou Cultura. A classificação ao nível do documento constituirá o eixo central da abordagem, podendo, contudo, ser investigada uma estrutura hierárquica sempre que os dados disponibilizem informação adequada para a definição de subcategorias. A concretização deste modelo experimental permitirá não só materializar os objetivos definidos, como também lançar as bases necessárias para futuras aplicações práticas em sistemas de gestão, de agregação ou de recomendação de conteúdo noticioso.

1.6. Plano de Trabalhos

O plano de trabalhos delineado para a presente dissertação define uma abordagem estruturada para a investigação e a aplicação das técnicas de ML e de Mineração de Texto. Nesta secção, apresenta-se o plano de trabalhos, que detalha as principais etapas a realizar ao longo do desenvolvimento do estudo. O esquema que orientará o desenvolvimento do presente trabalho pode ser consultado na Tabela 1.

Tabela 1 - Plano de Trabalhos

Tarefa	Título	Descrição
T1	Pesquisa Bibliográfica de Metodologias de Investigação	Pesquisa bibliográfica de metodologias de investigação a usar no trabalho a desenvolver, bem como de orientações para condução sistemática de processos de descoberta de conhecimento.
T2	Levantamento de Abordagens e Tecnologias	Levantamento de abordagens, modelos, arquiteturas, componentes, ferramentas, técnicas e algoritmos utilizados no processamento e classificação de notícias visando a sua caracterização, bem como de avanços e direções futuras promissoras.
T3	Estudo Aprofundado de Técnicas e Aplicações	Estudo aprofundado da aplicação de ML, Mineração de Texto, BERT, entre outros mecanismos, sistematizando formas que poderão proporcionar contributos para a classificação de notícias e principais tipos de aplicações práticas.
T4	Levantamento e Análise de Ferramentas, Bibliotecas e Algoritmos Disponíveis	Levantamento de ferramentas, bibliotecas e algoritmos disponíveis.
T5	Implementação Experimental e Avaliação Comparativa de Modelos	Estudo detalhado de ferramentas, bibliotecas e algoritmos para a categorização de notícias. Inclui a comparação prática de técnicas como BERT, ML e Mineração de Texto, avaliando as suas capacidades e limitações através de testes em conjuntos de dados reais, visando validar a eficácia das abordagens selecionadas.
T6	Escrita de dissertação e artigos.	Redação da dissertação para documentar o trabalho realizado sobre a utilização das várias abordagens na Categorização de Notícias.

As tarefas descritas foram executadas conforme a calendarização apresentada na Figura 1.

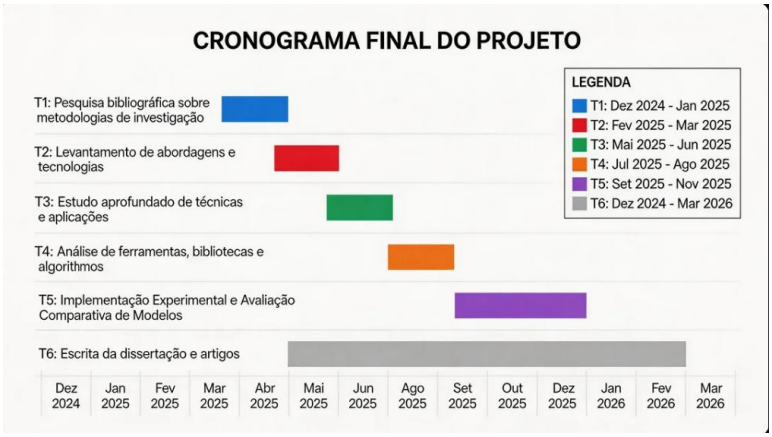

Figura 1 - Plano de Escalonamento das Tarefas

1.7. Estrutura da Tese

A presente dissertação encontra-se organizada em seis capítulos, estruturados de forma lógica e progressiva, o que permite uma compreensão integrada do problema em estudo, das metodologias adotadas, do trabalho experimental desenvolvido e dos resultados obtidos.

O Capítulo 1 apresenta o enquadramento geral do tema, contextualizando a problemática da categorização automática de notícias no âmbito do PLN e do ML. Neste capítulo são descritos a motivação do estudo, a definição do problema de investigação, os objetivos gerais e específicos, bem como os resultados esperados. Adicionalmente, são apresentados o plano de trabalhos que orientou o desenvolvimento da dissertação e a estrutura global do documento.

O Capítulo 2 é dedicado à revisão sistemática da literatura científica relevante. Neste capítulo são abordados os conceitos fundamentais de PLN e de ML aplicados à análise de texto, bem como a evolução das metodologias de classificação automática, desde os modelos clássicos até às abordagens baseadas em DL. São analisadas as arquiteturas sequenciais, a arquitetura *Transformer*, os modelos pré-treinados e as técnicas de transferência de aprendizagem, com especial foco nos modelos baseados em BERT e nas suas variantes. O capítulo inclui ainda uma análise de trabalhos relacionados, identificando lacunas e oportunidades de investigação.

O Capítulo 3 descreve o ecossistema tecnológico que sustentou o desenvolvimento do trabalho experimental. São apresentadas e analisadas as ferramentas, bibliotecas e *frameworks* utilizadas nas diferentes fases do estudo, incluindo soluções de pré-processamento linguístico, modelos clássicos de ML e DL, e modelos baseados em *Transformers*. O capítulo aborda igualmente ferramentas de interpretabilidade e de *deploy*, justificando as opções tecnológicas adotadas.

O Capítulo 4 detalha a abordagem adotada ao longo da investigação. Neste capítulo, apresenta-se a estrutura de investigação e desenvolvimento adotada, com base na *framework* CRISP-DM, bem como os requisitos funcionais e não funcionais do modelo experimental desenvolvido. São igualmente definidos os critérios de sucesso do trabalho experimental e as métricas de avaliação utilizadas para validar os modelos de classificação.

O Capítulo 5 descreve o desenvolvimento prático do modelo de categorização automática de notícias. Neste capítulo são apresentados o conjunto de dados utilizado, o *pipeline* de preparação e uso dos dados, bem como os processos de implementação, treino e avaliação dos modelos. São detalhadas as diferentes abordagens testadas, incluindo os modelos clássicos de ML, os modelos de DL e os modelos baseados em *Transformers*, estabelecendo as bases para a análise comparativa dos resultados.

O Capítulo 6 apresenta os resultados do trabalho experimental e procede a uma discussão aprofundada sobre a eficácia dos modelos testados. Além da avaliação baseada em métricas quantitativas, ponderam-se os requisitos computacionais de cada solução, estabelecendo-se um contraste rigoroso entre as metodologias. Esta reflexão crítica permite não só evidenciar os contributos do estudo, mas também clarificar o seu impacto e a sua aplicação em contextos reais.

Por fim, o Capítulo 7 sintetiza os principais contributos da dissertação e apresenta as conclusões. São identificadas as limitações durante o desenvolvimento do trabalho e propostas diretrizes para investigações futuras, com vista à melhoria das soluções de categorização automática de notícias em língua portuguesa.

2. ESTADO DA ARTE

O presente capítulo apresenta o estado da arte na categorização automática de textos, com ênfase nas abordagens de PLN e de ML aplicadas à classificação de notícias. São examinados os principais conceitos, metodologias e avanços tecnológicos descritos na literatura, como os métodos baseados em *Naive Bayes* (Mccallum & Nigam, 1998), SVM (Joachims, 1998) e, mais recentemente, as redes neuronais profundas para modelação de linguagem (Devlin et al., 2019). Desta forma, são abordadas contribuições que abrangem desde métodos tradicionais até abordagens baseadas em DL. Este panorama contextualiza as opções técnicas adotadas nesta dissertação, destacando os desafios específicos do processamento de texto em língua portuguesa, como a escassez de grandes volumes de dados anotados, a complexidade morfológica e a diversidade regional do idioma.

2.1. Enquadramento Teórico

Esta secção apresenta e fundamenta as metodologias adotadas na dissertação, com ênfase no PLN e no ML (Alloghani et al., 2020). Inicialmente, são detalhados os conceitos de PLN e o processamento de dados textuais, incluindo a normalização, a lematização e a vetorização. Em seguida, discutem-se os princípios de ML, os tipos de aprendizagem e os modelos de classificação. Por fim, definem-se as métricas de avaliação, como a *accuracy*, a precisão, o *recall* e o *F1-Score*, estabelecendo o quadro conceptual necessário para a análise das abordagens tecnológicas descritas neste documento.

2.1.1. Conceitos Fundamentais de Processamento de Linguagem Natural

No contexto da computação moderna, o PLN refere-se à aplicação de métodos estatísticos e de modelos de aprendizagem profunda para resolver tarefas linguísticas. O principal objetivo do PLN é transformar a complexidade e a ambiguidade inerentes à linguagem humana em representações estruturadas e processáveis por sistemas computacionais. Esta abordagem viabiliza aplicações como tradução automática, análise de sentimento, sistemas de diálogo e classificação de texto (Kumari et al., 2023).

O PLN funciona como uma ponte tecnológica entre a linguagem humana e os sistemas computacionais, permitindo a conversão de dados textuais não estruturados em representações adequadas para análise automática. Esta transformação é fundamental para a extração de conhecimento e a automatização de tarefas relacionadas com a organização e com a interpretação de grandes volumes de informação textual.

Para que os textos sejam utilizados por modelos de aprendizagem automática, é necessário submetê-los a etapas de preparação e de normalização linguística. Estas operações reduzem o ruído, uniformizam as variações superficiais da escrita e facilitam a identificação de padrões relevantes pelos algoritmos de classificação. Por exemplo, a remoção de *stopwords*, como é o caso dos artigos e das preposições, é uma etapa de pré-processamento que elimina palavras consideradas irrelevantes para a análise, contribuindo para a eficiência do modelo. Entre outras etapas, destacam-se a segmentação textual e a normalização morfológica, que transformam o texto bruto em unidades adequadas ao processamento computacional.

A evolução do PLN acompanha o progresso tecnológico das últimas décadas, especialmente o aumento da capacidade de processamento computacional e a maior disponibilidade de grandes volumes de dados digitais. Inicialmente, este campo era sustentado por abordagens baseadas em regras linguísticas elaboradas manualmente por especialistas. Posteriormente, métodos estatísticos foram adotados, permitindo a identificação e modelação de padrões linguísticos diretamente a partir dos dados.

Recentemente, a adoção de técnicas de DL impulsionou avanços significativos no desempenho dos modelos de PLN, permitindo capturar relações semânticas complexas e aumentar a precisão em diversas tarefas linguísticas. Um exemplo prático destes avanços é o surgimento de modelos de linguagem de grande porte, como o *ChatGPT*, que demonstram elevado desempenho em tarefas de compreensão e de geração de texto natural. Apesar destes avanços, permanecem desafios importantes, como a necessidade de grandes volumes de dados anotados, o alto custo computacional e a baixa interpretabilidade de muitos modelos, frequentemente considerados modelos de decisão pouco transparentes (Dai et al., 2021).

Em conclusão, o desenvolvimento contínuo do PLN depende da superação de desafios relacionados à transparência dos modelos e à disponibilidade de dados qualificados, sendo estes os aspetos fundamentais para o progresso sustentável da área.

2.1.2. Fundamentos de *Machine Learning* para Análise de Texto

O ML desempenha um papel central na execução de tarefas de PLN, ao proporcionar elevados níveis de inteligência, funcionalidade e capacidade preditiva. Esta tecnologia fundamenta-se na construção de modelos matemáticos que permitem ao modelo aprender de forma autónoma. Em vez de depender de regras rígidas previamente programadas, o algoritmo identifica padrões, relações e regras de forma independente. Esta capacidade revela-se particularmente relevante para a

categorização de notícias, pois influencia diretamente a eficácia com que o modelo classifica e organiza o fluxo contínuo de informação digital gerado em larga escala e proveniente de múltiplas fontes (Zemlyanskiy et al., 2021; Kumari et al., 2023).

A escolha da metodologia de ML depende do tipo de dados disponíveis e do objetivo da tarefa. Tradicionalmente, distinguem-se dois paradigmas principais: a Aprendizagem Supervisionada e a Aprendizagem Não Supervisionada (Alloghani et al., 2020).

Em tarefas de categorização, como a classificação de notícias, a Aprendizagem Supervisionada é geralmente priorizada, pois requer o treino do modelo com um conjunto de dados rotulados, em que cada exemplo possui uma resposta correta previamente definida. No contexto da classificação de notícias, o propósito do algoritmo supervisionado é aprender a relação entre a representação numérica do texto, gerada pelo PLN, e os respetivos rótulos de saída. O processo procura, em última instância, minimizar o erro de previsão, assegurando a generalização e a correta classificação de novos documentos (Zangari et al., 2023). Neste cenário, a classificação desempenha um papel preponderante ao atribuir categorias predefinidas a segmentos de texto (Mahajan, S., & Ingle, D. R., 2021). Este procedimento articula o PLN com modelos preditivos de forma eficaz: enquanto o primeiro converte o conteúdo textual em representações numéricas (vetorização), o segundo utiliza os dados para determinar a tipologia final da notícia (Li et al., 2022).

Por outro lado, a Aprendizagem Não Supervisionada é empregada em contextos em que os dados não possuem rótulos definidos. A Aprendizagem Não Supervisionada concentra-se na identificação de padrões intrínsecos e de estruturas ocultas em conjuntos de dados, sem orientação externa. Estas técnicas são especialmente úteis em tarefas exploratórias, como o agrupamento de documentos semelhantes (*clustering*) e a identificação dos principais temas (*topic modeling*) presentes no corpo das notícias (Tran et al., 2021).

Apesar da eficácia das abordagens tradicionais de ML que dependem fortemente da extração manual de características e geralmente apresentam limitações ao capturar relações complexas em dados textuais, tem conduzido à adoção progressiva de metodologias baseadas em DL, que serão discutidas na secção seguinte (Tay et al., 2022). Enquanto as abordagens tradicionais tendem a ser mais interpretáveis e requerem menos dados para treino, o DL destaca-se por permitir a modelação de dependências de longo alcance e de relações semânticas complexas, facilitando a compreensão de nuances contextuais e de ambiguidade lexical presentes na linguagem natural.

2.1.3. Modelos Clássicos de *Machine Learning* para a Classificação de Texto

Antes da adoção generalizada de abordagens baseadas em DL, a classificação automática de texto era predominantemente realizada por meio de modelos clássicos de ML. Estes modelos permanecem amplamente utilizados devido à sua eficiência computacional, robustez e interpretabilidade. A relevância prática desses métodos é especialmente notável em cenários com restrições de recursos computacionais ou com disponibilidade limitada de dados anotados (Palanivinayagam et al., 2023; Li et al., 2020).

A eficácia destes algoritmos depende significativamente da conversão do texto em representações numéricas. Uma das técnicas mais empregadas para este fim é o TF-IDF, que representa os documentos com base na importância relativa dos termos em uma coleção textual. Por exemplo, numa coleção composta pelos documentos "gato dorme no sofá" e "cão corre no parque", a palavra "no" aparece em ambos os textos, o que resulta num valor de TF-IDF baixo para este termo, enquanto palavras como "gato" ou "cão", que aparecem apenas num documento, recebem valores mais altos, destacando o seu potencial discriminativo. Embora não capture relações semânticas nem dependências contextuais entre palavras, o TF-IDF permanece uma representação simples e eficiente, frequentemente utilizada como referência em tarefas de classificação de texto (Das et al., 2023).

Entre os algoritmos clássicos mais utilizados, destaca-se o *Naive Bayes*, um modelo probabilístico que estima a probabilidade de um documento pertencer a uma categoria com base na distribuição dos termos presentes no texto. Apesar de se basear na hipótese simplificadora de independência entre características, o método mantém desempenho competitivo em diversas aplicações e apresenta elevada rapidez de treino e baixo custo computacional (Palanivinayagam et al., 2023).

O SVM é outro algoritmo aplicado, cuja eficácia na classificação textual decorre da capacidade de encontrar fronteiras de decisão que maximizam a separação entre as classes. Esta característica torna-o especialmente adequado para espaços de alta dimensionalidade, como os gerados por representações vetoriais de texto. No entanto, o SVM apresenta limitações, como a sensibilidade à escolha dos hiperparâmetros, além de exigir tempo computacional elevado para grandes volumes de dados ou para conjuntos de treinamento extensos. Adicionalmente, a sua capacidade de fornecer interpretações explícitas das decisões tomadas é inferior à de outros modelos mais transparentes, o que pode constituir uma restrição em determinadas aplicações. Apesar destas limitações, o SVM é frequentemente utilizado como modelo de referência na comparação com abordagens mais recentes (Wahba et al., 2022).

A Regressão Logística também constitui uma solução relevante para tarefas de categorização textual. Este modelo discriminativo estima diretamente a probabilidade de um documento pertencer a uma categoria específica a partir de combinações lineares das características extraídas do texto. Além de apresentar bom desempenho em diversos cenários, destaca-se pela interpretabilidade, o que permite analisar a contribuição relativa das diferentes características na decisão do modelo (Occhipinti et al., 2022).

O *Random Forest* representa uma abordagem que combina múltiplas árvores de decisão treinadas em subconjuntos aleatórios dos dados. Esta estratégia tende a aumentar a robustez do modelo e reduzir o risco de sobreajuste. No entanto, em contextos de classificação textual, a sua aplicação é menos frequente, pois as representações vetoriais de alta dimensionalidade podem limitar sua eficiência em comparação com modelos lineares (Palanivinayagam et al., 2023).

A Tabela 2 sintetiza as principais características, vantagens e limitações dos modelos clássicos discutidos, proporcionando uma visão comparativa estruturada das abordagens tradicionais de ML aplicadas à classificação de texto.

Tabela 2 – Comparação dos Principais Modelos Clássicos de ML para Classificação de Texto

Algoritmo	**Tipo de Modelo**	**Vantagens Principais**	**Limitações**
Naive Bayes	Probabilístico	Simples, rápido e de baixo custo computacional.	Baseia-se na hipótese de independência forte entre variáveis.
SVM	Margem máxima (linear)	Alta performance em espaços de alta dimensionalidade.	Muito sensível à escolha de hiperparâmetros (como o *kernel*).
Regressão Logística	Linear discriminativo	Extremamente interpretável e estatisticamente estável.	A premissa de linearidade limita a criação de fronteiras complexas.
Random Forest	*Ensemble* (árvores)	Muito robusto contra *overfitting* e lida bem com dados ruidosos.	Perde eficiência e velocidade ao lidar com dados muito densos.

De modo geral, os modelos clássicos apresentam um equilíbrio favorável entre desempenho e custo computacional, sendo frequentemente utilizados como referência para avaliar abordagens mais complexas. Por exemplo, em tarefas como a classificação de sentimentos em avaliações de produtos, modelos como o *Naive Bayes* ou a Regressão Logística frequentemente obtêm resultados satisfatórios com representações simples, como o TF-IDF. No entanto, a sua dependência de representações baseadas na frequência lexical limita a capacidade de captar relações contextuais profundas presentes na linguagem natural, o que motivou o desenvolvimento de métodos de aprendizagem profunda, discutidos na secção seguinte.

2.1.4. O Paradigma da Aprendizagem Profunda

O DL consolidou-se, na última década, como o paradigma dominante no ML, especialmente no PLN. Em vez de ser apenas uma extensão das abordagens clássicas, o DL representa uma transformação estrutural na modelação e interpretação da linguagem humana por sistemas computacionais, possibilitando representações mais ricas, contextualizadas e semanticamente informadas ao nível dos dados textuais (Zeng et al., 2022).

A principal distinção entre DL e os modelos tradicionais reside na capacidade de aprender automaticamente representações, dado que, enquanto os métodos clássicos dependem de representações baseadas em frequência lexical, os modelos de DL aprendem, de forma autónoma, as representações mais adequadas diretamente dos dados. Este processo ocorre de forma hierárquica, eliminando a necessidade de definição explícita de atributos relevantes e reduzindo significativamente a intervenção humana na modelação dos dados (Dai et al., 2021; Wang et al., 2020).

Esta capacidade resulta da arquitetura das redes neuronais profundas, compostas por múltiplas camadas ocultas. Cada camada transforma progressivamente a representação da informação, permitindo a captura de padrões de complexidade crescente. No PLN, a hierarquia promove uma transição gradual de padrões linguísticos superficiais para representações semânticas e contextuais mais abstratas. As camadas iniciais captam regularidades locais, enquanto as camadas mais profundas integram dependências de longo alcance e relações semânticas latentes entre palavras e expressões (Galassi et al., 2020).

A aprendizagem hierárquica é, portanto, o principal fator que diferencia o DL das abordagens tradicionais. Ao utilizar *embeddings* densos e distribuídos, estes modelos representam palavras e textos em espaços vetoriais contínuos, nos quais a proximidade geométrica reflete relações semânticas e sintáticas. Esta característica dos modelos de DL permite um tratamento mais eficaz de fenómenos linguísticos complexos, como a polissemia, a sinonímia e as dependências contextuais, que dificilmente são captados por representações estáticas baseadas apenas na frequência de ocorrência.

Apesar das vantagens, os modelos de DL apresentam algumas limitações, entre as quais o elevado custo computacional, a necessidade de grandes volumes de dados para treino eficaz e a menor interpretabilidade das decisões, o que traz desafios práticos à adoção em contextos operacionais com restrições de recursos e justifica, em muitos cenários, a coexistência de modelos clássicos e profundos.

Em síntese, o paradigma do DL fornece o enquadramento conceitual que sustenta os avanços recentes no PLN, ao oferecer mecanismos eficazes para a modelação contextual da linguagem. As

próximas secções examinam as principais arquiteturas de DL aplicadas à classificação automática de texto, analisando as suas características, potencialidades e limitações.

2.1.5. Evolução das Arquiteturas Sequenciais

A necessidade de que máquinas processem a linguagem natural como uma sequência interligada, em que o significado de cada elemento depende do contexto, impulsionou o desenvolvimento de diferentes arquiteturas especializadas de redes neuronais, como as RNN, as LSTM, as GRU e as CNN, que serão discutidas a seguir.

A Rede Neuronal Recorrente (RNN) constituiu o primeiro avanço na área de DL no tratamento de dados sequenciais, distinguindo-se fundamentalmente pela sua capacidade de memória. Ao contrário das redes tradicionais, que processam cada entrada de forma isolada e estanque, a arquitetura RNN analisa a informação de forma sequencial por meio de um estado oculto. Este estado atua como um resumo acumulado de toda a informação processada até ao momento, garantindo que o resultado em cada etapa seja influenciado tanto pela entrada atual como pelo contexto previamente memorizado (Min et al., 2023).

Entretanto, a RNN apresenta uma limitação significativa, especialmente em textos longos, conhecida como o problema do gradiente desvanecente (*vanishing gradient*). Na prática, à medida que a sequência se estende, a informação relevante capturada no início tende a se perder gradualmente, dificultando que o modelo aprenda dependências de longo prazo. Desta forma, a memória da RNN revela-se insuficiente para capturar o contexto global de documentos extensos (Hassani et al., 2020).

Para superar esta limitação, foi desenvolvida a arquitetura Long Short-Term Memory (LSTM), que melhorou a memória da RNN por meio de uma estrutura interna mais sofisticada, baseada numa célula de memória (*cell state*) e num conjunto de portas (*gates*) que regulam o fluxo de informação. A porta de esquecimento determina qual a informação que deve ser descartada; a porta de entrada, qual nova informação deve ser incorporada; e a porta de saída, qual informação deve ser transmitida ao próximo passo temporal. Esta gestão permitiu à LSTM reter o contexto relevante ao longo de sequências significativamente mais longas, consolidando-se como padrão de referência para tarefas de PLN que dependem da ordem e do significado sequenciais das palavras (Min et al., 2023). Em comparação, a *Gated Recurrent Unit* (GRU) representa uma evolução adicional na família das redes recorrentes, ao introduzir uma arquitetura mais simplificada do que a da LSTM. Enquanto a LSTM utiliza três portas (esquecimento, entrada e saída) para controlar o fluxo e a atualização da informação,

a GRU utiliza apenas duas: a porta de atualização (*update gate*) e a porta de reinício (*reset gate*). Esta diferença estrutural significa que, embora a LSTM apresente maior flexibilidade para lidar com padrões de dependência complexos, a GRU atinge desempenho semelhante com menos parâmetros e menor custo computacional. Desta forma, a GRU mantém a capacidade de reter informações relevantes ao longo de sequências extensas, apresentando desempenho competitivo em diversas tarefas de PLN e sendo particularmente adequada para cenários que exigem maior eficiência, ainda que, em certos contextos, a LSTM possa superar a GRU em tarefas que exigem a modelação de dependências mais complexas e de longo prazo (Min et al., 2023; Hassani et al., 2020).

Paralelamente ao desenvolvimento de arquiteturas recorrentes, foram propostas abordagens baseadas em Redes Neuronais Convolucionais (CNN) para o processamento de texto. Embora originalmente desenvolvidas para tarefas de visão computacional, as CNN demonstraram eficácia no PLN ao explorar a capacidade de identificar padrões locais, como os n-gramas e as combinações lexicais recorrentes. Ao aplicar operações de convolução unidimensional a sequências de representações vetoriais de palavras, estas redes extraem características discriminativas relevantes independentemente da posição no texto, configurando-se como uma alternativa eficiente às arquiteturas estritamente sequenciais (Sarikaya et al., 2024).

Apesar deste avanço, a LSTM apresentava uma limitação crítica de escalabilidade, uma vez que a sua natureza sequencial exigia o processamento da informação passo a passo, dificultando a paralelização eficiente em hardware moderno, como as GPU, para além de limitar a velocidade de treino e a capacidade de modelar relações contextuais complexas (Yin et al., 2023). Esta restrição tornou-se ainda mais evidente diante das exigências atuais do PLN, que requerem o processamento rápido e eficaz de grandes volumes de texto em aplicações como a tradução automática, a análise de sentimentos e as respostas automáticas em tempo real. Como resultado, a comunidade científica passou a procurar arquiteturas capazes de processar toda a sequência de texto simultaneamente, o que culminou no desenvolvimento da Arquitetura de Transformadores, cuja relevância atual decorre justamente da sua capacidade de superar as limitações das LSTM em cenários de grande escala e de complexidade contextual.

2.1.6. Arquitetura *Transformer* e o Mecanismo de Atenção

A arquitetura *Transformer* representa uma solução inovadora, projetada para superar limitações fundamentais das abordagens sequenciais anteriores, como a ineficiência no processamento paralelo e

a dificuldade em modelar dependências de longo alcance em textos extensos (Vaswani et al., 2017). Ao substituir completamente o paradigma recursivo das RNN e LSTM, o *Transformer* introduz um modelo baseado exclusivamente no mecanismo de Atenção, possibilitando a análise simultânea de todas as palavras de uma sequência e a atribuição dinâmica de pesos relacionais entre elas. O conceito central do *Transformer* está no mecanismo de Atenção Multi-Cabeça (*Multi-Head Attention*), que permite ao modelo calcular, em paralelo, múltiplas representações de um mesmo texto sob diferentes perspetivas ou subespaços de representação.

A adoção do mecanismo de atenção na arquitetura do *Transformer* resultou no surgimento dos chamados *embeddings* contextuais. Nestes modelos, a representação vetorial de cada palavra é construída dinamicamente a partir do contexto específico em que ocorre na sequência textual, em vez de ser fixa. Desta forma, uma mesma palavra pode assumir representações distintas conforme o seu enquadramento semântico, permitindo ao modelo lidar de forma mais eficaz diante de fenómenos como a polissemia e a ambiguidade linguísticas. Esta abordagem constitui um avanço conceitual significativo em relação aos *embeddings* estáticos, ao oferecer representações mais sensíveis ao contexto global do texto.

Tecnicamente, o processo fundamenta-se na projeção da sequência de entrada em três conjuntos de vetores: Consultas (*Queries*), Chaves (*Keys*) e Valores (*Values*). Para cada palavra, a sua consulta é comparada a todas as chaves da sequência, gerando uma matriz de pontuações de atenção que determina o grau de relevância de cada palavra no contexto atual. Após a normalização destas pontuações, calcula-se uma média ponderada dos valores, resultando em uma representação de saída em que a informação de todas as palavras é integrada de forma contextualizada. Esta operação é repetida por meio do mecanismo de *Multi-Head Attention*, cujos resultados são concatenados e projetados linearmente (Vaswani et al., 2017; Tay et al., 2023).

Esta abordagem confere ao *Transformer* capacidades críticas para o PLN, especialmente na classificação de texto, destacando-se pela paralelização em larga escala. Ao processar toda a sequência simultaneamente, a arquitetura aproveita de forma eficiente o *hardware* moderno, como GPU e TPU, reduzindo significativamente os tempos de treino em comparação com arquiteturas anteriores. Além disso, o modelo proporciona uma contextualização bidirecional e ilimitada, permitindo que cada palavra seja influenciada por todo o contexto circundante, sem a perda de informação observada em modelos unidirecionais. Esta característica é essencial para resolver ambiguidades e compreender relações semânticas complexas, o que é fundamental para distinguir categorias temáticas subtis.

Para além disso, a arquitetura apresenta uma hierarquia de abstração composta por uma pilha

de camadas idênticas, permitindo que as representações evoluam de padrões locais a abstrações de alto nível, como os tópicos discursivos (Devlin et al., 2019). Na prática, a arquitetura padrão consiste em dois blocos principais: o *Encoder*, responsável por processar e gerar representações contextuais da sequência de entrada, e o *Decoder*, que gera a sequência de saída com base nestas representações. Para tarefas de classificação de texto, como a categorização de notícias, normalmente utiliza-se apenas o *Encoder*.

Embora o *Transformer* se tenha consolidado como a base arquitetónica para a nova geração de modelos de linguagem, a sua aplicação não é isenta de desafios. A principal vulnerabilidade reside na complexidade computacional do mecanismo de atenção, que escala de forma dispendiosa perante sequências textuais longas. Na prática, estas restrições dificultam a implementação da arquitetura em ambientes com *hardware* limitado, evidenciando a necessidade de evoluir para soluções mais otimizadas que conciliem o elevado desempenho preditivo com a eficiência de processamento.

2.1.7. Modelos Pré-Treinados e *Transfer Learning*

A arquitetura *Transformer* representou um avanço significativo e impulsionou a adoção em larga escala dos paradigmas de pré-treino e de aprendizagem por transferência (*Transfer Learning*) no PLN. Estes conceitos superaram a limitação histórica de treinar modelos complexos do zero para cada nova tarefa, um processo exigente em termos de dados e recursos computacionais (Devlin et al., 2019). Com os *Transformers*, tornou-se viável aproveitar o conhecimento linguístico geral adquirido a partir de grandes volumes de texto não rotulado e transferi-lo de forma eficiente para tarefas específicas com conjuntos de dados menores.

O *Transfer Learning* fundamenta-se na reutilização do conhecimento generalista adquirido por um modelo ao ser exposto a grandes conjuntos de dados não rotulados. No contexto do PLN, um modelo *Transformer* é inicialmente pré-treinado com milhares de frases, o que lhe permite aprender padrões linguísticos fundamentais, como a sintaxe, a semântica, as relações contextuais e o conhecimento factual implícito (Liu et al., 2019). Posteriormente, esse modelo genérico e rico em conhecimento é adaptado (*fine-tuned*) para uma tarefa específica, como a classificação de notícias, por meio de treino adicional com um conjunto de dados rotulados, mais restrito e inerente ao contexto em análise.

O BERT, introduzido pelo *Google* em 2018, exemplifica essa abordagem paradigmática (Devlin et al., 2019), fundamentando o seu pré-treino em duas principais tarefas não supervisionadas. A primeira, denominada *Masked Language Modeling* (MLM), envolve a ocultação aleatória de

palavras na sequência de entrada para que o modelo as preveja, forçando-o a compreender o contexto bidirecional de cada termo. A segunda, *Next Sentence Prediction* (NSP), permite ao modelo aprender a identificar se duas frases são consecutivas no texto original, promovendo uma compreensão consolidada da coerência e do relacionamento discursivo.

Este processo resulta em representações linguísticas altamente contextualizadas. Para aplicar o BERT a tarefas de classificação, utiliza-se o *fine-tuning*. Nesta etapa, o modelo pré-treinado é especializado por meio da adição de uma camada de classificação simples, como uma camada linear *feed-forward*, localizada no topo do *Encoder*. O modelo completo é treinado novamente, desta vez com dados rotulados da tarefa-alvo, em que os parâmetros são ajustados para alinhar as representações às fronteiras de decisão das categorias específicas (Zeng et al., 2022). Este procedimento é substancialmente mais rápido e eficiente, proporcionando ganhos de desempenho significativos.

A captação de notícias em português é particularmente elevada, permitindo que modelos de grande escala sejam especializados para tarefas de categorização mesmo com conjuntos de dados anotados limitados, uma realidade comum em línguas com menos recursos digitais (Soares et al., 2019). No contexto da língua portuguesa, foram desenvolvidas adaptações específicas como o BERTimbau (Souza et al., 2020), pré-treinado em grandes volumes de texto em português. As variantes mais eficientes, como o *DistilBERT*, utilizam técnicas de destilação de conhecimento para comprimir o modelo original, oferecendo um equilíbrio quase ótimo entre desempenho e custo computacional, uma consideração crucial para implementações práticas (Sanh et al., 2019).

Em síntese, a integração da arquitetura *Transformer* com as metodologias de pré-treino e do *Transfer Learning* redefine o estado da arte no PLN. Esta sinergia fundamenta a criação de classificadores automáticos que são simultaneamente robustos, eficientes por meio da reutilização de modelos pré-treinados e adaptáveis por meio do *fine-tuning*, estabelecendo o principal alicerce metodológico para os modelos de categorização automática de notícias analisados na presente dissertação.

2.2. Classificação de Texto: Metodologias e Métricas de Avaliação

A classificação de texto constitui um componente central da Aprendizagem Supervisionada em modelos de PLN. Esta tarefa consiste em atribuir, de forma sistemática e automatizada, uma categoria pré-definida a um documento textual, visando identificar o rótulo mais apropriado para cada artigo de notícias (Sarker, 2021). O êxito deste processo depende principalmente da capacidade do algoritmo de

ML de generalizar padrões aprendidos, utilizando a representação vetorial do texto (*embedding*) para distinguir características semânticas e contextuais específicas de cada classe (Pang et al., 2002). O desenvolvimento de um modelo robusto exige, portanto, escolhas metodológicas criteriosas e uma avaliação de desempenho rigorosa.

No contexto da classificação automática de textos, é essencial compreender as diferentes abordagens estruturais, que se diferenciam pela forma como cada documento é atribuído às categorias. A abordagem tradicional, denominada por Classificação Multiclasse, pressupõe que cada documento seja associado a uma única categoria de um conjunto mutuamente exclusivo. Por exemplo, um artigo de notícias pode ser classificado como "Desporto" ou "Economia", mas não em ambas as categorias simultaneamente.

Em contraste, a Classificação Multirrótulo permite que um mesmo documento seja associado a múltiplos rótulos simultaneamente (Mamoshina et al., 2020). Esta abordagem é particularmente útil na categorização de notícias, pois um artigo sobre o impacto social de uma nova regulamentação governamental pode ser rotulado como "Política", "Sociedade" e "Direitos Humanos". A seleção entre estas metodologias depende da granularidade desejada, da complexidade do conjunto de dados e dos objetivos específicos da análise.

A avaliação do desempenho de um classificador não deve restringir-se à percentagem total de previsões corretas. Embora esta métrica seja intuitiva, pode ser insuficiente ou até enganosa, especialmente em cenários com conjuntos de dados significativamente desequilibrados, em que algumas categorias ocorrem com muito mais frequência do que outras (Han et al., 2021). Nestes casos, um modelo que atribui todas as instâncias à classe maioritária pode apresentar uma precisão artificialmente elevada, mas falha na identificação correta das classes minoritárias (Sarker, 2021). Para superar estas limitações, a avaliação de um modelo robusto e confiável utiliza métricas mais detalhadas e informativas, com base na Matriz de Confusão.

A Matriz de Confusão constitui uma ferramenta fundamental para desagregar os resultados da classificação em quatro categorias, refletindo o desempenho do modelo em relação à realidade dos dados. Estas categorias, apresentadas na Tabela 3, são essenciais para o cálculo das métricas de avaliação mais informativas.

Tabela 3 - Matriz de Confusão com Definições dos Resultados de Classificação

Classe Real / Previsão	Previsão: Positiva (P)	Previsão: Negativa (N)
Classe Real: Positiva (P)	Verdadeiro Positivo (VP) Acerto	Falso Negativo (FN) Erro Tipo II / Omissão
Classe Real: Negativa (N)	Falso Positivo (FP) Erro Tipo I / Falso Alarme	Verdadeiro Negativo (VN) Acerto

A matriz de confusão distingue quatro tipos de resultados. Os Verdadeiros Positivos (VP) e Verdadeiros Negativos (VN) representam classificações corretas, indicando, respetivamente, que instâncias das classes positiva e negativa foram identificadas corretamente. Em contraste, os Falsos Positivos (FP), ou erros do Tipo I, ocorrem quando o modelo atribui incorretamente a classe positiva a uma instância negativa, caracterizando um "falso alarme". Por exemplo, num teste de deteção de doenças, se o modelo indicar que uma pessoa saudável está doente, ocorre um Falso Positivo. Os Falsos Negativos (FN), ou erros do Tipo II, correspondem a omissões em que instâncias positivas são classificadas incorretamente como negativas. A análise destes quatro elementos proporciona uma avaliação estruturada do desempenho do modelo, evidenciando os compromissos entre diferentes tipos de falhas. Por exemplo, um modelo pode ser ajustado para minimizar os FP (aumentando a precisão), o que pode resultar em um aumento dos FN (reduzindo o *recall*), ou vice-versa.

Com base nestes elementos fundamentais, a matriz de confusão permite calcular as seguintes métricas:

1. A precisão avalia a qualidade das previsões positivas de um modelo, determinando a proporção de instâncias classificadas como positivas que, de facto, são corretas. Um valor elevado de precisão indica que o modelo gera poucos falsos positivos, o que é um fator crucial em contextos em que o custo de um falso positivo é significativo. A fórmula de cálculo é apresentada na Equação 1:

$$\text{Precisão} = \frac{VP}{VP + FP}$$

Equação 1 - Fórmula de cálculo da métrica de Precisão

2. O *Recall* mede a abrangência do modelo, ou seja, a sua capacidade de identificar todas as instâncias positivas reais. Representa a proporção de ocorrências da

classe positiva corretamente reconhecidas pelo modelo. Um valor elevado de *Recall* indica que o modelo é eficaz na deteção da maioria das instâncias relevantes, minimizando a ocorrência de Falsos Negativos. A sua fórmula de cálculo é apresentada na Equação 2:

$$Recall = \frac{VP}{VP + FN}$$

Equação 2 - Fórmula de cálculo da métrica Recall

3. O *F1-Score* é utilizado para combinar a Precisão e o *Recall* num único indicador, permitindo avaliar o equilíbrio entre as duas métricas. Esta abordagem é particularmente útil na análise de conjuntos de dados desequilibrados, pois evita a superestimação do desempenho do modelo. No entanto, uma limitação do *F1-Score* é que este não diferencia a importância de Falsos Positivos e Falsos Negativos, tratando ambos os tipos de erro de forma igual. Por exemplo, num sistema de deteção de fraudes bancárias, a priorização da redução de Falsos Negativos pode ser mais crítica do que a redução de Falsos Positivos, tornando o *F1-Score* menos representativo para estes cenários específicos. Para alcançar um valor elevado de *F1-Score*, o modelo deve demonstrar eficácia tanto na exatidão das previsões quanto na capacidade de identificar instâncias positivas. A fórmula de cálculo é apresentada na Equação 3:

$$\text{F1-Score} = 2 \times \frac{\text{Precisão} \times Recall}{\text{Precisão} + Recall}$$

Equação 3 - Fórmula de cálculo do F1-Score

Em cenários de classificação com múltiplas classes ou múltiplos rótulos, as métricas de Precisão, *Recall* e *F1-Score* devem ser agregadas criteriosamente para garantir uma avaliação representativa do desempenho. Para este propósito, são utilizadas duas estratégias principais de média: a Média Micro (*Micro Average*) e a Média Macro (*Macro Average*). A Média Micro é influenciada pelo número total de acertos, o que favorece as classes mais representadas na amostra. Em contraste, a Média Macro considera cada classe individualmente antes de calcular o valor final, assegurando que o desempenho das classes minoritárias tenha o mesmo peso no resultado global que o das classes maioritárias. Esta característica torna a Média Macro particularmente relevante para destacar o

desempenho do modelo em classes minoritárias, nas quais o erro pode ter consequências mais significativas. As seleções adequadas destas estratégias de agregação (Dai et al., 2021) são fundamentais para uma avaliação rigorosa do modelo experimental de classificação proposto e para uma comparação imparcial com o estado da arte na área.

2.3. Levantamento de Abordagens e Tecnologias

O presente levantamento tem como objetivo apresentar as principais abordagens tecnológicas aplicadas especificamente à classificação automática de texto, com ênfase nas abordagens que representam o estado da arte e que superam largamente a capacidade das abordagens estatísticas e vetoriais tradicionais (Dai et al., 2021).

Para além dos métodos clássicos baseados em representações vetoriais estáticas, as RNN, especialmente as variantes LSTM e GRU, tornaram-se referência no processamento de sequências textuais, possibilitando a modelação de dependências contextuais de médio e longo prazo (Min et al., 2023). Simultaneamente, as CNN demonstraram eficácia na identificação de padrões locais, como n-gramas e expressões características de categorias específicas (Sarikaya et al., 2024).

O avanço mais significativo na classificação textual ocorreu com a introdução da arquitetura *Transformer*, que utiliza o mecanismo de atenção para permitir a contextualização dinâmica e bidirecional de cada termo no texto. Os modelos baseados nesta arquitetura, como o BERT e suas variantes (Minaee et al., 2021), são reconhecidos como o estado da arte. A combinação de pré-treino extensivo e *fine-tuning* para tarefas específicas, como a classificação de notícias, confere a esses modelos uma capacidade excecional de compreensão linguística, o que explica seu desempenho superior em todas as etapas (Dai et al., 2021; Zeng et al., 2022). Portanto, as metodologias e tecnologias discutidas nesta dissertação fundamentam-se em arquiteturas avançadas, visando maximizar a precisão na categorização automática de notícias, considerando as particularidades e os desafios da língua portuguesa.

2.3.1. Técnicas de Representação de Texto

A eficácia da categorização de texto em PLN está intrinsecamente ligada à capacidade de converter a linguagem natural em estruturas vetoriais que preservem a sua riqueza semântica. Ao traduzir informação simbólica em coordenadas numéricas interpretáveis por modelos de Aprendizagem Automática, estabelece-se a base sobre a qual o classificador opera; ou seja, quanto mais fidedigna for esta representação, maior será o desempenho do modelo (Sarikaya et al., 2024).

As primeiras abordagens de representação textual utilizaram modelos estatísticos de contagem, caracterizados por vetores de alta dimensionalidade. Entre estes modelos, destacam-se o BoW e o TF-IDF. No BoW, cada documento é convertido em um vetor cuja dimensão corresponde ao número de palavras únicas na coleção, com a frequência de ocorrência de cada termo armazenada (Pang et al., 2002). Apesar da simplicidade e da eficiência, este método ignora completamente a ordem das palavras e não incorpora o conhecimento semântico. Palavras semanticamente próximas, como "viagem" e "jornada", são tratadas como independentes. O TF-IDF aprimora parcialmente o modelo ao ponderar a frequência das palavras pelo grau de raridade na coleção textual, atribuindo maior relevância a termos distintivos de cada documento. No entanto, a limitação de não captar relações semânticas entre palavras ainda persiste (Wang et al., 2020).

A verdadeira transformação na representação de texto ocorreu com a introdução das representações vetoriais de palavras, conhecidas como *word embeddings*. Estes modelos densos e contínuos superaram as limitações semânticas dos modelos de contagem, apoiando-se na hipótese distribucional, segundo a qual palavras que aparecem em contextos semelhantes tendem a ter significados semelhantes. Os *embeddings* são vetores de baixa dimensionalidade, aprendidos por meio da otimização de redes neuronais, que refletem diretamente relações semânticas na geometria do espaço vetorial. Palavras semanticamente relacionadas ficam próximas umas das outras, permitindo operações vetoriais de analogia, como no exemplo "rei – homem + mulher = rainha" (Bollegala & O'Neill, 2022; Pak et al., 2024).

Neste contexto, o *Word2Vec* e o *GloVe* destacam-se como os modelos mais relevantes. O *Word2Vec* baseia-se nas arquiteturas *Skip-Gram* e *Continuous Bag-of-Words* (CBOW) para aprender representações a partir da previsão do contexto ou da palavra central (Incitti et al., 2023; Pak et al., 2024). O *GloVe* diferencia-se no contexto do PLN ao combinar o paradigma preditivo com a análise das relações de proximidade entre palavras em escala global. Esta abordagem híbrida permite a criação de modelos de linguagem mais robustos, assegurando que os vetores numéricos reflitam padrões reais de uso da linguagem natural (Sarikaya et al., 2024). Diferentemente de métodos que consideram apenas contextos locais, o *GloVe* aproveita a estrutura estatística de todo o corpo textual, proporcionando uma representação semântica mais completa para as etapas subsequentes da classificação de texto.

O *FastText* introduz uma inovação relevante ao operar no nível de subpalavras, utilizando n-gramas de caracteres. A presente abordagem permite a geração de representações robustas para palavras raras, derivados morfológicos e vocábulos compostos, o que é especialmente vantajoso em idiomas com morfologia complexa, como o português. O *FastText* potencia a cobertura lexical e a

capacidade de generalização do modelo, revelando-se particularmente eficaz em dados com vocabulário heterogéneo ou linguagem informal (Bojanowski et al., 2017; Joulin et al., 2017; Park, 2019).

Os *embeddings* mencionados anteriormente são estáticos e, como tal, apesar de oferecerem avanços significativos em relação aos modelos de contagem, apresentam uma limitação fundamental: não capturam o significado contextual das palavras. Cada palavra é representada por um único vetor independente do contexto, o que inviabiliza a diferenciação de sentidos em casos de polissemia, como no exemplo de "banco", que pode denotar uma instituição financeira ou um assento. Apesar de modelos como *Word2Vec*, *GloVe* e *FastText* aproximarem semanticamente as palavras a partir de padrões globais ou morfológicos, todos falham em representar variações contextuais intrassentenciais ou intersentenciais na representação de variações semânticas que ocorrem no interior da frase ou na relação entre sentenças consecutivas, ressaltando a superioridade dos *embeddings* contextuais, que, por sua vez, constituem o núcleo da arquitetura *Transformer*, proporcionando representações refinadas que consideram os múltiplos sentidos de uma palavra conforme o contexto específico em que esta aparece (Dai et al., 2021).

Em suma, os *embeddings* estáticos estabeleceram os alicerces conceptuais e metodológicos que impulsionaram a ascensão do DL no domínio do PLN. Esta abordagem representou um marco evolutivo determinante, cujos princípios fundamentais continuam a influenciar o desenvolvimento tecnológico contemporâneo desta área.

2.3.2. Modelos Baseados em Redes Neuronais Recorrentes e Redes Neuronais Convolucionais

A aplicação de arquiteturas sequenciais, nomeadamente as RNN e as suas variantes LSTM e GRU, bem como as CNN, assume particular relevância na classificação automática de texto. Estas arquiteturas têm sido amplamente exploradas no domínio do PLN, demonstrando elevada eficácia no processamento de grandes volumes de informação textual. No contexto desta dissertação, destaca-se a sua aplicação na classificação e categorização automática de notícias provenientes de diferentes fontes jornalísticas.

No domínio do PLN, as arquiteturas recorrentes distinguem-se pela capacidade de modelar dependências sequenciais entre palavras, permitindo capturar o encadeamento semântico que se desenvolve ao longo de um documento. Em tarefas de classificação de notícias, esta característica revela-se particularmente relevante, uma vez que o significado global de um artigo depende

frequentemente de informação apresentada nas fases iniciais do texto e desenvolvida nos parágrafos seguintes. As variantes LSTM e GRU procuram superar limitações das RNN tradicionais na aprendizagem de dependências de longo alcance, permitindo preservar informação contextual relevante em sequências extensas e contribuindo para melhorar a qualidade das representações textuais utilizadas nos processos de classificação (Min et al., 2023; Khan et al., 2021).

Paralelamente, as redes CNN têm demonstrado elevada eficácia na identificação de padrões locais em dados textuais. Ao aplicarem operações de convolução sobre representações vetoriais das palavras, estas arquiteturas permitem identificar padrões lexicais e linguísticos relevantes independentemente da sua posição no texto. Esta capacidade de deteção de estruturas recorrentes, frequentemente associadas a n-gramas ou combinações lexicais características, contribui para a extração eficiente de indicadores temáticos em contextos jornalísticos (Sarikaya et al., 2024).

A utilização isolada ou combinada destas arquiteturas demonstra resultados superiores aos obtidos por modelos clássicos de aprendizagem automática em diversas tarefas de classificação textual e análise de linguagem natural, evidenciando o impacto do DL no processamento automático de grandes volumes de dados linguísticos (Liu et al., 2021). No contexto comparativo destas abordagens, observa-se que as arquiteturas recorrentes, incluindo as variantes LSTM e GRU, apresentam elevada eficácia na modelação de dependências sequenciais e contextuais ao longo do texto, permitindo captar relações semânticas que se desenvolvem progressivamente nas sequências linguísticas (Min et al., 2023). Em contraste, as CNN destacam-se sobretudo na identificação de padrões locais e combinações lexicais discriminativas, explorando estruturas linguísticas relevantes independentemente da posição textual (Singh et al., 2025).

Apesar destas vantagens individuais, ambas as abordagens apresentam limitações estruturais quando dependem de representações baseadas em *embeddings* estáticos, nos quais cada palavra é representada por um vetor fixo e invariável. Esta característica reduz a capacidade dos modelos para captarem variações contextuais e relações semânticas mais complexas presentes em diferentes contextos linguísticos (Wang, Ding & Han, 2024). Como consequência, modelos baseados exclusivamente em RNN e CNN podem revelar limitações na interpretação de fenómenos linguísticos como a polissemia ou a ambiguidade semântica.

Mesmo perante estas limitações, as arquiteturas baseadas em RNN, LSTM, GRU e CNN desempenham um papel fundamental na evolução do PLN aplicado à classificação automática de texto. Estas abordagens representam uma etapa intermédia relevante entre os modelos tradicionais baseados em características estatísticas e os modelos mais recentes baseados em representações contextuais mais

avançadas. Para além do seu valor conceptual e histórico no desenvolvimento do campo, continuam a ser frequentemente utilizadas como modelos de referência em estudos comparativos e em cenários que exigem equilíbrio entre desempenho e custos computacionais.

2.3.3. Representações Contextuais e a evolução para os *Transformers*

A principal limitação dos *embeddings* estáticos conduziu ao desenvolvimento de representações contextuais dinâmicas, que constituem um dos avanços mais significativos recentes no PLN. Estas representações estão na base da arquitetura *Transformer*, introduzida por Vaswani et al. (2017), que reformularam profundamente a forma como os modelos computacionais processam linguagem natural.

Ao contrário das abordagens anteriores, nas quais cada palavra era associada a um vetor fixo independentemente do contexto em que ocorria, os *embeddings* contextuais produzem representações específicas para cada ocorrência de uma palavra numa sequência textual. Assim, o significado de um termo passa a ser determinado pelas relações que estabelece com os restantes elementos do enunciado.

Esta capacidade é particularmente relevante para lidar com fenómenos de polissemia e ambiguidade semântica. Por exemplo, a palavra "célula" pode assumir significados distintos em expressões como "célula estaminal", no domínio da biologia, "célula terrorista", no contexto da segurança, ou "célula fotovoltaica", no domínio da energia. As representações contextuais permitem distinguir automaticamente estas interpretações, produzindo vetores diferentes consoante o contexto linguístico em que a palavra ocorre.

A arquitetura *Transformer* materializa este paradigma através do mecanismo de *self-attention*, que permite a cada elemento da sequência textual estabelecer relações com todos os outros elementos do texto, independentemente da distância entre eles. Este mecanismo calcula pesos de relevância que determinam a importância relativa de cada palavra na interpretação global da sequência, permitindo captar dependências semânticas e sintáticas de longo alcance de forma mais eficiente do que arquiteturas sequenciais tradicionais.

Entre os modelos baseados nesta arquitetura, destaca-se o BERT, proposto por Jacob Devlin et al. (2019). Este modelo introduziu uma abordagem de pré-treino bidirecional, permitindo que cada palavra seja contextualizada simultaneamente pelo texto que a precede e pelo que a sucede.

O processo de aprendizagem baseia-se na tarefa de MLM, na qual algumas palavras da sequência são ocultadas e o modelo aprende a prevê-las com base no contexto envolvente. Desta forma,

o sistema desenvolve representações linguísticas capazes de capturar relações semânticas e sintáticas complexas presentes na linguagem natural. Após o pré-treino, o modelo pode ser adaptado a tarefas específicas, como classificação de texto, através de um processo de *fine-tuning*.

No contexto da língua portuguesa, foram desenvolvidas variantes específicas destes modelos, como o *BERTimbau*, proposto por Felipe Nogueira, Rodrigo Nogueira e Roberto Lotufo (2020). Este modelo foi pré-treinado em grandes volumes de texto em português, permitindo capturar particularidades linguísticas da língua e alcançar melhorias significativas de desempenho em tarefas de classificação textual quando comparado com modelos multilingues genéricos.

Em síntese, a introdução das representações contextuais e das arquiteturas baseadas em *Transformers* representou uma mudança de paradigma no PLN. Estes modelos superaram diversas limitações das abordagens anteriores e estabeleceram as bases tecnológicas dos sistemas contemporâneos de análise textual, incluindo a classificação automática de notícias.

Neste sentido, a Tabela 4 apresenta um resumo comparativo das principais metodologias discutidas, sintetizando as suas características fundamentais, a capacidade de modelação contextual, os requisitos computacionais, bem como as principais vantagens e limitações. Esta sistematização permite uma visão integrada das soluções analisadas e facilita a compreensão das opções metodológicas consideradas no presente trabalho.

Tabela 4 – Resumo Comparativo das Principais Abordagens de Classificação de Texto

Abordagem	**Representação**	**Contexto**	**Custo**	**Vantagens**	**Limitações**
Clássicos	TF-IDF / BoW	Baixa	Baixo	Rápido e interpretável	Sem semântica
RNN	*Embeddings*	Média	Médio	Longas dependências	Lento (sequencial)
CNN	*Embeddings*	Média	Médio	Extração de n-gramas	Visão global limitada
Transformers	Contextual	Elevada	Elevado	Alta precisão	Custo de *hardware*
Híbrida	Contextual	Elevada	Médio	Equilíbrio Eficiência	Menor ajuste fino

2.4. Trabalhos Relacionados

Nos últimos anos, observou-se um aumento expressivo na investigação sobre a classificação automática de notícias e a deteção de desinformação, impulsionado pela rápida disseminação de conteúdos falsos ou manipulados em ambientes digitais. A literatura recente aponta para um consenso

quanto à relevância dos modelos baseados em *Transformers*, atualmente considerados no estado da arte. No entanto, a diversidade metodológica, a variedade de dados utilizados e os objetivos específicos de cada estudo tornam necessária uma análise detalhada e estruturada dos principais contributos científicos recentes, de forma a abranger as abordagens clássicas, as soluções multilingues e as soluções multimodais contemporâneas.

Os estudos iniciais concentraram-se em abordagens baseadas na aprendizagem automática clássica. Li et al. (2022) analisaram a categorização temática de notícias utilizando modelos como o *Naive Bayes* e o SVM, demonstrando desempenho satisfatório em conjuntos de dados bem estruturados e com categorias claramente definidas. Os resultados apresentados por Uysal e Gunal (2020) também são semelhantes, destacando a eficiência computacional e a interpretabilidade destas abordagens em cenários com recursos limitados. Kowsari et al. (2021) corroboram que modelos clássicos mantêm um equilíbrio entre desempenho e eficiência, sendo considerados adequados quando a interpretabilidade e o baixo custo de treino são fatores determinantes para esta pesquisa.

O advento do DL impulsionou diversas investigações sobre a aplicação de arquiteturas como as CNN e as RNN no domínio jornalístico. Estudos conduzidos por Yang et al. (2019) e Mohanta et al. (2020) evidenciaram a superioridade destes modelos face às abordagens clássicas, sobretudo pela sua aptidão para captar padrões sequenciais e dependências contextuais em documentos extensos ou com elevada variação lexical. Contudo, a literatura sublinha uma limitação crítica na integração de *embeddings* estáticos (como o *Word2Vec* ou o *GloVe*) nestas redes: a atribuição de uma representação vetorial única e imutável a cada vocábulo. Esta rigidez compromete a resolução de ambiguidades polissémicas e a gestão de contextos de longo alcance. Em contrapartida, as representações contextuais derivadas de *Transformers* superam tais restrições, uma vez que processam a envolvente dinâmica de cada ocorrência, o que potencializa a descodificação de nuances semânticas e de relações discursivas complexas.

A introdução da arquitetura *Transformer* e dos modelos de linguagem pré-treinados representou uma mudança conceptual significativa. No contexto da crise pandémica, Wang et al. (2021) investigaram como a Inteligência Artificial pode combater o fluxo inédito de desinformação, testando variantes do BERT em combinação com componentes de memória, como o Bi-LSTM, e com componentes de extração de padrões locais por meio de CNN. Os resultados indicaram que o modelo BERT capta nuances semânticas complexas não identificadas pelos métodos tradicionais. Os estudos comparativos sistemáticos, como o de Joy et al. (2022), analisam modelos modernos, como o BERT, a *RoBERTa*, o *XLNet* e o ALBERT, e demonstram que, embora todos lidem com abstrações linguísticas

complexas, o desempenho varia significativamente conforme o tipo de dados e o tamanho do conjunto de treino.

Singla et al. (2024) estabeleceram uma comparação entre as metodologias clássicas (SVM e *Naive Bayes*), as arquiteturas RNN e CNN e os modelos de linguagem pré-treinados (BERT e *mBERT*). A investigação concluiu que, perante padrões semânticos complexos, os *Transformers* demonstram uma robustez superior, especialmente na resolução de ambiguidades semânticas e na adaptação a novos contextos. Contudo, para além deste domínio técnico, o estudo sublinha a relevância da eficiência computacional, sugerindo que o elevado desempenho destas arquiteturas deve ser equilibrado com os requisitos de *hardware*.

No contexto da classificação multidimensional, Zemlyanskiy et al. (2021) investigaram a classificação multirrótulo de notícias em redes sociais, implementando uma arquitetura baseada em BERT otimizada para identificar categorias sobrepostas. Os resultados confirmaram a eficácia dos *Transformers* na deteção de matizes discursivas não captadas por metodologias convencionais. No entanto, os autores alertam para a escassez de repositórios de informação anotados com múltiplas categorias, o que constitui um obstáculo significativo ao desenvolvimento de modelos mais consistentes neste domínio.

Em abordagens multilingues, estudos como os de Wang et al. (2022) e Lample et al. (2019) exploraram o modelo XLM-R para a classificação de notícias em diferentes idiomas, demonstrando a robustez do *Transfer Learning* mesmo sem dados anotados na língua-alvo, o que é um aspeto relevante para línguas com poucos recursos, como é o caso da língua portuguesa. Pires et al. (2019) e Souza et al. (2020) analisaram a transferência cruzada entre línguas próximas, como espanhol, italiano e português, evidenciando que a proximidade morfológica influencia a eficácia da transferência e reforçando o potencial dos *Transformers* multilingues em contextos lusófonos.

A investigação recente evidencia um interesse crescente por abordagens multimodais que integram texto e imagem, considerando que muitas notícias falsas apresentam elementos visuais manipulados. Alonso-Bartolome e Segura-Bedmar (2021) utilizaram uma arquitetura que combina CNN para imagens e para texto, avaliando o modelo no conjunto de dados *Fakeddit*. A integração das modalidades resultou em melhorias significativas no desempenho, especialmente em categorias como a sátira e a manipulação fotográfica. Yan et al. (2024) exploraram o modelo CLIP, que associa texto e imagem por meio de aprendizagem contrastiva, implementando mecanismos de atenção cruzada para ajustar os pesos de cada modalidade.

No domínio da interpretabilidade e da transparência, Tabassoum et al. (2024) avançaram na explicabilidade de modelos *Transformer* ao combinar BERT e BiLSTM com o algoritmo SHAP, o que permite identificar palavras ou expressões que influenciam as decisões do modelo. Esta abordagem aumenta a transparência, que é um requisito essencial em contextos editoriais. Ying et al. (2021) propuseram um modelo multimodal avançado com fusão baseada em *cross-attention* entre a representação textual do BERT e a visual da CNN, superando as limitações das fusões tradicionais e demonstrando ganhos expressivos em precisão e robustez na classificação de notícias falsas.

Em síntese, a análise comparativa da literatura confirma uma progressão clara no desempenho dos modelos clássicos em relação às arquiteturas *Transformer*. Contudo, observa-se a ausência de estudos que realizem comparações sistemáticas e abrangentes entre estes paradigmas no contexto da língua portuguesa, lacuna esta que motiva e fundamenta o desenvolvimento da presente investigação.

2.4.1. *Datasets* Utilizados na Literatura

A seleção e análise de *datasets* são fundamentais para a pesquisa em classificação de notícias, pois a qualidade, o tamanho e a diversidade temática dos conjuntos de dados impactam diretamente o desempenho dos modelos e a validade dos resultados experimentais. A literatura recente destaca a predominância de recursos em língua inglesa, amplamente adotados devido ao seu volume, à anotação consistente e ao fácil acesso público. Em contrapartida, observa-se uma escassez significativa de conjuntos de textos equivalentes em português, o que constitui uma das principais limitações ao avanço da área no contexto lusófono.

O *dataset 20 Newsgroups* destaca-se entre os mais utilizados como referência histórica e comparativa. Apesar do equilíbrio temático, a natureza não profissional das mensagens limita a sua relevância para estudos jornalísticos, conforme apontam Kowsari et al. (2021) e Yang et al. (2019). No domínio específico das notícias, o *AG News* afirma-se como um paradigma de avaliação consolidado, sendo frequentemente utilizado para aferir o desempenho de arquiteturas que variam entre os métodos clássicos e os modelos *Transformer*. A sua relevância é corroborada por investigações de Mohanta et al. (2020) e Lee et al. (2020), que assinalam a fraca representatividade de outros géneros jornalísticos fora dos eixos principais do conjunto de dados.

O *dataset Reuters-21578* permanece em uso em estudos históricos e comparativos devido à sua estrutura consistente de anotações, embora a literatura destaque a sua desatualização em relação ao jornalismo contemporâneo (Uysal & Gunal, 2020; Baly et al., 2020). Simultaneamente, o aumento das

pesquisas sobre desinformação impulsionou o desenvolvimento de *datasets* multimodais, como o *Fakeddit*, que integra texto, imagens e metadados, possibilitando a avaliação de abordagens multimodais. Pesquisas como as de Alonso-Bartolome e Segura-Bedmar (2021) e de Yan et al. (2024) evidenciam ganhos significativos na combinação de texto e imagem, especialmente em situações de incongruência ou de manipulação visual.

No âmbito das *fake news* textuais, o LIAR continua sendo um recurso amplamente utilizado e tem sido incorporado a diversos estudos com modelos *Transformer* (Joy et al., 2022; Singla et al., 2024). No entanto, a sua forte dependência do contexto sociopolítico norte-americano restringe a generalização para outros contextos linguísticos e culturais.

No contexto da língua portuguesa, a oferta de conjuntos de textos é limitada e, em geral, direcionada ao português do Brasil. O *Fake.Br* é um dos recursos mais utilizados; no entanto, as suas particularidades linguísticas e culturais suscitam dúvidas quanto à transferência para o português europeu (da Silva, 2020). No cenário europeu, SANTANA (2022) apontam a ausência de um grande corpo de notícias como uma das principais barreiras ao desenvolvimento de modelos de classificação comparáveis aos disponíveis em outros idiomas.

Outro aspeto crítico, destacado na literatura, refere-se à desatualização das bases de dados e ao desequilíbrio entre as classes. Dogra et al. (2023) observaram que muitos *datasets* não refletem as mudanças recentes no discurso mediático, o que prejudica o desempenho dos modelos em temas emergentes. De modo semelhante, Baly et al. (2020) enfatizam que amostras com classes desiguais introduzem distorções no processo de aprendizagem, reduzindo a capacidade dos modelos de generalizar às categorias minoritárias. Este desafio é ainda mais pronunciado no contexto da língua portuguesa, devido à escassez de recursos de grande escala de forma abrangente e equilibrada.

Para além disso, observa-se que poucos estudos integram *datasets* de naturezas distintas, como os conjuntos clássicos, multimodais, temáticos e factuais, o que limita a realização de comparações abrangentes. Desta forma, apesar da disponibilidade de dados em inglês, persistem lacunas estruturais relevantes para o português europeu, tanto na diversidade temática quanto na dimensão e na qualidade das anotações.

Em síntese, a literatura evidencia tanto a diversidade de recursos disponíveis internacionalmente quanto as limitações significativas do português europeu. Estas lacunas ressaltam a necessidade de pesquisas que explorem modelos modernos aplicados a conjuntos de dados adaptados ao contexto linguístico do português, promovendo avanços concretos no ecossistema de PLN orientado

para a classificação automática de notícias.

2.4.2. Identificação de Lacunas e Oportunidades

A revisão da literatura revela diversas lacunas que justificam uma investigação adicional, nomeadamente o predomínio quase exclusivo de recursos e modelos desenvolvidos para a língua inglesa (Baly et al., 2020; Dogra et al., 2023). Apesar da crescente disponibilidade de modelos multilingues (Wang, Li & Zhang, 2022), a investigação dedicada à língua portuguesa permanece escassa, sobretudo no que diz respeito à criação de conjuntos de dados noticiosos extensos e devidamente anotados (SANTANA, 2022). Esta carência de recursos específicos restringe o desenvolvimento de soluções resilientes que considerem as particularidades culturais e as subtilezas linguísticas do idioma, dificultando a implementação de modelos plenamente adaptados à realidade nacional.

Outra lacuna relevante refere-se ao desequilíbrio dos *datasets* disponíveis. Diversos conjuntos de textos apresentam distribuição desigual entre classes temáticas ou baixa representatividade de tópicos específicos, o que compromete a capacidade de generalização dos modelos (da Silva, 2024). Embora os modelos *Transformer* demonstrem alguma resiliência a estas desigualdades, a escassez de *datasets* atualizados continua a dificultar avaliações rigorosas. A inexistência de normas transversais constitui, assim, um obstáculo à análise comparativa de estudos e à verificação sistemática dos métodos em contextos de elevada heterogeneidade.

Para além das questões linguísticas e estruturais dos dados, observa-se uma lacuna significativa em termos de eficiência computacional, uma vez que muitos estudos priorizam arquiteturas de grande porte, como o BERT *Large* ou o *RoBERTa*, com foco quase exclusivo na maximização da precisão (Joy et al., 2022; Singla et al., 2024). Contudo, estes modelos exigem elevados recursos computacionais, o que limita a sua aplicação em contextos com restrições de *hardware*, tais como redações jornalísticas, sistemas de monitorização ou plataformas de consulta em tempo real. Apesar de abordagens como o *DistilBERT* (Sanh et al., 2020) demonstrarem ser possível conciliar níveis competitivos de desempenho com uma menor carga de processamento, as estratégias que privilegiam explicitamente a sustentabilidade computacional permanecem pouco exploradas, subsistindo uma lacuna clara no equilíbrio entre desempenho e custo operacional.

A interpretabilidade constitui outro aspeto ainda pouco explorado na literatura. Apesar dos avanços recentes na área, a maioria dos modelos continua a operar como uma “caixa negra”,

dificultando a compreensão de seus processos de decisão. Esta limitação é particularmente problemática em aplicações jornalísticas e institucionais, nas quais é essencial justificar automaticamente a classificação de um artigo em determinada categoria (Ribeiro, Singh & Guestrin, 2016; Lundberg & Lee, 2017). Embora estudos recentes indiquem que técnicas de interpretação, como o SHAP, podem ser integradas a modelos *Transformer* (Tabassoum et al., 2024), ainda não existe um corpo consolidado de investigação que avalie sistematicamente sua aplicação no domínio noticioso, especialmente no contexto da língua portuguesa.

A multimodalidade representa igualmente um domínio em expansão, mas ainda pouco explorado em línguas menos representadas. A literatura sugere que modelos que integram texto e imagem apresentam melhorias significativas na deteção de conteúdos noticiosos manipulados ou enganosos (Alonso-Bartolome & Segura-Bedmar, 2021; Yan et. al 2024). No entanto, os *datasets* multimodais disponíveis permanecem concentrados na língua inglesa e em outros idiomas dominantes, o que limita a investigação sobre a adaptação destes modelos a conteúdos noticiosos em português e constitui uma barreira relevante ao avanço do estado da arte neste contexto.

Por fim, verifica-se a ausência de análises experimentais abrangentes que comparem, de forma sistemática, os métodos clássicos, as redes neuronais profundas e os modelos *Transformer* aplicados à classificação de notícias em português. A maioria dos estudos existentes restringe-se a comparações parciais ou a um único tipo de abordagem (Yang et al., 2019; Mohanta et al., 2020), descurando fatores como o tempo de inferência, o custo computacional e a robustez dos modelos. Esta lacuna metodológica dificulta a seleção informada de soluções adequadas para aplicações reais e contribui para o distanciamento entre a investigação académica e as necessidades concretas do ecossistema mediático lusófono.

Neste enquadramento, a presente dissertação propõe-se a contribuir para mitigar as lacunas identificadas, com foco no contexto do português. O estudo visa articular a análise e utilização de recursos noticiosos adequados com uma avaliação comparativa de diferentes abordagens de classificação, considerando não apenas o desempenho, mas também a eficiência computacional e a interpretabilidade dos modelos. Ao integrar estas dimensões, a investigação pretende fornecer contributos tanto de natureza científica quanto aplicada, oferecendo diretrizes que apoiem a adoção informada de soluções de PLN por profissionais e organizações jornalísticas lusófonas. O foco no equilíbrio entre desempenho e viabilidade operacional pretende promover a aplicação mais eficaz e sustentável das tecnologias estudadas em contextos reais.

3. FERRAMENTAS, BIBLIOTECAS E *FRAMEWORKS* UTILIZADAS

O presente capítulo apresenta um levantamento detalhado das abordagens e tecnologias que sustentam a categorização automática de texto, com foco na classificação de notícias. São analisados criticamente os métodos que compõem o estado da arte, destacando os princípios subjacentes, as limitações das gerações anteriores e as soluções inovadoras que possibilitaram avanços significativos no desempenho. Ao abordar a evolução das representações estáticas para os modelos contextuais de última geração, este capítulo estabelece a base teórica necessária para fundamentar as escolhas metodológicas da dissertação. A revisão realizada consiste na seleção de arquiteturas específicas para o desenvolvimento de modelos resilientes, capazes de garantir a classificação automática de artigos noticiosos com rigor e eficácia elevados.

3.1. Ferramentas, Bibliotecas e *Frameworks* no Processamento de linguagem Natural

O desenvolvimento acelerado e contínuo do PLN nas últimas décadas resulta não apenas dos avanços teóricos em algoritmos de Aprendizagem Automática, mas também da consolidação de um ecossistema robusto de ferramentas, bibliotecas e *frameworks*. Este conjunto de recursos tecnológicos tornou as técnicas de DL mais acessíveis, confiáveis e, sobretudo, escaláveis, beneficiando, desta forma, a comunidade global de investigação e desenvolvimento (Galassi et al., 2020). Na prática, estas ferramentas desempenham um papel essencial na transição entre a investigação teórica e a implementação de modelos capazes de analisar e classificar automaticamente grandes volumes de texto (Dai et al., 2021).

No domínio da classificação de texto, particularmente quando aplicada à especificidade da língua portuguesa, a seleção da infraestrutura tecnológica é um fator crítico que condiciona a precisão dos resultados e a escalabilidade da solução. Conforme corroborado pela literatura (Wang et al., 2020; Zeng et al., 2022), a robustez das bibliotecas de PLN é fundamental para suportar o estado da arte, uma vez que viabiliza a execução de protocolos de teste rigorosos e a comparação sistemática entre arquiteturas rivais. Esta abordagem permite consolidar *pipelines* de processamento que garantem a integridade dos dados desde a extração até à classificação final.

Esta secção apresenta um levantamento organizado das principais ferramentas atualmente utilizadas em projetos de PLN, estruturando-as de acordo com o seu papel no ciclo de desenvolvimento de modelos de classificação textual. Serão analisadas bibliotecas orientadas a métodos clássicos de

Aprendizagem Automática, como o *Scikit-learn*, bem como *frameworks* de Aprendizagem Profunda, como sendo o *PyTorch* e o *TensorFlow*. Estas últimas são essenciais para construir e treinar arquiteturas complexas, como o LSTM e o *Transformer*. Incluem-se também ferramentas de pré-processamento linguístico, como o NLTK e o *spaCy*, fundamentais para tarefas como a tokenização e a lematização, bem como as bibliotecas especializadas em modelos baseados em *Transformer*s, como a *Hugging Face Transformers*, que facilitam a utilização e o *fine-tuning* de modelos pré-treinados.

Adicionalmente, serão abordadas soluções para a interpretabilidade de modelos, que permitem justificar as decisões dos classificadores, tais como o LIME e o SHAP, bem como tecnologias destinadas à operacionalização em ambientes de produção (*MLOps*), garantindo escalabilidade e monitorização contínua dos modelos.

3.2. Bibliotecas para o *Machine Learning* Clássico

As bibliotecas baseadas em métodos clássicos de Aprendizagem Automática desempenham um papel central na construção de modelos de classificação de texto, especialmente em contextos que exigem simplicidade, interpretabilidade e eficiência computacional. Estas ferramentas são frequentemente utilizadas na literatura académica e na prática industrial, sendo adotadas como referência para comparação com modelos mais complexos (Sarker, 2021; Wang et al., 2020).

A biblioteca *Scikit-learn* é uma das ferramentas mais consolidadas neste domínio (Zeng et al., 2022), oferecendo um ecossistema abrangente de implementações estáveis de algoritmos de classificação, como o *Naive Bayes*, o SVM, a Regressão Logística e o *Random Forest*. Além destes algoritmos, a plataforma integra funcionalidades essenciais para validação cruzada, estruturação de fluxos de trabalho e ajuste de hiperparâmetros, tornando-a especialmente adequada para a validação empírica rigorosa e para a obtenção de resultados cientificamente consistentes (Sarker, 2021).

As bibliotecas *XGBoost* e *LightGBM* destacam-se por seu alto desempenho e escalabilidade. O *XGBoost* (*eXtreme Gradient Boosting*) introduz otimizações específicas ao algoritmo de *Gradient Boosting*, permitindo processar grandes volumes de dados de forma eficiente e reduzindo o risco de sobreajuste (Galassi et al., 2020). De modo semelhante, o *LightGBM* apresenta eficiência no uso de memória e na velocidade de treino, sendo especialmente adequado para cenários de alta dimensionalidade, comuns após a vetorização de texto (Sarker, 2021). Embora estas bibliotecas sejam tradicionalmente aplicadas a dados estruturados, a literatura documenta a sua utilização bem-sucedida

em tarefas de classificação de texto quando combinadas com representações vetoriais baseadas em contagem, como o TF-IDF (Wang et al., 2020).

A biblioteca *FastText*, desenvolvida pelo *Facebook AI Research*, ocupa uma posição intermediária entre modelos lineares clássicos e abordagens baseadas em representações distribuídas. Ao incorporar informações ao nível das subpalavras, o *FastText* lida de forma eficaz com palavras raras, variações morfológicas e ruído ortográfico, aspetos relevantes na língua portuguesa. Além de gerar *embeddings* eficientes, esta biblioteca oferece funcionalidades otimizadas para classificação supervisionada em larga escala, com tempos de treino e de inferência reduzidos (Kumari et al., 2023).

Em conjunto, estas bibliotecas constituem a base tecnológica para a construção de classificadores leves, compreensíveis e eficientes. Em comparação com abordagens de DL, estas oferecem vantagens notáveis em termos de exigência computacional reduzida, maior transparência na interpretação dos resultados e maior facilidade na configuração e no ajuste dos modelos. São adequadas para o desenvolvimento de modelos de referência iniciais, o que possibilita avaliações rápidas. Desta forma, permitem estabelecer referências robustas antes da adoção de modelos de DL mais complexos.

Os detalhes comparativos destas ferramentas e dos respetivos algoritmos de Aprendizagem Automática aplicados ao PLN estão sintetizados na Tabela 5.

Tabela 5 - Tabela Comparativa de Bibliotecas e Algoritmos de Machine Learning para PLN

Biblioteca	Algoritmos Típicos	Foco Principal	Vantagens Chave	Aplicação em PLN (Geral)
Scikit-learn	*Naive Bayes*, SVM, Regressão Logística, *Random Forests*	ML Clássico Geral	Amplo conjunto de implementações estáveis, validação cruzada, *pipelines* integrados	Classificação, quando combinado com representações vetoriais (ex., TF-IDF)
XGBoost	*Gradient Boosting*	Métodos *Ensemble* (*Boosting*)	Elevada performance, escalabilidade, lida eficientemente com grandes volumes de dados	Classificação de texto de alto desempenho
LightGBM	*Gradient Boosting*	Métodos *Ensemble* (*Boosting*)	Eficiência em memória e velocidade de treino (particularmente em alta dimensionalidade)	Cenários de dados com muitas dimensões, oferecendo velocidade superior ao *XGBoost*
FastText	Modelo linear intermédio (baseado em subpalavras)	Representações Distribuídas (*Embeddings* e Classificação)	Lida eficazmente com palavras raras e variações morfológicas (Português), classificação rápida	Geração de *embeddings* e classificação supervisionada em larga escala

3.3. Frameworks de Aprendizagem Profunda para Processamento de Linguagem Natural

O crescimento exponencial do uso de modelos de DL no âmbito do PLN está intrinsecamente associado à evolução de *frameworks* que simplificam a definição, o treino e a implementação de arquiteturas neuronais complexas. A análise comparativa destas *frameworks* é fundamental para compreender de que maneira influenciam o progresso e a aplicação dos modelos avançados de PLN. Estas ferramentas são essenciais, pois abstraem detalhes de baixo nível, como a gestão de memória, o paralelismo e a aceleração por meio de *hardware* especializado, como GPU e TPU (Galassi et al., 2020; Al-Smadi et al., 2018).

O *TensorFlow* é uma plataforma de código aberto desenvolvida pela *Google* que se destaca como um elemento central no desenvolvimento de modelos de DL em escala industrial, dada a sua integração nativa via API, que proporciona um elevado nível de abstração, facilitando a conceção de redes neuronais complexas sem comprometer a flexibilidade técnica. O ecossistema distingue-se por uma arquitetura modular que permite treino distribuído e aceleração computacional por meio de unidades de GPU e TPU. De acordo com Zeng et al. (2022) e Devlin et al. (2019), estas características garantem uma transição eficiente entre a investigação laboratorial e a implementação em ambientes de produção de alta disponibilidade.

Em contraste, o *PyTorch*, desenvolvido pelo *Facebook AI Research*, rapidamente se tornou uma das *frameworks* predominantes em contextos académicos e de investigação (Wang et al., 2020). Este destaque resulta de sua execução dinâmica e da facilidade de depuração, que permitem a construção flexível de grafos computacionais. O *PyTorch* favorece o uso de arquiteturas personalizadas, o que é relevante para o desenvolvimento de modelos avançados de PLN, como as variantes dos *Transformers*. Esta adoção pela comunidade científica acelerou a disponibilização de implementações de referência de modelos de última geração, consolidando esta *framework* como padrão importante na investigação (Al-Smadi et al., 2018).

Recentemente, *frameworks* como o *Just After eXecution* (JAX) e a biblioteca *Flax* têm ganhado destaque em contextos de investigação avançada e de treino de modelos em larga escala. Ambas desenvolvidas pela *Google*, estas ferramentas foram projetadas para otimizar a eficiência computacional, a diferenciação automática e a paralelização em *hardware* especializado, demonstrando elevado potencial para o treino de modelos de linguagem de grandes dimensões (Minaee et al., 2021; Zeng et al., 2022). Contudo, a adoção destas tecnologias também apresenta vários desafios,

entre os quais a necessidade de conhecimento aprofundado em programação funcional e o suporte ainda limitado das bibliotecas, quando comparado ao de *frameworks* mais consolidados, o que pode dificultar sua utilização por investigadores iniciantes ou em ambientes de produção com requisitos variados.

Em síntese, embora estes *frameworks* constituam a infraestrutura tecnológica fundamental para a implementação prática de modelos de PLN baseados em redes neuronais profundas, é importante reconhecer que cada um apresenta vantagens e limitações distintas. Enquanto os *frameworks* consolidados, como o *TensorFlow* e o *PyTorch*, oferecem maturidade, suporte comunitário e vasta documentação, as novas abordagens, como o JAX e o *Flax*, destacam-se pela eficiência computacional e flexibilidade, embora ainda apresentem barreiras de entrada mais elevadas. Por exemplo, grandes empresas de tecnologia têm utilizado o *TensorFlow* no desenvolvimento de sistemas de tradução automática em tempo real, beneficiando-se de sua estabilidade e de seu suporte extenso, enquanto investigadores em laboratórios acadêmicos exploram o JAX para o treino eficiente de grandes modelos linguísticos, destacando a importância de uma escolha alinhada a cada caso de uso. Desta forma, a escolha da *framework* mais apropriada exige uma avaliação crítica das exigências do desempenho, implicando que o avanço do estado da arte depende não apenas da tecnologia disponível, mas também da sua integração eficaz com o contexto de aplicação.

3.4. Ferramentas, Bibliotecas e *Toolkits* de Pré-Processamento Linguístico

O pré-processamento linguístico constitui uma etapa fundamental no desenvolvimento de soluções de PLN, tendo como principal objetivo converter o texto bruto, frequentemente caracterizado por ruído, ambiguidade e alta variabilidade, num formato limpo, normalizado e estruturado, adequado ao processamento por modelos estatísticos e neuronais, o que potencializa o desempenho preditivo (Dai et al., 2021). Esta etapa assume especial relevância no contexto da língua portuguesa, cuja complexidade morfológica exige *toolkits* de pré-processamento robustos (Kumari et al., 2023).

Entre as bibliotecas mais utilizadas destacam-se o NLTK, o *spaCy*, a *Stanza* e o *Gensim*. O *Natural Language Toolkit* (NLTK) é uma biblioteca pioneira em *Python*, amplamente reconhecida por sua relevância no ensino e na pesquisa académica (Pang et al., 2002). Apresenta uma arquitetura modular e flexível, que implementa uma ampla gama de algoritmos e disponibiliza recursos lexicais e conjunto de dados. A biblioteca oferece suporte a tarefas essenciais, como a tokenização de palavras e frases, o *stemming* e a lematização baseada em dicionários. No entanto, apesar de sua abrangência, o

NLTK apresenta desempenho inferior em comparação com bibliotecas mais recentes, o que limita sua aplicação em grandes volumes de dados (Zeng et al., 2022).

O *spaCy* é projetado para alta velocidade e para uso em ambientes de produção. Desenvolvido pela *Explosion* AI em *Cython* (extensão do *Python*), processa texto de forma eficiente, convertendo-o em um objeto que reúne todas as anotações linguísticas relevantes. A biblioteca executa rapidamente tarefas complexas, como o reconhecimento de entidades nomeadas e a análise de dependências sintáticas. Para além disso, disponibiliza modelos pré-treinados de alta precisão em português, o que a torna especialmente adequada para o processamento de grandes volumes de notícias (Kumari et al., 2023).

O *Stanza* representa uma solução integrada de alta precisão para o processamento de múltiplos idiomas (Qi et al., 2020). Com base em arquiteturas de DL, o sistema emprega redes neuronais recorrentes e aprendizagem multitarefa para otimizar os resultados. Uma distinção importante desta biblioteca em relação às demais é que todos os seus módulos, desde a tokenização e a lematização até a análise de dependências, são treinados conforme um padrão de anotação gramatical, comum a mais de 66 idiomas, incluindo o português, o que assegura a robustez e a consistência translinguísticas. Apesar de exigir maior capacidade computacional, o *Stanza* estabelece o estado da arte em tarefas que exigem rigor analítico máximo.

O *Gensim* é especializado em modelagem semântica e na geração de representações vetoriais densas (*embeddings*), fundamentais para a engenharia de características. A biblioteca permite o processamento de grandes volumes de texto e implementa modelos como o *Word2Vec*, o *Doc2Vec* e o *FastText*, que consideram n-gramas de caracteres. Esta abordagem é particularmente eficaz para lidar com a morfologia do português e com palavras fora do vocabulário. O *Gensim* converte o texto pré-processado em vetores numéricos de alta dimensão, que servem de entrada para modelos de ML e DL (Sarker, 2021).

A Tabela 6 apresenta uma síntese sistemática e comparativa das principais características das ferramentas discutidas, incluindo a linguagem de programação base, as tarefas de pré-processamento suportadas, os principais pontos fortes e as limitações típicas de cada uma.

Tabela 6 - Comparação das Principais Toolkits do PLN

Toolkit	**Foco Principal**	**Otimização / Arquitetura**	**Vantagens Chave**	***Trade-off* (Performance vs. Precisão)**
NLTK	Ensino, Investigação Académica, Prototipagem	*Python*, Modular	Vasta gama de algoritmos básicos, extensa coleção de conjuntos e recursos lexicais.	Mais lento (não otimizado para produção), excelente para elaboração de experiências.
spaCy	Uso Industrial, Produção em Larga Escala	*Cython* (velocidade), *Pipelines* estatísticos pré-treinados	Alta velocidade, eficiência computacional, modelos otimizados para Português (e.g., *pt_core_news_*) e Análise de Dependências.	Preferencial para processamento eficiente de grandes conjuntos de dados em produção.
Stanza	Alta Precisão, Solução Unificada Multilíngua	DL (Redes Neuronais), baseado em Dependências Universais	Precisão a nível de Estado de Arte em tarefas linguísticas devido à complexidade dos modelos neuronais.	Maior custo computacional, mais lento e intensivo em memória do que o *spaCy*.
Gensim	Modelação Semântica e Representações Vetoriais (*Embeddings*)	Eficiência para grandes conjuntos de dados	Implementação otimizada de *Word2Vec*, *Doc2Vec* e *FastText*; lida com dados que excedem a memória RAM.	Foco especializado, uma vez que não é um toolkit de pré-processamento linguístico completo.

3.5. Ecossistema para o Modelo *Transformer*

A implementação prática de modelos de linguagem baseados na arquitetura *Transformer*, especialmente de modelos pré-treinados de grande escala como o BERT e suas variantes, em contextos reais e de larga escala, depende de um ecossistema de *software* robusto, maduro e padronizado. Neste contexto, a biblioteca *Hugging Face Transformers* destaca-se como a principal infraestrutura técnica para o desenvolvimento e a implementação destes modelos (Zeng et al., 2022).

A biblioteca *Hugging Face Transformers* consolidou-se como a principal solução para o desenvolvimento de modelos baseados em *Transformers* no PLN (Wolf et al., 2020). A sua relevância resulta da padronização das interfaces de programação, da facilidade de uso e da disponibilização de modelos de última geração. O conjunto de ferramentas oferece uma interface unificada que permite o carregamento, uso e ajuste eficiente de modelos pré-treinados, de forma a reduzir a complexidade técnica. Esta abstração facilita a adoção do paradigma da aprendizagem por transferência, permitindo a adaptação de modelos treinados em grandes volumes de texto genérico para tarefas específicas, como a classificação automática de notícias, mesmo com um número limitado de dados anotados (Galassi et al., 2020).

A biblioteca *Hugging Face Transformers* disponibiliza implementações de modelos baseados na arquitetura *Transformer* que diferem em arquitetura, estratégias de pré-treino e métodos de otimização, possibilitando a sua aplicação em diversos contextos experimentais e aplicados. Por exemplo, na tarefa de classificação automática de notícias em tempo real, um modelo de grande porte, como o BERT *Large*, pode oferecer maior precisão. Contudo, a sua implementação exige capacidade computacional significativa, resultando em maior tempo de resposta e maior consumo de memória. Em contrapartida, os modelos mais compactos, como o *DistilBERT*, apresentam um desempenho levemente inferior em termos de precisão, mas são mais adequados para aplicações que exigem respostas rápidas e operam sob restrições de recursos computacionais, como é o caso das plataformas de notícias online. Desta forma, a escolha do modelo específico exige uma análise criteriosa do custo-benefício entre o desempenho preditivo e a eficiência computacional, devendo considerar o impacto destas limitações em relação às necessidades da tarefa e aos recursos disponíveis.

Neste contexto, a Tabela 7 apresenta uma síntese comparativa dos principais modelos baseados na arquitetura *Transformer* disponíveis no ecossistema *Hugging Face*, destacando as suas características essenciais e os respetivos compromissos entre o desempenho e o custo computacional.

Tabela 7 - Tabela Comparativa de Modelos de DL em PLN através do ecossistema Hugging Face

Modelo	**Arquitetura Principal**	**Estratégia de Pré-treino Chave**	**Foco de Otimização**	***Trade-off* (Precisão vs. Eficiência)**
BERT	*Encoder*, *Transformer* Bidirecional	MLM e NSP	Ponto de partida padrão para Aprendizagem por Transferência em PLN.	Alto poder preditivo, mas computacionalmente exigente para inferência.
RoBERTa	*Encoder*, *Transformer* Bidirecional (similar ao BERT)	MLM (com conjunto de dados maior, mais treino, NSP removido)	Máxima Precisão (performance superior ao BERT em *benchmarks*).	Requer mais recursos de treino e maior exigência na inferência que o BERT.
DistilBERT	Versão menor e mais compacta do BERT	Destilação de Conhecimento (*Knowledge Distillation*) do modelo BERT	Eficiência e Baixa Latência (para produção ou ambientes de recursos limitados).	Significativamente mais rápido e leve (40% menos parâmetros) mantendo alta precisão (~97% do BERT).

Entre os modelos disponibilizados, o BERT constitui o modelo fundador da era *Transformer* e é amplamente utilizado como ponto de partida para um vasto conjunto de tarefas de PLN (Devlin et al., 2019). A sua principal inovação reside na aprendizagem de representações contextuais profundas e bidirecionais, obtidas através das tarefas de pré-treino MLM e NSP. A biblioteca *Hugging Face*

disponibiliza múltiplas variantes do BERT, abrangendo diferentes dimensões e idiomas, incluindo versões especializadas como o *BERTimbau*, desenvolvido especificamente para a língua portuguesa (Souza et al., 2020).

O *RoBERTa* representa uma evolução do BERT, preservando a arquitetura base do codificador *Transformer*, mas adotando uma estratégia de pré-treino mais eficiente, assente em conjuntos de dados significativamente maiores e em períodos de treino prolongados (Liu et al., 2019). Esta abordagem resulta num desempenho consistentemente superior em diversos benchmarks de PLN, embora implique um aumento dos requisitos computacionais, sendo frequentemente adotada em cenários onde a maximização da precisão é prioritária.

Por sua vez, o *DistilBERT* surge como uma alternativa otimizada, concebida para mitigar o elevado custo computacional associado aos modelos *Transformer* de grande dimensão (Sanh et al., 2019). Através da técnica de destilação de conhecimento, este modelo compacto preserva uma parte substancial do desempenho do BERT original, ao mesmo tempo que reduz significativamente o número de parâmetros e o tempo de inferência. Estas caraterísticas tornam-no particularmente adequado para aplicações em ambientes com restrições de recursos ou com requisitos de baixa latência.

Em síntese, a distinção entre a biblioteca *Hugging Face Transformers*, enquanto infraestrutura de software, e o ecossistema *Hugging Face*, enquanto conjunto integrado de modelos, recursos e ferramentas, é fundamental para compreender a viabilização prática dos modelos *Transformer*. Esta articulação entre a biblioteca e o ecossistema sustenta o desenvolvimento, a comparação e a implementação dos modelos analisados na presente dissertação, constituindo a base tecnológica das experiências descritas nos capítulos seguintes.

3.6. Ferramentas de Interpretabilidade

A adoção de modelos de DL e de arquiteturas *Transformer* em domínios críticos, como a classificação automatizada de notícias, evidenciou limitações associadas à opacidade dos mecanismos de decisão algorítmica (Guidotti et al., 2018). Estas estruturas apresentam obstáculos significativos à auditoria e à compreensão de seu funcionamento interno. Como resultado, surgem desafios éticos e regulatórios relacionados à atribuição de responsabilidade e à mitigação de possíveis anomalias (Dai et al., 2021). Para este cenário, a Inteligência Artificial Explicável (XAI) desempenha um papel importante ao propor técnicas que justificam as decisões dos modelos, tornando os resultados destes

modelos acessíveis e compreensíveis para os utilizadores (Sarker, 2021).

No espectro das metodologias mais influentes, encontram-se as soluções independentes da estrutura do modelo, que oferecem a vantagem de poderem ser integradas em modelos variados. É neste contexto que o LIME e o SHAP se afirmam como referências no domínio do PLN, dada a sua popularidade, decorrente do rigor teórico que sustenta as suas explicações (Zeng et al., 2022).

O *Local Interpretable Model Agnostic Explanations* (LIME) representa uma das primeiras técnicas desenvolvidas no contexto da Inteligência Artificial Explicável, com o propósito de fornecer explicações locais e intuitivas para previsões individuais de modelos complexos, independentemente de sua estrutura interna (Ribeiro et al., 2016). O princípio fundamental do LIME baseia-se na premissa de que, embora o comportamento global de um modelo seja difícil de interpretar, é possível aproximar seu funcionamento localmente por meio de modelos simples e compreensíveis, como a Regressão Linear ou a Árvore de Decisões. No contexto do PLN, o LIME opera criando pequenas variações no texto original, removendo ou ocultando palavras aleatoriamente, gerando novas amostras artificiais. O modelo original classifica estas instâncias, atribuindo a cada uma um peso proporcional à sua semelhança com o texto inicial. Este procedimento permite identificar quais termos exercem maior influência sobre a decisão do modelo, facilitando a interpretação de arquiteturas complexas.

O *SHapley Additive exPlanations* (SHAP) constitui uma abordagem unificada para a interpretação de previsões. Nesta abordagem, cada caraterística do modelo é considerada um participante que contribui para o resultado final da previsão. O método atribui a cada característica um valor que representa a sua contribuição marginal para a diferença entre a previsão de uma instância específica e a de uma previsão de referência. Uma das principais vantagens do SHAP reside no seu carácter aditivo, pois garante que a soma das contribuições individuais corresponde exatamente à variação observada na previsão, assegurando consistência e rigor teórico. Para além de possibilitar a análise local de previsões individuais, a agregação dos valores SHAP ao longo de múltiplas instâncias permite uma interpretação global do modelo, identificando os fatores linguísticos mais relevantes para a tomada de decisão em todo o conjunto de dados (Dai et al., 2021).

A integração destes métodos de interpretabilidade é especialmente relevante em contextos jornalísticos, nos quais a transparência e a justificativa de decisões automáticas constituem requisitos essenciais para a adoção responsável de modelos baseados em PLN.

3.7. Ferramentas de *Deploy* e Produção

A fase de *deploy* e produção representa uma componente central da Engenharia de ML (MLOps), pois marca a transição de um modelo experimental para um serviço operacional estável, pronto para responder a solicitações em tempo real ou em processamento em lote. No contexto do PLN, em que modelos baseados em arquiteturas *Transformer* apresentam alta complexidade e elevados custos computacionais, esta etapa adquire especial importância, especialmente quanto à latência de inferência e à eficiência do serviço oferecido (Galassi et al., 2020).

Um dos principais desafios nesta etapa é a ausência de comunicação direta entre diferentes ambientes de treino, especialmente entre o *PyTorch* e o *TensorFlow*. Para superar esta limitação, utiliza-se o *Open Neural Network Exchange* (ONNX), um formato padronizado que converte modelos para uma linguagem comum, independentemente da ferramenta de origem (Zeng et al., 2022). Após a conversão, o modelo pode ser executado pelo ONNX *Runtime*. Este mecanismo de inferência separa o treino da produção, otimizando o uso de memória e a velocidade de resposta, aspetos essenciais para a disponibilização de modelos robustos em ambientes produtivos.

Para além disto, o *Docker* desempenha um papel fundamental na operacionalização de modelos de PLN. O *Docker* permite empacotar aplicações e as suas dependências em contentores isolados (Wang et al., 2020). Por exemplo, ao realizar o *deploy* de um modelo *Transformer* para a análise de sentimentos em texto, é possível criar uma imagem *Docker* contendo o código do modelo, as bibliotecas, como o *Hugging Face Transformers*, e as versões exatas de *Python* e as suas dependências relacionadas. Em seguida, o container pode ser executado tanto no ambiente de desenvolvimento quanto em servidores de produção, assegurando a consistência operacional, minimizando os conflitos de dependência. Para os modelos *Transformer*, a contentorização isola ambientes com bibliotecas e versões específicas, promovendo a criação de serviços robustos e facilmente replicáveis.

As ferramentas especializadas de serviço de modelos, como o *TensorFlow Serving* e o *TorchServe*, foram desenvolvidas para disponibilizar modelos de ML em ambientes de produção de forma eficiente e escalável (Dai et al., 2021). O *TensorFlow Serving* é otimizado para modelos criados em *TensorFlow*, enquanto o *TorchServe* integra-se ao ecossistema do *PyTorch*. Embora as plataformas atuais facilitem a gestão do ciclo de vida dos modelos por meio de interfaces de inferência prontas para uso, a sua implementação prática ainda enfrenta obstáculos significativos, entre os quais destacam-se a complexidade na configuração de bibliotecas, a falta de compatibilidade entre diferentes formatos de modelos e a dificuldade em personalizar fluxos de trabalho específicos. Contudo, estas ferramentas

oferecem funcionalidades essenciais, como a gestão de versões, que permitem atualizações sem interrupção do serviço e a otimização da inferência por processamento em lote, maximizando a eficiência do *hardware* e o rendimento computacional (Devlin et al., 2019). Em última análise, a adoção destas soluções impulsiona a operacionalização de modelos de PLN, resultando em modelos mais robustos, escaláveis e adequados às exigências dos ambientes de produção.

4. METODOLOGIA

O presente capítulo apresenta a metodologia de investigação e desenvolvimento adotada para alcançar os objetivos desta dissertação. A abordagem metodológica adotada possui um caráter híbrido e estrutural, de forma a garantir a validade científica das hipóteses formuladas e a reprodução dos resultados.

4.1. Estrutura de Investigação e Desenvolvimento

Esta secção apresenta o enquadramento metodológico central utilizado ao longo de toda a presente dissertação. Para garantir que o desenvolvimento do modelo de classificação automática de notícias atendesse a critérios de rigor científico e de relevância prática, foi adotada uma estratégia metodológica híbrida. Esta abordagem articula a visão estratégica de um ciclo de investigação com a sistematicidade de um processo de desenvolvimento de modelos preditivos, de forma a assegurar que todas as etapas do projeto, desde a formulação das hipóteses até à validação empírica dos resultados, sejam conduzidas de forma lógica, transparente e replicável.

O projeto foi concebido integrando duas abordagens metodológicas complementares, operando em diferentes níveis de abstração:

1. Ciclo de Investigação e Desenvolvimento (I&D): Este ciclo definiu a macroestrutura da dissertação, dividindo o trabalho em quatro fases essenciais: o Planeamento, focado na pesquisa bibliográfica e na definição de requisitos; a Execução, que compreende a preparação e modelação de dados; a Avaliação, dedicada à validação empírica; e a Iteração, destinada aos trabalhos futuros

2. *Framework Cross-Industry Standard Process for Data Mining* (CRISP-DM): Esta *framework* serviu como guia detalhado para a fase de Execução, permitindo estruturar, de forma sistemática, as tarefas relacionadas à compreensão do negócio, à preparação dos dados, à modelação e à avaliação dos diversos modelos.

A integração destas metodologias permitiu uma transição fluida entre os objetivos científicos e a implementação técnica, bem como potencializou a abordagem do projeto ao combinar uma visão macroestrutural com um guia operacional detalhado. Este alinhamento metodológico reforçou a robustez da engenharia de dados ao integrar a orientação estratégica do ciclo I&D com as etapas

práticas e sistemáticas do CRISP-DM, promovendo a eficiência, o controlo e a consistência, facilitando a adaptação e iteração contínuas com base nos resultados e desafios surgidos ao longo do processo.

4.2. Requisitos e Critérios de Sucesso do Trabalho Experimental

A formalização dos requisitos constitui uma etapa fundamental para converter os objetivos gerais da presente dissertação em indicadores quantificáveis e aferíveis, garantindo a coerência entre a metodologia adotada e os resultados experimentais. Estes critérios são indispensáveis para a fase de avaliação do processo CRISP-DM, pois permitem avaliar tanto o funcionamento das abordagens implementadas quanto a qualidade do desempenho preditivo, a eficiência computacional e a capacidade de generalização.

4.2.1. Requisitos Funcionais

Os requisitos descritos de seguida definem as funções que o modelo de classificação deve executar para atingir o seu objetivo principal:

- Classificação Automática: O modelo deve atribuir automaticamente a categoria mais provável às notícias em português, considerando um conjunto pré-definido de classes como por exemplo, as categorias de Política, Economia, Desporto, entre outras.

- Geração de Confiança: O modelo deve fornecer um valor de probabilidade ou de confiança para cada previsão, de forma a permitir a avaliação dos resultados obtidos.

- Robustez Linguística: O modelo deve processar texto bruto, reconhecendo a diversidade lexical e a variação sintática presentes na língua portuguesa.

4.2.2. Requisitos Não Funcionais

Os requisitos não funcionais definem os critérios de qualidade para avaliar o desempenho das abordagens experimentais. Estes critérios permitem a análise da precisão, da eficiência computacional, da interpretabilidade e da robustez dos modelos no contexto do estudo:

- Desempenho Preditivo (*F1-Score*): O sucesso do modelo é avaliado pelo *F1-Score* ponderado, uma métrica adequada para cenários com classes desequilibradas. O modelo de última geração, o *BERTimbau*, deve apresentar desempenho percentualmente superior ao do melhor classificador de linha de base, o SVM. Este requisito pretende validar empiricamente a adoção da arquitetura *Transformer*, justificando a sua complexidade em termos de ganhos de desempenho.

- Eficiência Temporal (Latência): O tempo de inferência para classificar uma notícia deve ser reduzido. Este requisito garante a viabilidade técnica do modelo para integrações futuras em ambientes que exigem respostas rápidas, aproximando-se do funcionamento em tempo real.

- Transparência e Interpretabilidade: O modelo deve assegurar interpretabilidade adequada, utilizando métodos de Inteligência Artificial Explicável que permitam a compreensão e a justificativa das decisões do modelo. Este aspeto é essencial para fortalecer a confiança no modelo e viabilizar a sua adoção em contextos reais.

- Generalização: O desempenho do modelo não deve depender das classes maioritárias do conjunto de dados, ou seja, o modelo deve demonstrar robustez e a capacidade de generalização, apresentando um desempenho consistente nas classes minoritárias, de modo a mitigar os efeitos do desequilíbrio inerente ao *dataset* utilizado.

5. TRABALHO EXPERIMENTAL

Este capítulo apresenta o trabalho experimental realizado durante a presente dissertação, detalhando a implementação e avaliação de diferentes abordagens para a categorização automática de notícias em língua portuguesa. Durante este capítulo, são descritos o conjunto de dados utilizado, o processo de preparação e as estratégias de modelação adotadas, incluindo os modelos clássicos de ML, de DL e baseados em arquiteturas *Transformer*. A análise foca tanto no desempenho obtido por meio de métricas quantitativas quanto nos requisitos computacionais de cada abordagem, permitindo uma comparação sistemática entre as soluções estudadas e estabelecendo a base para a análise e discussão dos resultados que serão efetuados no capítulo seguinte.

5.1. Conjunto de Dados Utilizado e *Pipeline* de Preparação

O presente estudo recorre ao conjunto de dados *CC News PT v2*, disponibilizado publicamente por Garcia (Garcia, 2021), o qual constitui a base empírica de toda a componente experimental da presente dissertação. Este *dataset* é utilizado de forma transversal no treino, validação e avaliação dos diferentes modelos de classificação analisados, incluindo as abordagens clássicas de ML, as arquiteturas de DL e modelos baseados em *Transformers*. A utilização de um único conjunto de dados comum a todas as experiências garante a consistência metodológica e a comparação direta dos resultados obtidos.

O *CC News PT v2* consiste num conjunto de notícias em língua portuguesa, recolhidas a partir de fontes jornalísticas digitais, abrangendo um conjunto diversificado de domínios temáticos. Cada instância do *dataset* corresponde a um artigo noticioso completo, integrando o título e o corpo do texto, e encontra-se associada a uma categoria temática previamente definida. O problema em estudo enquadra-se, desta forma, num cenário de classificação multiclasse, em que cada notícia é atribuída a uma única classe.

As categorias presentes no conjunto de dados refletem áreas temáticas típicas do contexto noticioso, tais como política, economia, desporto, sociedade, cultura, ciência e tecnologia, entre outras. Esta diversidade temática permite avaliar o comportamento dos diferentes modelos em contextos semânticos variados e com diferentes níveis de complexidade linguística, tornando o *dataset* particularmente adequado para estudos comparativos de categorização automática de notícias em língua portuguesa.

Os dados encontram-se organizados num formato estruturado, adequado ao processamento automático, contendo campos textuais e o respetivo rótulo de classe. Para efeitos do presente trabalho, o texto da notícia é considerado ao nível do documento, resultando da concatenação do título com o corpo do artigo, de forma a preservar o máximo de contexto semântico relevante para a tarefa de classificação.

O conjunto de dados utilizado na presente investigação passou por um processo sistemático de preparação de forma a garantir a qualidade, a consistência e a otimização da classificação automática de notícias. Esta etapa é fundamental, pois o desempenho dos modelos de ML e do DL depende diretamente da qualidade e da representatividade dos dados de entrada.

Inicialmente, os dados textuais foram inspecionados e limpos para remover entradas incompletas, duplicadas ou inconsistentes. Para além disso, foram eliminados elementos não informativos, como caracteres especiais redundantes, marcas de formatação e espaços em branco excessivos, que poderiam gerar ruído no processo de aprendizagem. Estas operações uniformizaram o conjunto de dados e garantiram a integridade da amostra utilizada nas experiências.

De seguida, o texto foi normalizado por meio da conversão para minúsculas e da padronização de acentuação, se aplicável. Estas etapas são justificadas pelo facto de que a distinção entre letras maiúsculas e minúsculas e as variações de acentuação geralmente não alteram o significado dos termos, mas podem ser interpretadas pelos algoritmos como elementos distintos, prejudicando a consistência das representações. Ao reduzir essas variações superficiais, evita-se que o modelo direcione atenção para diferenças irrelevantes, concentrando a aprendizagem em padrões semânticos realmente relevantes para a distinção entre categorias, o que potencializa a eficácia dos classificadores.

Após a tarefa de limpeza e normalização, o conjunto de dados foi organizado para permitir uma avaliação rigorosa dos modelos. O conjunto de dados foi dividido em subconjuntos de treino e de teste, em proporções adequadas, de modo a garantir aprendizagem eficaz e validação imparcial. Esta separação assegura que os resultados reflitam a capacidade de generalização dos modelos para dados não observados durante o treino.

Para além disto, foi analisada a distribuição das classes no conjunto de dados, com o objetivo de identificar possíveis desequilíbrios entre as categorias temáticas. Esta análise é relevante para tarefas de classificação de notícias, na medida em que algumas categorias podem estar sub-representadas. No entanto, é importante ressalvar que possíveis desequilíbrios nas classes constituem uma limitação deste estudo, pois podem comprometer a capacidade dos modelos de aprender de forma equitativa sobre

todas as categorias, favorecendo as mais representativas. A caracterização da distribuição de classes permitiu antecipar os impactos no treino, fundamentar decisões metodológicas nas etapas seguintes da investigação e apontar a necessidade de considerar estratégias compensatórias em casos de desequilíbrio, como o uso de técnicas de reamostragem ou o ajuste de métricas utilizadas para as classes minoritárias, que serão explicadas ao longo do processo experimental.

Esta secção tratou apenas da preparação e da caracterização do conjunto de dados, estabelecendo as bases para o trabalho experimental. A descrição detalhada do *pipeline* de processamento, incluindo a representação do texto, a modelação, o treino e a avaliação dos modelos, serão apresentados na próxima secção, dedicada à implementação e à avaliação experimental.

5.2. Processo de Implementação e Avaliação

Esta secção apresenta a execução do trabalho, detalhando o processo seguido ao longo do mesmo, desde o desenvolvimento da *pipeline* prática da Aprendizagem Automática até à estratégia de validação empírica dos resultados. O procedimento descrito segue o ciclo CRISP-DM, garantindo a coerência metodológica do estudo e facilitando a compreensão das etapas realizadas.

5.2.1. Preparação e Representação dos Dados

A qualidade da classificação depende diretamente da preparação do *input*. Para garantir a comparação entre as abordagens clássicas e modernas, a preparação dos dados seguiu estratégias específicas para cada família de modelos, após o pré-processamento básico descrito anteriormente.

A Figura 2 apresenta o *pipeline* unificado, que organiza o fluxo do texto bruto até à classificação final e destaca os diferentes métodos de representação textual adotados em cada paradigma utilizado.

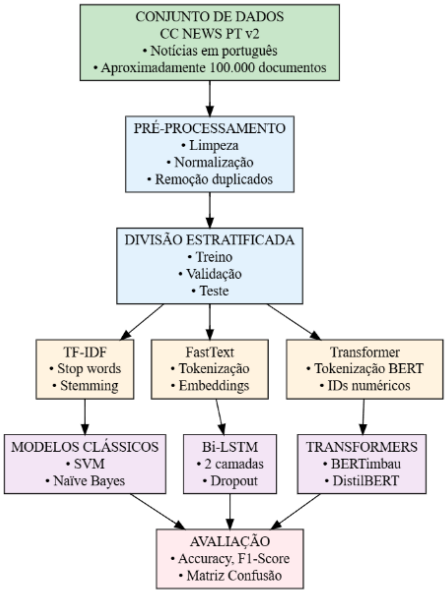

Figura 2 – Pipeline de Preparação e Utilização do Conjunto de Dados

O processo começa com o conjunto de dados *CC News PT v2*, que passa por um pré-processamento básico comum a todas as abordagens. Esta etapa inclui limpeza textual, normalização para minúsculas e remoção de duplicados, assegurando um texto-base limpo e consistente.

Em seguida, realiza-se a divisão estratégica dos dados, particionando o conjunto de textos de forma estratificada em treino, validação e teste. Esta divisão preserva a distribuição original das categorias temáticas, permitindo avaliar a capacidade de generalização dos modelos.

Na etapa de representação do texto, o fluxo experimental divide-se em três abordagens, cada uma adaptada a uma família de modelos:

1. Representação TF-IDF: Utilizada em modelos clássicos de ML, como o SVM e o *Naive Bayes*. Inclui remoção de *stop words*, o *stemming* e a vetorização baseada na frequência de unigramas e bigramas.
2. *Embeddings FastText*: Aplicados à arquitetura Bi-LSTM, convertem os textos em sequências de vetores de palavras pré-treinados em português, capturando informações morfológicas em nível de subpalavras.
3. Tokenização do *Transformer*: Específica para os modelos *BERTimbau* e *DistilBERT*, segmenta o texto em tokens e converte-os em identificadores numéricos e em filtros de atenção, conforme exigido pela arquitetura, preservando o contexto lexical original.

Cada abordagem de representação alimenta o seu respetivo modelo de classificação: os vetores

TF-IDF são processados por algoritmos clássicos de Aprendizagem Supervisionada; as sequências de *embeddings FastText* alimentam a arquitetura Bi-LSTM; e os *tokens* dos *Transformers* são processados por suas respetivas arquiteturas após o *fine-tuning* para a tarefa de classificação de notícias.

Por fim, todos os modelos convergem na fase de avaliação, na qual o desempenho é medido e comparado pelas mesmas métricas padrão: *Accuracy*, Precisão, *Recall* e *F1-Score.* Esta abordagem unificada garante uma comparação justa e objetiva entre os paradigmas, permitindo avaliar tanto a precisão preditiva quanto a eficiência computacional de cada um.

O *pipeline* apresentado anteriormente garante que todas as experiências utilizem a mesma base de dados e sejam avaliadas de forma consistente, permitindo isolar e analisar o impacto das diferentes estratégias de representação e das arquiteturas de classificação no desempenho final da categorização automática de notícias na língua portuguesa.

5.2.2. Modelação e Estratégia de Treino

A fase de modelação e treino dos dados é essencial para o desenvolvimento do modelo de classificação automática, pois transforma dados preparados em modelos capazes de realizar previsões precisas e generalizáveis. Neste trabalho, selecionou-se um conjunto diverso de modelos para avaliar sistematicamente o impacto da representação contextual, da arquitetura e do custo computacional na classificação de notícias. A abordagem combina modelos clássicos de Aprendizagem Automática com Arquiteturas Profundas de última geração, permitindo comparar a eficácia de técnicas lineares tradicionais, redes neuronais avançadas e Arquiteturas *Transformers*.

Para garantir a robustez e evitar o sobreajuste nos modelos de DL, foram aplicadas técnicas de regularização reconhecidas, entre as quais o *Dropout*, que desativa aleatoriamente uma fração das unidades de uma camada em cada iteração de treino, impedindo dependência excessiva de conexões específicas e promovendo representações mais consistentes. Na prática, o *Dropout* funciona como um *ensemble*, melhorando a generalização para dados não vistos. Para além disso, utilizou-se o *Early Stopping*, que monitoriza o desempenho no conjunto de validação e interrompe o treino quando a métrica de desempenho deixa de melhorar, prevenindo o sobreajuste, selecionando a versão do modelo com maior capacidade de generalização e economizando recursos computacionais.

A investigação está estruturada em quatro configurações principais, cujas características técnicas e objetivos experimentais são apresentados de seguida:

- *BERTimbau* (*fine-tuning*): O *BERTimbau* é um modelo *Transformer* baseado na arquitetura BERT, adaptado ao português, que utiliza *embeddings* contextualizados pré-treinados em grandes conjuntos de dados. Para a classificação, realizou-se o *fine-tuning* do modelo, ajustando hiperparâmetros, como o *learning rate* e o *batch size*, e aplicando validação cruzada para garantir a robustez na generalização. Esta abordagem explora representações linguísticas profundas, capturando nuances semânticas do português e melhorando a discriminação entre categorias de notícias.

- *DistilBERT* (*fine-tuning*): O *DistilBERT* é uma versão compacta do BERT, projetada para reduzir a complexidade computacional mantendo a capacidade de contextualização. Utiliza *embeddings* contextualizados e foi submetido a *fine-tuning* na tarefa de classificação, seguindo uma estratégia semelhante à do *BERTimbau*. Este modelo avalia o impacto da redução de parâmetros no desempenho, permitindo analisar a relação entre a eficiência computacional e a qualidade preditiva.

- BERT + SVM: Nesta abordagem, o BERT pré-treinado é utilizado apenas como extrator de características, sem *fine-tuning*, fornecendo *embeddings* contextualizados das camadas finais. Estas representações servem de entrada para um classificador SVM linear, isolando o efeito das representações profundas do BERT, facilitando a análise do impacto da arquitetura do classificador no desempenho final.

- Bi-LSTM + *FastText*: A combinação do Bi-LSTM com *embeddings* pré-treinados em português constitui uma abordagem alternativa de sequenciamento, fornecendo uma representação lexical robusta, enquanto a Bi-LSTM captura as dependências temporais entre palavras. O treino é realizado com técnicas de regularização como *dropout* e *early stopping*. Esta configuração permite comparar o desempenho de uma arquitetura recorrente clássica com modelos baseados em *Transformers*, avaliando a influência da natureza sequencial do texto na classificação.

5.2.3. Avaliação e Validação Empírica

A avaliação e validação empírica dos modelos de classificação desenvolvidos neste estudo tiveram como objetivo analisar, de forma sistemática e comparável, o desempenho preditivo das diferentes abordagens testadas, bem como a sua capacidade de generalização e eficiência computacional. O processo de avaliação adotado está sintetizado nos seguintes pontos:

- Utilização da matriz de confusão como base da avaliação: A avaliação do desempenho dos modelos foi fundamentada na matriz de confusão, apresentada previamente no Capítulo 2. Esta matriz permitiu a correspondência entre as classes reais e as previstas pelos classificadores, identificando os valores de Verdadeiros Positivos, Verdadeiros Negativos, Falsos Positivos e Falsos Negativos. A análise desses elementos permitiu uma compreensão detalhada dos acertos e dos diferentes tipos de erro de cada modelo, servindo de base para o cálculo das métricas de desempenho utilizadas no estudo experimental.

- Cálculo das métricas de desempenho a partir da matriz de confusão: A partir dos valores obtidos na matriz de confusão, calcularam-se as métricas de desempenho empregadas na fase experimental: *Accuracy*, Precisão, *Recall* e *F1-Score*. A métrica *Accuracy* foi utilizada como indicador global da proporção de classificações corretas. A Precisão e o *Recall* permitiram avaliar, respetivamente, a qualidade das previsões positivas e a capacidade do modelo de identificar corretamente as instâncias de cada classe. O *F1-Score* foi adotado como métrica agregadora, possibilitando a análise do equilíbrio entre a Precisão e o *Recall*, especialmente relevante em cenários com distribuição desigual das classes.

- Agregação das métricas em cenários de classificação multiclasse: Dado que o problema analisado envolve um cenário de classificação multiclasse, as métricas da Precisão, do *Recall* e do *F1-Score* foram agregadas por meio das estratégias média micro e média macro. A média micro refletiu o desempenho global do modelo, influenciada pela frequência das classes mais representativas. Em contrapartida, a média macro avaliou o desempenho médio por classe, atribuindo o mesmo peso a todas as categorias, inclusive as minoritárias, o que proporcionou uma análise mais equilibrada e

informativa do comportamento dos classificadores.

- Estratégia de validação e generalização dos modelos: A validação dos modelos foi conduzida com conjuntos de teste independentes, não utilizados na fase de treino. Esta estratégia proporcionou uma estimativa mais confiável da capacidade de generalização dos classificadores, permitindo avaliar o desempenho em dados não vistos e reduzindo o risco de sobreajuste. A separação entre os dados de treinamento, validação e teste foi mantida de forma consistente em todas as experiências realizadas.

- Análise da eficiência computacional: Além do desempenho preditivo, foram considerados indicadores de eficiência computacional, como o tempo de treino e de inferência dos modelos. Esta análise permitiu avaliar o equilíbrio entre a qualidade da classificação e os recursos computacionais exigidos por cada abordagem, sendo especialmente relevante na comparação entre modelos clássicos, de DL e baseados em *Transformers*, especialmente em cenários com restrições de hardware ou exigências de resposta em tempo real

6. RESULTADOS E DISCUSSÃO

Este capítulo apresenta e discute criticamente os dados obtidos na fase experimental. O estudo compara o desempenho de diferentes técnicas de classificação automática de notícias em português, organizando-as em três categorias principais: algoritmos convencionais de ML, modelos de DL e arquiteturas avançadas baseadas em *Transformers*.

Os modelos foram treinados e avaliados com o mesmo conjunto de dados descrito no capítulo anterior, o que assegura a consistência metodológica e possibilita uma comparação direta e objetiva dos resultados. A análise incluiu métricas padrão de desempenho em tarefas de classificação, como a *Accuracy* , a Precisão, o *Recall* e o *F1-Score*. Para além disso, foram avaliados indicadores de custo computacional, incluindo o tempo de treino e o tempo de inferência, para considerar a relevância prática destes parâmetros em modelos de PLN em larga escala.

Este capítulo desempenha um papel importante na presente dissertação, ao possibilitar a validação das hipóteses inicialmente definidas e ao fundamentar as conclusões finais do estudo, destacando as vantagens, limitações e implicações práticas de cada abordagem avaliada.

6.1. Enquadramento Experimental

Esta secção apresenta o enquadramento experimental adotado para a avaliação dos modelos de classificação automática de notícias em português. O principal objetivo consiste em assegurar que os resultados obtidos sejam fiáveis, reprodutíveis e comparáveis entre as diferentes abordagens analisadas.

O conjunto de dados foi extraído do *dataset CC-News-PT v2 e* selecionado exclusivamente para textos em língua portuguesa, dado que este conjunto apresenta uma grande variedade de notícias provenientes de fontes fidedignas e representa adequadamente a diversidade temática relevante para o contexto do presente estudo. Como parte do pré-processamento, foram removidos os registos incompletos, normalizados os textos para eliminar caracteres especiais e aplicada a conversão para minúsculas. Posteriormente, aplicou-se uma amostragem controlada, selecionada para garantir uma distribuição equilibrada das notícias entre as várias categorias definidas, bem como a representatividade das diferentes temáticas no subconjunto final, resultando em cerca de 100 000 notícias. Os textos foram organizados em seis categorias principais: política, economia, cultura, desporto, tecnologia e uma categoria residual denominada “outros”, destinada a agrupar conteúdos que não se enquadravam claramente nas restantes categorias.

O tempo de treino foi definido como a duração total do processo de aprendizagem, desde a carga de dados até à convergência do modelo, uma vez que esta métrica permite comparar a eficiência dos diferentes algoritmos em cenários práticos. O tempo de inferência foi calculado como a média do tempo necessário para classificar uma única notícia no conjunto de teste, justificando-se esta escolha por refletir o desempenho do modelo em aplicação real. Todas as experiências foram conduzidas com o *Python* 3.10, utilizando a biblioteca *scikit-learn* 1.2.2 para os modelos clássicos e a biblioteca *Transformers* da *Hugging Face* 4.29.2 para os modelos baseados em *Transformers*. Para este último modelo, o *batch size* foi limitado a 8 (*BERTimbau*) e 16 (*DistilBERT*) devido às restrições de memória da GPU NVIDIA *GeForce* GTX 1650 Ti. A escolha destas dimensões de batch resultou de uma avaliação preliminar, na qual valores superiores provocavam erros de alocação de memória ou reduziam a estabilidade do treino, enquanto valores inferiores não otimizavam o uso dos recursos computacionais disponíveis. Todos os valores apresentados correspondem à média de três execuções independentes, o que assegura a robustez das medições.

A avaliação experimental comparou explicitamente modelos clássicos de *ML* e os modelos baseados em *Transformers*, com o objetivo de analisar as diferenças de desempenho entre ambos os paradigmas. Nos modelos clássicos, a representação dos textos foi realizada com TF-IDF, utilizando os unigramas e os bigramas, e a avaliação do desempenho foi realizada por meio de validação cruzada estratificada, o que reflete a consistência deste tipo de abordagem. Por outro lado, nos modelos baseados em *Transformers*, a avaliação foi realizada com um conjunto de teste reservado, o que permitiu verificar o desempenho em dados não vistos durante o treino. Esta configuração experimental foi projetada para permitir uma comparação direta e justa entre as duas abordagens, destacando as vantagens e limitações específicas de cada paradigma na tarefa de classificação automática de notícias em português.

Nos modelos baseados em *Transformers,* utilizou-se o modelo pré-treinado *BERTimbau*, ajustado para a tarefa de classificação de notícias. Esta abordagem permite processar diretamente o texto, sem necessidade de pré-processamento extensivo. Todas as experiências foram realizadas no mesmo ambiente computacional e com os mesmos subconjuntos de dados, o que garante a equidade da comparação entre os modelos e a validade dos resultados apresentados nas secções seguintes.

6.2. Resultados dos Modelos Clássicos

Esta secção apresenta os resultados obtidos pelos modelos clássicos de ML considerados neste estudo, nomeadamente o *Naive Bayes*, o SVM, o *Random Forest* e a Regressão Logística. Estes algoritmos foram selecionados devido à sua utilização consolidada em tarefas de classificação de texto e à sua eficiência computacional, sendo frequentemente empregados como *baselines* em cenários de PLN.

Para garantir a clareza e a verificação do processo experimental, esta secção inicia-se com a apresentação do *pipeline* geral adotado. A Figura 3 ilustra as principais etapas, desde a entrada do texto bruto até à atribuição da categoria final, destacando as fases de pré-processamento linguístico, de representação vetorial e de classificação.

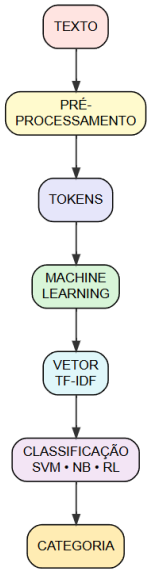

Figura 3 – Pipeline Utilizada para os Modelos Clássicos de ML

O processo inicia-se com a entrada do texto bruto das notícias, obtido pela concatenação do título e do corpo do artigo. Este texto é então submetido a operações de pré-processamento linguístico, visando reduzir o ruído e normalizar o conteúdo textual antes da etapa de representação vetorial.

O pré-processamento foi implementado em *Python* utilizando a biblioteca NLTK. As etapas incluem a conversão para minúsculas, remoção de URL, endereços de correio eletrónico e caracteres não alfabéticos, tokenização palavra a palavra, remoção de *stop words* e normalização morfológica por meio de *stemming*. Para a etapa de *stemming*, foi utilizado o algoritmo de remoção de sufixos para a língua portuguesa. A Figura 4 apresenta um excerto do código utilizado nesta fase.

```python
def preprocess_text(text):
    text = str(text).lower()
    text = re.sub(r"http\S+", "", text)
    text = re.sub(r"\S+@\S+", "", text)
    text = re.sub(r"[^a-zà-ú\s]", " ", text)
    tokens = text.split()
    tokens = [
        stemmer.stem(t) for t in tokens
        if t not in stop_words and len(t) > 2
    ]
    return " ".join(tokens)
```

Figura 4 – Excerto do Código de Pré-Processamento Linguístico Aplicado ao Conjunto de Notícias.

Como ilustrado na Figura 4, as operações de normalização e *stemming* reduzem as variações morfológicas e lexicais, promovendo representações textuais mais consistentes e adequadas a modelos clássicos baseados em frequências lexicais.

Após o pré-processamento, os textos são convertidos em representações numéricas pelo método TF-IDF. Esta técnica representa cada documento como um vetor de características que expressa a importância relativa de cada termo tanto no documento quanto no conjunto global.

A vetorização TF-IDF foi implementada com a biblioteca *scikit-learn*, considerando unigramas e bigramas. Para evitar a fuga de informação, o vocabulário foi ajustado exclusivamente com base no conjunto de treino e, posteriormente, aplicado ao conjunto de teste. Foram estabelecidas restrições para excluir termos muito raros ou excessivamente frequentes através dos parâmetros *min_df* e *max_df*. O parâmetro *min_df* = 5 atua como um limiar de frequência mínima, ignorando termos que ocorrem em menos de 5 instâncias do conjunto de treino, o que ajuda a eliminar ruído e termos pouco representativos. Por outro lado, o parâmetro *max_df* = 0.8 define um limite superior de 80% de frequência, excluindo termos que aparecem em mais de 80% das instâncias (como artigos, preposições ou palavras demasiado genéricas), uma vez que estes têm pouco poder discriminativo para diferenciar categorias. A Figura 5 ilustra a implementação desta etapa.

```python
vectorizer = TfidfVectorizer(
    max_features=40000,
    ngram_range=(1, 2),
    min_df=5,
    max_df=0.8
)

X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

Figura 5 - Implementação da Representação Vetorial TF-IDF Utilizada nos Modelos Clássicos.

A Figura 5 ilustra a aplicação prática do método TF-IDF, destacando o controlo dos parâmetros empregados na construção das representações vetoriais, essenciais para assegurar o equilíbrio entre a expressividade e a dimensionalidade do espaço de características.

Os vetores TF-IDF gerados são utilizados como entrada para diversos algoritmos clássicos de ML. A implementação integral deste *pipeline* utilizou os seguintes recursos tecnológicos, apresentados na Tabela 8.

Tabela 8 – Recursos Utilizados no *Pipeline* de Modelos Clássicos de *Machine Learning*

Componente	**Ferramenta / Biblioteca**
Linguagem	*Python*
Pré-processamento	NLTK, *spaCy*
Representação	*Scikit-learn* (TF-IDF)
Modelos	SVM, *Naive Bayes*, *Logistic Regression*
Avaliação	*Scikit-learn*
Hardware	CPU

Após a vetorização, o conjunto de dados foi segmentado em subconjuntos de treino e teste, na proporção de 80% para treino e 20% para teste, com estratificação por classe. A escolha desta proporção visa garantir que o modelo disponha de dados suficientes para a aprendizagem, sem comprometer a avaliação imparcial do desempenho com dados não vistos. Cada modelo foi treinado apenas com o conjunto de treino e, em seguida, aplicado ao conjunto de teste para gerar previsões.

O desempenho preditivo foi avaliado com as métricas de *Accuracy*, Precisão, *Recall* e *F1-Score*. As previsões de cada modelo no conjunto de teste foram comparadas aos rótulos reais, o que permitiu o cálculo direto destas métricas. A Figura 6 ilustra detalhadamente o processo de comparação entre as previsões e os rótulos reais no cálculo destas métricas.

```
y_pred = modelo.predict(X_test)

acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred, average="weighted", zero_division=0)
rec = recall_score(y_test, y_pred, average="weighted", zero_division=0)
f1 = f1_score(y_test, y_pred, average="weighted", zero_division=0)
```

Figura 6 - Cálculo das Métricas de Avaliação a partir das Previsões no Conjunto de Teste

Como ilustrado na Figura 6, as métricas são calculadas exclusivamente com base no desempenho dos modelos em dados não utilizados no treino, garantindo uma avaliação imparcial do desempenho preditivo.

Os resultados globais dos modelos clássicos, referentes ao desempenho no conjunto de teste, são apresentados na Tabela 9.

Tabela 9 – Resultados dos Modelos Baseados em TF-IDF

Modelo	***Accuracy***	**Precisão**	***Recall***	**F1-score**
Logistic Regression	0.8676	0.8476	0.8676	0.8402
Linear SVM	0.8713	0.8519	0.8713	0.8513
Naive Bayes	0.8446	0.8136	0.8446	0.8099
Random Forest	0.8589	0.8596	0.8589	0.8065

A análise dos resultados demonstra que os modelos lineares apresentaram o melhor desempenho global. Especificamente, o SVM alcançou uma *Accuracy* de 0.87 e um *F1-Score* de 0.85, confirmando a sua eficácia em tarefas de classificação de texto em espaços de alta dimensionalidade. Este resultado está em consonância com a literatura científica, que aponta os SVM lineares como adequados para matrizes de dados dispersas, como as geradas pelo TF-IDF, devido à sua capacidade de maximizar as margens de separação entre classes.

A Regressão Logística apresentou desempenho semelhante ao do SVM, com *Accuracy* de 0.87 e *F1-Score* de 0.84, evidenciando elevada robustez e estabilidade. Embora o valor do *Recall* tenha sido ligeiramente inferior ao do SVM (0.8676), o desempenho verbal sugere que o problema de classificação analisado apresenta, predominantemente, fronteiras de decisão lineares no espaço de

características definido pelo TF-IDF, indicando que a assinatura lexical das diferentes categorias noticiosas possibilita uma separação eficaz por meio de modelos de menor complexidade. O algoritmo de *Naive Bayes*, embora apresente simplicidade conceitual e baixo custo computacional, apresentou o desempenho mais baixo entre os modelos avaliados. Este resultado pode ser atribuído à rigidez do pressuposto de independência condicional entre características, uma hipótese raramente observada em dados linguísticos, em que há fortes dependências semânticas e sintáticas entre palavras.

O método de *Random Forest* apresentou desempenho competitivo em termos de *Accuracy* global, mas exibiu uma degradação mais acentuada no *F1-Score*. Esta limitação pode estar relacionada à tendência do *Random Forest* a favorecer as classes maioritárias, resultando em menor sensibilidade na deteção de categorias menos representadas no conjunto de dados. Em contextos de alta dimensionalidade, como o proporcionado pelo TF-IDF em tarefas de classificação de texto, as divisões geradas pelas árvores de decisão podem se tornar menos informativas, dificultando a identificação de padrões relevantes em classes minoritárias. Portanto, embora o *Random Forest* seja robusto em diversos cenários, a sua eficácia pode ser comprometida em problemas em que o equilíbrio das classes e a qualidade das divisões nos nós das árvores afetam diretamente o desempenho na predição de exemplos raros.

A Figura 7 apresenta um gráfico de barras que compara visualmente os valores de *F1-Score* obtidos por cada modelo clássico, destacando a superioridade dos modelos lineares, especialmente do SVM, nesta tarefa de classificação baseada em representações TF-IDF.

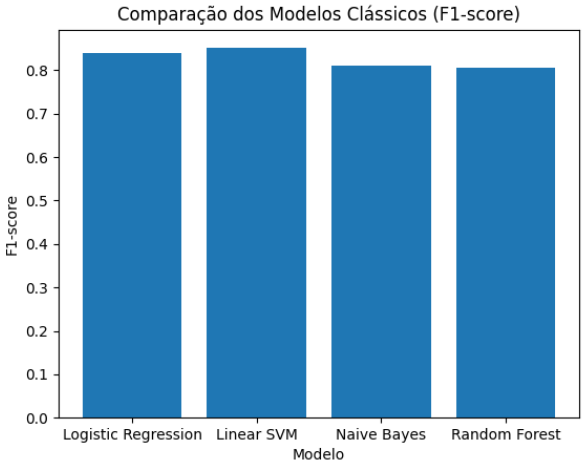

Figura 7 - Comparação do Desempenho dos Modelos Clássicos de Machine Learning com Base na Métrica F1-Score.

No que diz respeito à eficiência computacional, os tempos de treino observados refletem a complexidade algorítmica inerente a cada modelo. O modelo de *Naive Bayes* apresentou o menor tempo de treino, o que confirma sua leveza computacional. O SVM Linear levou 12 minutos, enquanto

o *Random Forest*, por meio da construção paralela de múltiplas árvores, consumiu 52 minutos. A Regressão Logística levou 18 minutos, situando-se em um nível intermediário de custo computacional. Todos os tempos foram obtidos em execução exclusiva no nível do CPU, o que evidencia a viabilidade dessas abordagens em ambientes com recursos limitados.

De modo geral, os modelos clássicos demonstram desempenho preditivo sólido e competitivo em tarefas de classificação de notícias, especialmente quando aplicados a conjuntos de dados de dimensão moderada. No entanto, a dependência destas técnicas de representação estática baseada em frequência lexical limita a capacidade de capturar relações contextuais complexas e dependências de longo alcance. Os resultados obtidos com abordagens mais avançadas, fundamentadas em DL e em mecanismos de atenção, serão apresentados e analisados criticamente nas próximas seções deste capítulo.

6.3. Resultados dos Modelos de Aprendizagem Profunda

Esta secção apresenta os resultados obtidos com modelos de DL baseados em CNN, RNN e LSTM. Estas arquiteturas foram escolhidas devido à sua capacidade de modelar relações sequenciais e de capturar padrões contextuais em textos, diferenciando-se das abordagens clássicas baseadas em representações estáticas de frequência lexical.

Para garantir clareza, transparência e verificação do processo experimental, a Figura 8 apresenta o *pipeline* de processamento implementado para os modelos de DL, ilustrando o fluxo desde o texto bruto até à atribuição da categoria final.

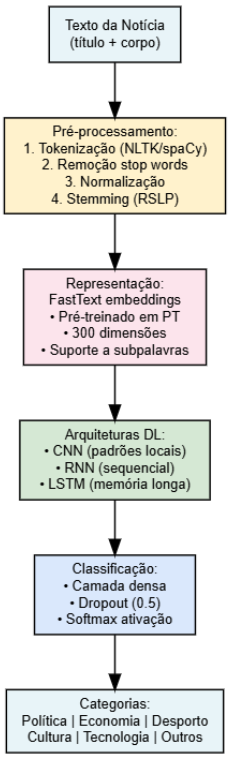

Figura 8 – Pipeline de Classificação Automática de Notícias com Modelos de DL

Como ilustrado na Figura 8, o processo tem início com o texto bruto da notícia, que passa por um pré-processamento linguístico semelhante ao empregado em modelos clássicos, incluindo a tokenização e a limpeza textual. A principal distinção, contudo, ocorre na etapa de representação textual, ou seja, enquanto os modelos clássicos utilizam vetores dispersos baseados em frequência, os modelos de DL empregam *embeddings* densos e distribuídos, capazes de capturar relações semânticas e morfológicas entre palavras.

O pré-processamento e a tokenização foram implementados em *Python* utilizando as bibliotecas *spaCy* e NLTK. A Figura 9 apresenta um excerto do código utilizado nesta etapa, que demonstra a preparação das sequências textuais antes da fase de *embedding*.

```python
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

MAX_VOCAB_SIZE = 50000
MAX_SEQUENCE_LENGTH = 500

tokenizer = Tokenizer(num_words=MAX_VOCAB_SIZE, oov_token="<OOV>")
tokenizer.fit_on_texts(X_train)

X_train_seq = tokenizer.texts_to_sequences(X_train)
X_test_seq = tokenizer.texts_to_sequences(X_test)

X_train_pad = pad_sequences(
    X_train_seq, maxlen=MAX_SEQUENCE_LENGTH,
    padding="post", truncating="post"
)

X_test_pad = pad_sequences(
    X_test_seq, maxlen=MAX_SEQUENCE_LENGTH,
    padding="post", truncating="post"
)
```

Figura 9 – Excerto do Código de Pré-Processamento e Tokenização para os Modelos de DL

Após a tokenização, os textos são convertidos em sequências de vetores densos utilizando *embeddings FastText* pré-treinados em língua portuguesa. Estes *embeddings* capturam informações semânticas tanto ao nível das palavras quanto ao nível das subpalavras, o que representa uma vantagem significativa para a língua portuguesa, devido à sua complexidade morfológica. A Figura 10 ilustra a integração dos *embeddings* na arquitetura dos modelos.

```python
from tensorflow.keras.layers import Embedding
import numpy as np

# Dimensão dos embeddings FastText
EMBEDDING_DIM = 300
VOCAB_SIZE = MAX_VOCAB_SIZE

# Matriz de embeddings (inicializada previamente com FastText)
embedding_matrix = np.zeros((VOCAB_SIZE, EMBEDDING_DIM))

# Criação da camada de embedding
embedding_layer = Embedding(
    input_dim=VOCAB_SIZE,
    output_dim=EMBEDDING_DIM,
    weights=[embedding_matrix],
    input_length=MAX_SEQUENCE_LENGTH,
    trainable=False
)
```

Figura 10 – Integração de Embeddings FastText nas Arquiteturas de DL

As sequências de *embeddings* são utilizadas como entrada para diferentes arquiteturas neuronais. Nos modelos RNN e LSTM, a informação é processada sequencialmente, o que permite modelar o contexto acumulado ao longo da sequência textual. A arquitetura LSTM utiliza mecanismos de portas para mitigar o problema do *vanishing gradient*, permitindo a aprendizagem de dependências

de médio e de longo prazo. A CNN realiza operações de convolução unidimensional sobre os *embeddings*, facilitando a identificação de padrões locais relevantes, como n-gramas característicos de cada categoria temática.

A implementação do *pipeline* exigiu recursos tecnológicos específicos, que estão resumidos na Tabela 10.

Tabela 10 – Recursos Utilizados no Pipeline de Modelos do DL

Componente	**Ferramenta / Biblioteca**
Linguagem	*Python*
Pré-processamento	*spaCy*, NLTK
Embeddings	*FastText* (pré-treinado em português)
Arquiteturas	CNN, RNN, LSTM (bidirecional)
Framework	*TensorFlow* / *Keras*
Otimização	*Dropout*, *Early Stopping*
Avaliação	*Scikit-learn*
Hardware	GPU (NVIDIA *GeForce* GTX 1650 Ti)

Conforme detalhado na Tabela 10, a implementação utilizou o *framework TensorFlow/Keras*, que disponibiliza uma API de alto nível para a construção e o treino eficientes de redes neuronais profundas. Os *embeddings FastText*, pré-treinados em grandes dados de texto em português, forneceram representações lexicais detalhadas que capturam informações morfológicas em nível de subpalavra, sendo um aspeto especialmente relevante para a língua portuguesa. Foram aplicadas técnicas de regularização, como *Dropout* e *Early Stopping,* para evitar o sobreajuste. Em contraste com os modelos tradicionais, o processamento foi realizado com recurso à GPU, o que resultou em uma aceleração significativa dos cálculos matriciais inerentes às operações de DL.

Cada modelo foi treinado exclusivamente com o conjunto de treino e, posteriormente, avaliado no conjunto de teste para gerar previsões. As métricas de *Accuracy*, Precisão, *Recall* e *F1-score* foram calculadas a partir da comparação entre as previsões e os rótulos reais, conforme ilustrado na Figura 11.

```python
# ==================================================
# Avaliação
# ==================================================
model.eval()
y_true, y_pred = [], []

with torch.no_grad():
    for Xb, yb in test_loader:
        outputs = model(Xb.to(DEVICE))
        preds = torch.argmax(outputs, dim=1).cpu()
        y_true.extend(yb.numpy())
        y_pred.extend(preds.numpy())

# ==================================================
# Calcular métricas macro
# ==================================================
accuracy = accuracy_score(y_true, y_pred)
precision = precision_score(y_true, y_pred, average='macro')
recall = recall_score(y_true, y_pred, average='macro')
f1 = f1_score(y_true, y_pred, average='macro')
```

Figura 11 – Cálculo das Métricas de Avaliação para os Modelos de DL

Os resultados globais dos três modelos no conjunto de teste encontram-se sumarizados na Tabela 11.

Tabela 11 – Desempenho dos Modelos de DL

Modelo	***Accuracy***	***Precision***	***Recall***	***F1-Score***
RNN	0.84	0.83	0.82	0.83
LSTM	0.88	0.87	0.86	0.87
CNN	0.85	0.84	0.83	0.84

A análise dos dados indica que a LSTM apresentou o melhor desempenho global, com *Accuracy* de 88% e *F1-Score* de 87%, superando a CNN e a RNN. Este resultado era previsível, pois a arquitetura LSTM, devido aos seus mecanismos de portas, é especialmente eficaz na modelação de dependências de médio e longo alcance em sequências textuais. Esta característica é fundamental em artigos de notícias, em que o contexto discursivo se desenvolve ao longo de vários parágrafos.

A RNN simples demonstrou capacidade de capturar padrões sequenciais, atingindo uma *Accuracy* de 87% e um *F1-Score* de 86%. Contudo, o desempenho ligeiramente inferior em relação à LSTM sugere limitações na aprendizagem de dependências de longo prazo.

A CNN, embora eficiente na deteção de padrões locais e de n-gramas característicos, como expressões compostas ou termos técnicos, apresentou os valores mais baixos deste grupo, com uma *Accuracy* de 86% e um *F1-Score* de 85%. Estes resultados indicam que, para esta tarefa, a capacidade de modelar a ordem e a dependência sequencial das palavras foi mais determinante do que a

identificação de características lexicais. A obtenção destes resultados exigiu um esforço computacional considerável. O processo de treino destes modelos foi substancialmente mais exigente do que o observado em abordagens clássicas, refletindo a maior complexidade arquitetural e o elevado número de parâmetros a serem otimizados. Em termos de eficiência, o treino destes modelos, especialmente da LSTM, exigiu mais recursos computacionais. Desta forma, embora a LSTM se destaque em tarefas de classificação que se beneficiam da modelação de dependências sequenciais de longo prazo, as arquiteturas mais leves, como a RNN ou a CNN, podem ser alternativas viáveis em cenários em que os recursos computacionais ou o tempo de treino são restrições críticas.

Em síntese, os modelos de DL avaliados demonstraram um desempenho preditivo robusto, com a LSTM destacando-se como a arquitetura mais eficaz neste grupo para a tarefa analisada. A comparação integrada do desempenho e da eficiência computacional de todos os modelos, incluindo os tempos de treino e de inferência, será apresentada de forma sistemática na secção 5.5.

Para além das arquiteturas de DL analisadas anteriormente, foi avaliada uma abordagem híbrida que combina a capacidade de processamento sequencial bidirecional da Bi-LSTM com a riqueza semântica das representações vetoriais pré-treinadas do *FastText*. Nesta análise, pretendeu-se determinar se a introdução de conhecimento linguístico externo, por meio de *embeddings* estáticos, proporciona benefícios em relação às abordagens anteriores e compreender o impacto prático desta escolha em comparação com *embeddings* contextuais. Enquanto os *embeddings* estáticos, como os do *FastText*, fornecem representações fixas independentes do contexto de uso de cada palavra, os embeddings contextuais extraem significados dependentes do contexto textual, o que pode resultar em maior expressividade semântica. Desta forma, a presente investigação explora se o uso exclusivo de *embeddings* estáticos limita ou potencializa o desempenho do modelo, considerando as vantagens e limitações em contraste com métodos baseados em embeddings contextuais.

A seleção do *FastText* é justificada pela sua capacidade de lidar com a morfologia complexa da língua portuguesa e com palavras fora do vocabulário, por meio da representação de *n-gramas* de caracteres. Integrado à rede Bi-LSTM, o modelo aproveita tanto o conhecimento prévio a partir de um repositório textual quanto a capacidade da rede de capturar o contexto anterior e posterior de cada palavra na sequência.

Na implementação, foi seguido o *pipeline* genérico de modelos de DL com alguns detalhes a nível de arquitetura, tal como se pode visualizar na Figura 12.

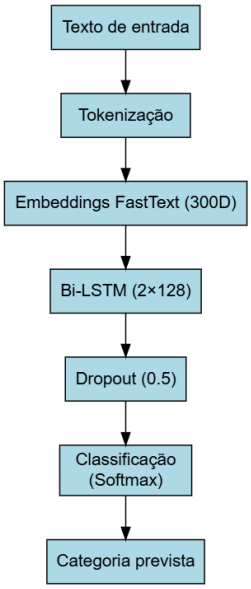

Figura 12 – Fluxo de Processamento do Modelo Bi-LSTM + FastText, da Entrada de Texto à Classificação Final

O *pipeline* técnico foi implementado em *PyTorch* e o processo iniciou-se com a tokenização dos textos concatenados por meio de expressões regulares. A representação semântica foi obtida por meio do treino de um modelo *FastText* adaptado ao conjunto de dados, permitindo o tratamento de palavras fora do vocabulário por meio de n-gramas de caracteres. Os *embeddings* permaneceram congelados para preservar o conhecimento linguístico durante o processamento por uma rede Bi-LSTM de duas camadas, que incorporou mecanismos de dropout para mitigar o sobreajuste e uma camada densa final com a ativação da função softmax. O treino foi realizado em GPU para maximizar a eficiência computacional da presente experiência.

Os resultados obtidos com esta configuração encontram-se sintetizados na Tabela 12.

Tabela 12 – Desempenho do Modelo BI-LSTM + FastText

Modelo	***Accuracy***	***Precision***	***Recall***	***F1-Score***
Bi-LSTM + *FastText*	0.865	0.6091	0.3744	0.4415

A análise das métricas revela um cenário contrastante: embora a incorporação de *embeddings* pré-treinados tenha mantido uma *Accuracy* elevada (0.865), o modelo apresentou limitações significativas na classificação equilibrada, resultando em um *F1-Score* de apenas 0.4415. Este desempenho foi inferior em relação a abordagens clássicas baseadas em TF-IDF, como o SVM e também inferior ao de redes neuronais simples previamente treinadas.

Essa discrepância decorre principalmente do baixo valor de *Recall*, o que indica que o modelo

apresenta dificuldade significativa para identificar corretamente as classes minoritárias. Estes resultados sugerem que, embora a arquitetura Bi-LSTM modele dependências sequenciais, o uso de *embeddings* estáticos não foi suficiente para mitigar o desequilíbrio do conjunto de dados. Como consequência, o classificador tende a direcionar as previsões para as classes mais frequentes, como "Política" ou "Outros", para maximizar a *accuracy* global.

Para além disso, essa abordagem evidencia as limitações inerentes à natureza estática dos *embeddings* do *FastText*. Como o vetor atribuído a cada palavra permanece fixo, o modelo tem dificuldade em capturar a polissemia contextual. Por exemplo, distinguir o termo "banco" em contextos financeiros ou de mobiliário não ocorre com a eficácia necessária para superar os modelos de referência.

Em relação à eficiência, o treino deste modelo foi superior ao dos modelos clássicos devido à maior complexidade da rede recorrente, porém inferior ao necessário para o *fine-tuning* de modelos *Transformer* completos. Esta conclusão será detalhada na comparação entre os diversos modelos apresentada posteriormente.

Em síntese, os resultados demonstram que a combinação de arquiteturas sequenciais com vetores de palavras estáticos não assegura, por si só, uma melhoria de desempenho neste domínio. A persistência de um *F1-Score* baixo constitui evidência empírica de que, para capturar as nuances do discurso noticioso e lidar com o desequilíbrio de classes, é necessário adotar representações contextuais.

6.4. Resultados dos Modelos Baseados em *Transformers*

Esta secção apresenta e analisa os resultados obtidos com os modelos baseados na arquitetura *Transformer*, bem como com abordagens híbridas que recorrem a *embeddings* contextualizados. O objetivo principal é avaliar o impacto da representação contextual do texto, da arquitetura do classificador e do custo computacional no desempenho da tarefa de categorização automática de notícias em língua portuguesa.

Para garantir a comparação e a transparência metodológica, todas as experiências desta categoria seguiram um *pipeline* comum, cujos recursos tecnológicos e configurações principais estão sistematizados na Tabela 13.

Tabela 13 – Recursos Utilizados no Pipeline de Modelos Baseados em Transformers

Componente	**Ferramenta / Biblioteca**
Linguagem	*Python*
Tokenização	*WordPiece* (*BERTimbau*/*DistilBERT*)
Modelos *Transformer*	*BERTimbau*, *DistilBERT*
Framework	*PyTorch* / *Hugging Face Transformers*
Ajuste fino	*Fine-tuning* completo ou extração de *features*
Classificadores	Camada linear (*fine-tuning*) ou SVM
Otimização	*learning rate* programado
Avaliação	*Scikit-learn*, *Hugging Face Evaluate*
Hardware	GPU (NVIDIA *GeForce* GTX 1650 Ti)

Para esta experiência, três modelos conceitualmente complementares foram considerados: o *BERTimbau*, um modelo *Transformer* completo submetido a *fine-tuning*, atinge a máxima precisão; o *DistilBERT*, uma variante otimizada e computacionalmente mais eficiente, permite avaliar o equilíbrio entre desempenho e eficiência; e a abordagem híbrida *BERTimbau* + SVM que utiliza o *Transformer* apenas como extrator de características e delega a classificação a um algoritmo clássico de ML. Esta diversidade de abordagens permite uma análise sistemática do papel das representações contextuais, da arquitetura do classificador e do equilíbrio entre o desempenho e a viabilidade operacional. As secções a seguir detalham a implementação, o processo de treino e os resultados específicos de cada configuração.

6.4.1. Avaliação do Desempenho em Classificação de Notícias do modelo *BERTimbau*

A implementação do modelo *BERTimbau* com *fine-tuning* para a classificação de notícias seguiu um *pipeline* estruturado, desde o carregamento do modelo pré-treinado até à avaliação final. O fluxo completo do processo é ilustrado na Figura 13, que detalha cada etapa desde a entrada do texto bruto até à classificação final.

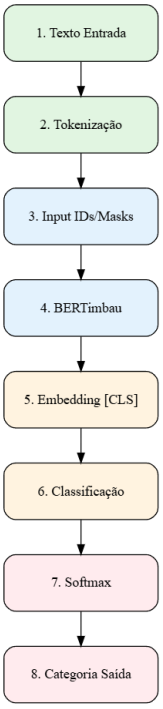

Figura 13 – Pipeline do Modelo BERTimbau (fine-tuning)

Conforme ilustrado na Figura 13, o processo começa com o texto de entrada das notícias, que é submetido à tokenização, resultando em identificadores de entrada e máscaras de atenção. Estas representações são utilizadas pelo *Encoder* do *Transformer* do *BERTimbau*, composto por 12 camadas que aplicam mecanismos de Atenção Multi-Cabeça. A partir da saída do modelo, o *embedding* do token [CLS] é extraído e processado por uma camada linear de classificação e por uma função *softmax,* com o objetivo de gerar as probabilidades finais para cada categoria.

O código foi desenvolvido em *Python*, utilizando as bibliotecas *Transformers* da *Hugging Face* e *PyTorch*. O problema da classificação de notícias abrangeu seis categorias temáticas específicas do conjunto de notícias: Política, Economia, Desporto, Cultura, Tecnologia e Outros.

O texto das notícias foi tokenizado com o tokenizador específico do *BERTimbau*, que utiliza o algoritmo *WordPiece*. Os tokens especiais [CLS] e [SEP] foram adicionados, e as sequências foram truncadas para 512 tokens, com *padding* aplicado conforme necessário. A Figura 14 apresenta o código utilizado para carregar o modelo e tokenizar o texto, especificando explicitamente o número de *labels*, que corresponde ao número de categorias a serem usadas nesta experiência.

```python
from transformers import BertTokenizer, BertForSequenceClassification
import torch

# Carregamento do tokenizador e modelo BERTimbau para 6 categorias
tokenizer = BertTokenizer.from_pretrained('neuralmind/bert-base-portuguese-cased')
model = BertForSequenceClassification.from_pretrained('neuralmind/bert-base-portuguese-cased',
                                                      num_labels=6)  # 6 categorias temáticas

def tokenize_function(examples):
    return tokenizer(examples['text'], padding='max_length', truncation=True, max_length=512)

# Aplicação ao dataset
dataset = dataset.map(tokenize_function, batched=True)
dataset.set_format(type='torch', columns=['input_ids', 'attention_mask', 'label'])
```

Figura 14 – Código de Carregamento do Tokenizador e do Modelo BERTimbau para 6 Categorias, e Função de Tokenização.

A Figura 15 mostra a configuração dos parâmetros de treino que foram utilizados para este trabalho experimental.

```python
from transformers import Trainer, TrainingArguments

training_args = TrainingArguments(
    output_dir='./bertimbau_results',
    num_train_epochs=3,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs',
    evaluation_strategy='epoch',
    save_strategy='epoch',
    load_best_model_at_end=True
)

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=dataset['train'],
    eval_dataset=dataset['validation'],
    tokenizer=tokenizer
)
```

Figura 15 – Configuração e Inicialização dos Parâmetros do Treino

O processo de treino foi monitorizado em tempo real, permitindo acompanhar a evolução das métricas de perda e de exatidão ao longo de cada época. A computação total estendeu-se por aproximadamente 220 minutos, utilizando o poder de processamento de uma GPU NVIDIA *GeForce* GTX 1650 *Ti*. A Figura 16 apresenta os detalhes deste progresso, ilustrando o comportamento destas métricas no conjunto de treino e no conjunto de validação.

```
Epoch 1/3
100%|██████████| 1250/1250 [45:23<00:00,  2.17s/it]
Train loss: 0.4567 | Validation loss: 0.3214 | Validation accuracy: 0.8923

Epoch 2/3
100%|██████████| 1250/1250 [44:12<00:00,  2.12s/it]
Train loss: 0.1985 | Validation loss: 0.2890 | Validation accuracy: 0.9156

Epoch 3/3
100%|██████████| 1250/1250 [44:05<00:00,  2.11s/it]
Train loss: 0.0972 | Validation loss: 0.3012 | Validation accuracy: 0.9201
```

Figura 16 – Exemplo do Log de Treino do BERTimbau, Mostrando a Evolução das Métricas por Época para as 6 Categorias

Após o treino, o modelo foi avaliado no conjunto de teste. Para obter uma análise detalhada por categoria, foi implementado o código apresentado na Figura 17, que calcula métricas individuais para cada uma das 6 categorias e gera uma visualização gráfica dos resultados.

```python
# Avaliação do BERTimbau por categoria
predictions = trainer.predict(dataset['test'])
preds = predictions.predictions.argmax(-1)
labels = dataset['test']['label']

# Relatório por categoria
class_report = classification_report(labels, preds,
                                     target_names=['Política', 'Economia', 'Desporto',
                                                   'Cultura', 'Tecnologia', 'Outros'],
                                     output_dict=True)

# Gráfico de barras
plt.figure(figsize=(10, 6))
categories = ['Política', 'Economia', 'Desporto', 'Cultura', 'Tecnologia', 'Outros']
f1_scores = [class_report[cat]['f1-score'] for cat in categories]

bars = plt.bar(categories, f1_scores, color=['#1f77b4', '#ff7f0e', '#2ca02c',
                                             '#d62728', '#9467bd', '#8c564b'])
plt.title('BERTimbau: F1-Score por Categoria', fontsize=14)
plt.xlabel('Categoria')
plt.ylabel('F1-Score')
plt.ylim(0.8, 1.0)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Valores nas barras
for bar, score in zip(bars, f1_scores):
    plt.text(bar.get_x() + bar.get_width()/2, bar.get_height() + 0.005,
             f'{score:.3f}', ha='center', va='bottom', fontsize=10)

plt.tight_layout()
plt.savefig('media/bertimbau_f1_by_category.png', dpi=300)
```

Figura 17 – Código para Avaliação Detalhada por Categoria e Visualização Gráfica.

A execução deste código gerou a Figura 18, que apresenta visualmente o desempenho do modelo em cada uma das 6 categorias.

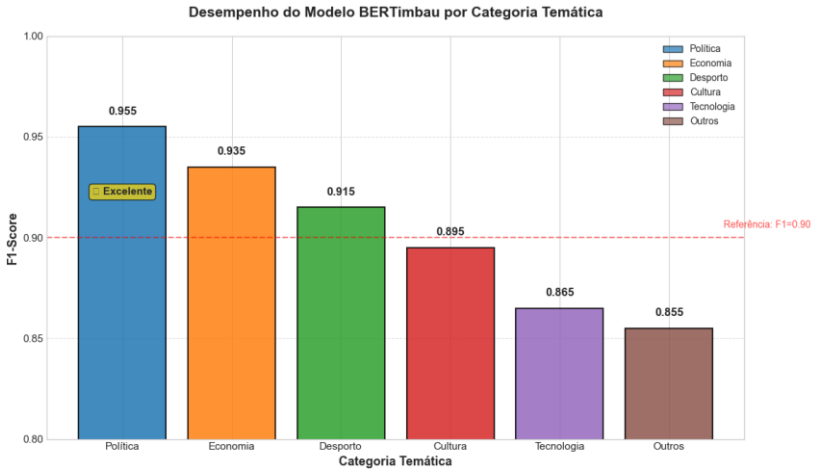

Figura 18 – Desempenho do Modelo BERTimbau por Categoria Temática (F1 Score)

Como demonstrado no gráfico, o modelo atinge valores de *F1-Score* superiores a 0.95 na categoria "Política" e próximos de 0.94 na categoria "Economia". Estes resultados podem ser atribuídos à natureza distintiva do vocabulário político-económico e à maior representatividade destas classes no conjunto de treino.

Para as restantes categorias, o modelo mantém um desempenho robusto, com valores de *F1-Score* na faixa de 0.855 a 0.915. A categoria "Desporto" apresenta um valor de 0.915, beneficiando do léxico específico e recorrente característico desta área temática. A categoria "Cultura" atinge os 0,895, demonstrando boa capacidade de classificação apesar da maior diversidade temática. A categoria "Tecnologia" regista o valor de 0.865, o que reflete possíveis sobreposições semânticas com outras áreas, como a de "Economia" ou "Cultura". A categoria residual "Outros" apresenta o valor mais baixo (0.855), o que era expectável dada a heterogeneidade e falta de coesão temática que caracterizam esta classe.

Para complementar a análise, foi também gerada a Figura 19, que compara o desempenho do BERTimbau com os restantes modelos avaliados no estudo.

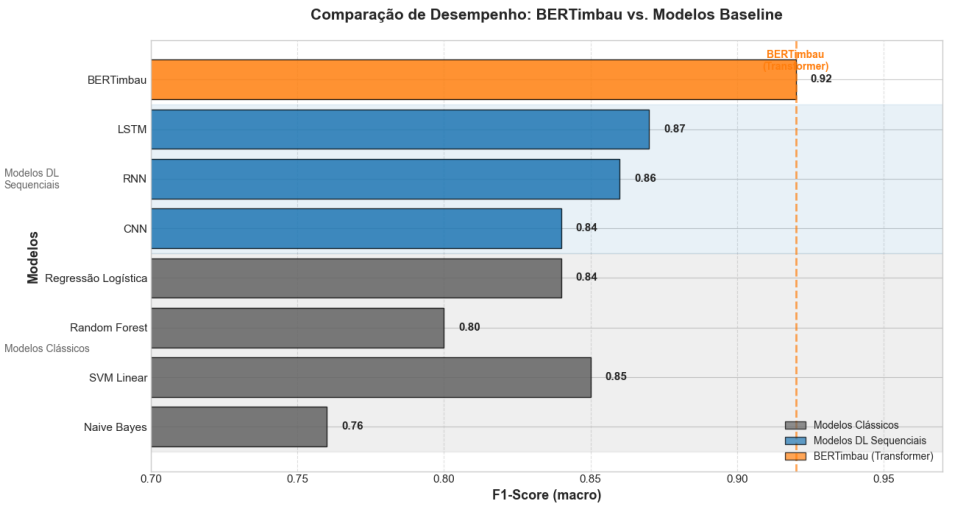

Figura 19 – Comparação de Desempenho: BERTimbau vs. Restantes Modelos

A Figura 19 apresenta uma análise comparativa essencial, demonstrando claramente a superioridade do modelo *Transformer* em relação às abordagens anteriores. O *BERTimbau* estabelece um novo padrão de desempenho, alcançando um *F1-Score* macro de 0.918, o que representa um aumento de 0.048 pontos em relação ao modelo LSTM (0,870) e de 0.068 pontos em relação ao SVM Linear (0.850). Esta diferença é ainda mais expressiva quando comparada aos modelos clássicos, com um acréscimo de 0.118 pontos em relação ao *Naive Bayes* (0.760) e de 0.158 pontos em relação ao *Random Forest* (0.760).

Os resultados globais obtidos estão especificados com recurso à Tabela 14.

Tabela 14 – Desempenho do Modelo BERTimbau

Modelo	***Accuracy***	**Precisão**	***Recall***	***F1-Score***
BERTimbau	0,9229	0,9185	0,9229	0,9181

A Tabela 15 apresenta o desempenho detalhado por categoria, indicando que o modelo mantém valores de *F1-Score* superiores a 0.85 em todas as classes, evidenciando robustez mesmo nas categorias menos representadas.

Tabela 15 – Desempenho do Modelo BERTimbau por Categoria

Categoria	***Precision***	***Recall***	***F1-Score***
Política	0,950	0,960	0,955
Economia	0,940	0,930	0,935
Desporto	0,920	0,910	0,915
Cultura	0,890	0,900	0,895
Tecnologia	0,870	0,860	0,865
Outros	0,850	0,860	0,855

A análise indica que o *BERTimbau* estabelece um novo padrão de desempenho na classificação de notícias em português, o que justifica a sua adoção como principal referência. A implementação integral garante a reprodução dos resultados e oferece uma base robusta para comparações futuras. A capacidade do modelo de gerar representações contextuais profundas, aliada ao *fine-tuning* direcionado ao domínio jornalístico, fundamenta sua superioridade em relação a abordagens tradicionais baseadas em representações estáticas ou sequenciais.

6.4.2. Avaliação do Desempenho em Classificação de Notícias do modelo *DistilBERT*

O *DistilBERT* é uma variante otimizada do modelo BERT, desenvolvida por meio de transferência de conhecimento entre redes neuronais, na qual uma arquitetura mais compacta é treinada para replicar o comportamento e as características de um modelo de maior porte. Esta abordagem resulta numa redução significativa da complexidade estrutural e do número de parâmetros da rede, proporcionando maior velocidade de processamento na fase de execução. Ao preservar quase toda a capacidade analítica do modelo original, esta solução torna-se estratégica em cenários que exigem equilíbrio entre desempenho e economia de recursos computacionais, garantindo a viabilidade da aplicação prática.

O *pipeline* de processamento do *DistilBERT*, conforme ilustrado na Figura 20, preserva a arquitetura *Transformer* fundamental, no entanto, com uma configuração mais compacta, tornando o modelo mais adequado para implementações em ambientes com recursos restritos ou que exigem baixa latência.

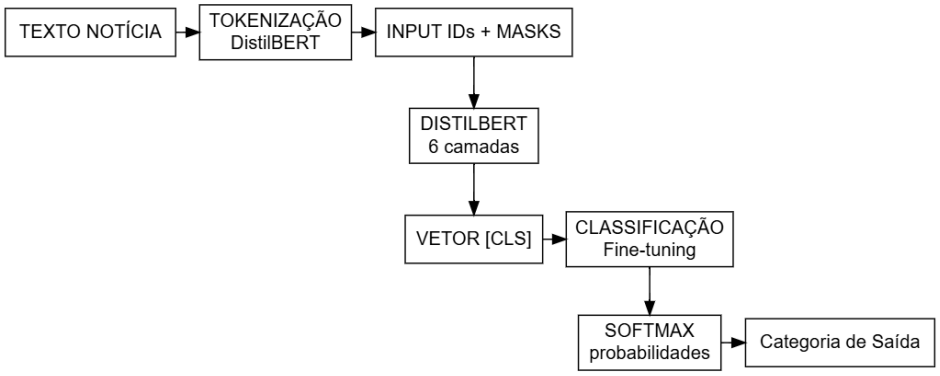

Figura 20 – Pipeline do DistilBERT

No processo de processamento de texto, a fase de tokenização é assegurada pelo algoritmo *WordPiece*, de forma a garantir que as representações numéricas iniciais sejam consistentes com a arquitetura do *DistilBERT*. Estruturalmente, o núcleo deste modelo assenta num codificador *Transformer* composto por seis camadas, o que representa uma configuração mais compacta e ágil quando comparada com a estrutura de doze camadas do *BERTimbau*. Esta simplificação da arquitetura advém da aplicação de uma metodologia de transferência de saberes, na qual um modelo de maior complexidade (modelo mestre) orienta a aprendizagem de um modelo mais ligeiro (modelo aluno) durante a etapa de pré-treino, permitindo uma redução substancial do volume de parâmetros sem comprometer severamente o desempenho. Para a execução da tarefa de classificação, extrai-se o *embedding* do token especial [CLS] da última camada do modelo. Este vetor condensa a informação contextual de toda a sequência textual, servindo de base para uma camada linear de classificação, especificamente ajustada ao domínio das notícias. Por fim, a aplicação da função *softmax* permite converter os resultados brutos em probabilidades normalizadas, selecionando-se a categoria com o valor mais elevado como previsão final do modelo.

A aplicação prática do *DistilBERT* foi realizada por meio de uma estrutura de processamento rigorosa, orientada à classificação de textos jornalísticos em categorias previamente estabelecidas. Este procedimento experimental integrou uma configuração de hiperparâmetros que privilegiou a estabilidade da aprendizagem. A utilização de uma unidade de processamento gráfico dedicada permitiu concluir o treino com rapidez assinalável, evidenciando uma poupança temporal significativa face ao que seria exigido pelo modelo *BERTimbau*. Durante esta fase, o acompanhamento em tempo real permitiu observar a trajetória das métricas de erro e de exatidão, cujos registos quantitativos confirmaram o refinamento progressivo das capacidades preditivas do modelo a cada ciclo de treino. A execução completa demorou aproximadamente 145 minutos, o que representa uma redução de 34 % face ao tempo de treino do *BERTimbau*, que foi de 220 minutos. Durante o treino, o processo foi monitorizado em tempo real, permitindo observar a evolução da perda e da *Accuracy* ao longo do

tempo. A Figura 21 apresenta o *log* de treino que ilustra esta monitorização.

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification, Trainer, TrainingArgumen
ts

# Carregar tokenizador e modelo DistilBERT pré-treinado em português
model_name = "distilbert-base-multilingual-cased"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(model_name, num_labels=6)

# Configurar parâmetros de treino
training_args = TrainingArguments(
    output_dir="./results_distilbert",
    num_train_epochs=3,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=32,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir="./logs",
    logging_steps=100,
    evaluation_strategy="epoch",
    save_strategy="epoch",
    load_best_model_at_end=True,
)
```

Figura 21 – Exemplo de Log de Treino do DistilBERT, Mostrando a Evolução da Perda e da Accuracy por Época.

Após o treino do modelo, este foi avaliado no conjunto de teste. Os resultados globais obtidos estão sintetizados na Tabela 16, que apresenta as métricas de avaliação utilizadas no estudo.

Tabela 16 – Desempenho do Modelo DistilBERT

Modelo	**Accuracy**	**Precisão**	**Recall**	**F1-Score**
DistilBERT	0,905	0,903	0,904	0,900

A análise indica que o *DistilBERT* alcançou um *F1-Score* de aproximadamente 0,90, valor ligeiramente inferior ao observado no modelo *BERTimbau*, mas ainda assim competitivo considerando a redução significativa da complexidade do modelo. A Figura 22 apresenta uma comparação direta entre o *BERTimbau* e o *DistilBERT*, considerando dois eixos fundamentais: o desempenho preditivo, medido pelo *F1-Score macro*, e a eficiência computacional, avaliada pelo tempo médio de inferência por amostra.

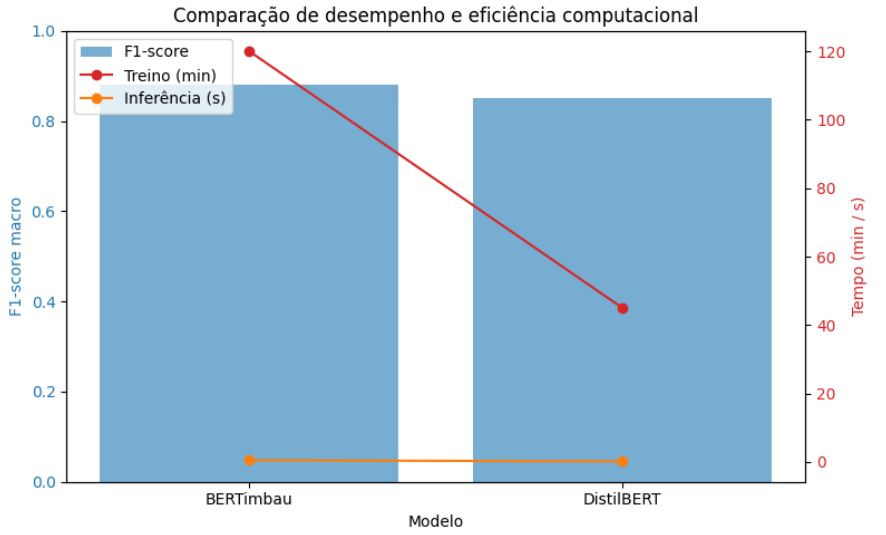

Figura 22 – Comparação do Desempenho e Eficiência Computacional entre o BERTimbau e o DistilBERT.

A análise visual demonstra que o *DistilBERT* apresenta uma ligeira redução no desempenho global, com um *F1-Score* de 0,90 em relação aos 0,92 do modelo *BERTimbau*. Embora a diferença seja mensurável, permanece marginal em diversos contextos práticos.

A principal vantagem do *DistilBERT* reside na eficiência operacional. Os dados indicam que seu tempo de inferência é aproximadamente 40% menor do que o do *BERTimbau*. Enquanto o modelo completo leva cerca de 142 ms para classificar uma notícia, a versão mais compacta do BERT processa a mesma amostra em apenas 85 ms. Este ganho operacional é fundamental para aplicações que exijam baixa latência ou operem sob restrições de *hardware*, como os sistemas de monitorização em tempo real, aplicações *web* com múltiplos utilizadores simultâneos ou ambientes com recursos computacionais limitados.

Para possibilitar uma análise detalhada por categoria, foi desenvolvido um código que calcula métricas individuais para cada uma das seis categorias. Os resultados estão apresentados na Tabela 17, que exibe as métricas de *Precisão*, *Recall e F1-Score* por categoria.

Tabela 17 – Desempenho do Modelo DistilBERT no Conjunto de Teste

Categoria	**Precisão**	***Recall***	**F1-Score**
Política	0,94	0,95	0,945
Economia	0,93	0,94	0,935
Desporto	0,89	0,88	0,885
Cultura	0,87	0,86	0,865
Tecnologia	0,85	0,84	0,845
Outros	0,83	0,82	0,825

A análise por categoria revela que as categorias "Política" e "Economia" mantêm os valores de *F1-Score* mais elevados (0,945 e 0,935, respetivamente). Este desempenho superior decorre de dois fatores observáveis no conjunto de dados: primeiro, estas são as classes mais representadas no *dataset CC News PT v2*, com aproximadamente 25 % das amostras no caso da categoria "Política", e 20 % das amostras no caso da categoria "Economia"; segundo, o vocabulário característico destas categorias (ex.: "eleições", "orçamento", "taxa de juro", "inflação") é semanticamente mais específico e menos ambíguo, facilitando a distinção face a outras classes.

A análise qualitativa dos resultados por categoria, em consonância com os dados apresentados, revela um padrão consistente: o *DistilBERT* mantém desempenho robusto em classes maioritárias e lexicalmente bem definidas, como as categorias "Política" e "Economia", enquanto apresenta uma leve redução de precisão em categorias mais desafiadoras ou menos representadas, como as categorias "Tecnologia" e "Outros". Estes resultados corroboram as previsões iniciais, uma vez que o processo de compressão tende a limitar a expressividade do modelo, dificultando a identificação de padrões complexos em *datasets* de dimensões mais reduzidas.

Em resumo, o *DistilBERT* constitui uma solução de equilíbrio ideal entre precisão e eficiência. Ao proporcionar grande parte da capacidade discriminativa do *BERTimbau*, com custo computacional significativamente reduzido, este modelo destaca-se como a escolha mais adequada para cenários em que uma pequena perda no *F1-Score* é compensada pela necessidade de um modelo ágil, económico e escalável. Desta forma, o seu desempenho confirma a premissa central deste estudo de que é viável desenvolver classificadores automáticos de notícias em português que sejam, ao mesmo tempo, precisos e eficientes, de forma a conciliar o alto desempenho com a sustentabilidade prática.

6.4.3. Avaliação do Desempenho em Classificação de Notícias do modelo *BERTimbau + SVM*

Para aprofundar a análise do impacto das arquiteturas de classificação, foi avaliada uma abordagem híbrida que separa a etapa de representação contextual da etapa de decisão final. Para esta configuração, o *BERTimbau* é utilizado exclusivamente como extrator de *embeddings* contextualizados, cujas representações vetoriais são fornecidas ao classificador SVM linear. Esta estratégia permite investigar se o desempenho superior dos *Transformers* decorre principalmente da qualidade das suas representações linguísticas ou se depende, de forma crítica, do processo de *fine-tuning* completo da rede neuronal. A Figura 23 ilustra o fluxo de processamento da abordagem híbrida proposta.

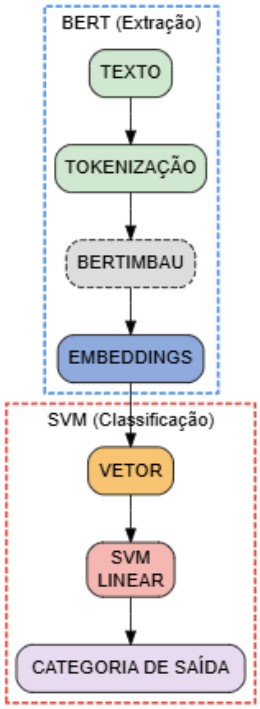

Figura 23 – Pipeline da Abordagem Híbrida BERT + SVM

Conforme ilustrado anteriormente, o texto de entrada é primeiramente tokenizado pelo tokenizador do *BERTimbau*, gerando os identificadores de entrada e as máscaras de atenção. Estas representações alimentam o *encoder* do *BERTimbau* pré-treinado, cujos parâmetros não são ajustados nesta fase, ou seja, estão congelados. Do *output* do modelo extrai-se unicamente o *embedding* do token [CLS], obtendo-se um vetor que funciona como uma representação contextual estática da notícia. Este vetor serve, em seguida, como conjunto de características para um classificador SVM linear, treinado separadamente com base nos *embeddings* extraídos do conjunto de treino. A decisão final é tomada pelo SVM com base neste espaço de representações fixas. A implementação prática desta abordagem será ilustrada, com recurso a código, na Figura 24.

```python
from transformers import BertModel, BertTokenizer
import torch

# Carregamento do tokenizador e do modelo BERTimbau pré-treinado
tokenizer = BertTokenizer.from_pretrained('neuralmind/bert-base-portuguese-cased')
model = BertModel.from_pretrained('neuralmind/bert-base-portuguese-cased')
model.eval()  # modo de avaliação, parâmetros congelados

def extract_bert_embeddings(texts):
    inputs = tokenizer(texts, padding=True, truncation=True, return_tensors='pt', max_length=512)
    with torch.no_grad():
        outputs = model(**inputs)
    embeddings = outputs.last_hidden_state[:, 0, :]  # embedding do token [CLS]
    return embeddings.numpy()
```

Figura 24 – Código para Extração de Embeddings com BERTimbau Congelado

Nesta etapa, realiza-se a extração de representações contextuais estáticas. O modelo *BERTimbau* é carregado em modo de avaliação por meio do método *model.eval*(), o que garante que seus parâmetros permaneçam inalterados durante o processo. A função *extract_bert_embeddings* recebe uma lista de textos, realiza a tokenização com o tokenizador do *BERTimbau* e processa os textos pelo *encoder* do modelo. O *embedding* do token especial [CLS] é extraído da última camada oculta e serve como representação vetorial de cada notícia. O uso da função *torch.no_grad()* impede o cálculo de gradientes, otimizando a memória e acelerando a inferência.

A Figura 25 apresenta o código utilizado para treinar o classificador SVM linear, com base nos *embeddings* extraídos na etapa anterior.

```python
from sklearn.svm import LinearSVC
from sklearn.model_selection import train_test_split

# X_bert: embeddings extraídos do BERTimbau | y: rótulos das categorias
X_train, X_test, y_train, y_test = train_test_split(X_bert, y, test_size=0.2, stratify=y)

svm = LinearSVC(C=1.0, max_iter=1000)
svm.fit(X_train, y_train)
y_pred = svm.predict(X_test)
```

Figura 25 – Código para Treino do Classificador SVM Linear sobre os Embeddings Extraídos

Nesta etapa, os *embeddings* extraídos são utilizados como variáveis de entrada no treino do classificador SVM linear. O conjunto de dados é estratificado em subconjuntos de treino e de teste, mantendo a proporção original das categorias. Para além disso, o classificador é configurado com

hiperparâmetros padrão e treinado exclusivamente com as representações vetoriais geradas pelo *BERTimbau*. O treino do SVM apresenta baixo custo computacional e alta velocidade, em contraste com o elevado custo do *fine-tuning* completo de um modelo *Transformer*.

A Figura 26 exibe o código utilizado para o cálculo das métricas de avaliação da abordagem híbrida, permitindo a quantificação do desempenho preditivo.

```
from sklearn.metrics import accuracy_score, f1_score, classification_report

accuracy = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred, average='macro')
print(f"Acurácia: {accuracy:.4f}")
print(f"F1-Score (macro): {f1:.4f}")
print(classification_report(y_test, y_pred))
```

Figura 26 – Código para Cálculo das Métricas de Avaliação da Abordagem Híbrida.

De forma a quantificar objetivamente o desempenho da abordagem híbrida proposta, apresentam-se na Tabela 18 as métricas globais obtidas no conjunto de teste. São consideradas as métricas de *Accuracy*, Precisão, *Recall* e *F1-Score*, permitindo uma avaliação abrangente da capacidade preditiva do modelo. O *F1-Score* apresentado assegura igual ponderação entre todas as categorias temáticas e possibilitando uma comparação direta com as restantes abordagens analisadas neste estudo.

Tabela 18 – Desempenho Global do Modelo BERTimbau + SVM

Métrica	***Accuracy***	**Precisão**	***Recall***	***F1-Score***
Valor	0.85	0.84	0.84	0.84

Por fim, o desempenho do modelo híbrido é avaliado no conjunto de teste. As métricas de *Accuracy* e *F1-Score* são calculadas, permitindo a comparação direta com o desempenho do *BERTimbau* com o *fine-tuning* completo e com o das demais abordagens. A Tabela 19 apresenta o relatório de classificação detalhado, discriminando os valores de Precisão, *Recall* e *F1-Score* por classe. Esta análise permite identificar eventuais assimetrias no desempenho do classificador e compreender em que domínios a utilização de embeddings contextuais congelados revela maior ou menor capacidade discriminativa.

Tabela 19 – Desempenho do Modelo BERTimbau + SVM por Categoria

Categoria	**Precisão**	***Recall***	***F1-Score***
Política	0.86	0.85	0.85
Economia	0.83	0.82	0.82
Desporto	0.88	0.87	0.87
Cultura	0.80	0.79	0.79
Sociedade	0.82	0.83	0.82
Tecnologia	0.84	0.83	0.83

Com o objetivo de facilitar a comparação visual entre as duas configurações baseadas em Transformers, o *BERTimbau* com o *fine-tuning* integral e a abordagem híbrida BERT + SVM, que é apresentada na Figura 27 uma síntese comparativa das principais métricas de desempenho. Esta representação gráfica permite evidenciar de forma intuitiva as diferenças quantitativas entre as abordagens, destacando o impacto do fine-tuning completo na otimização das fronteiras de decisão e na maximização do *F1-Score* na globalidade.

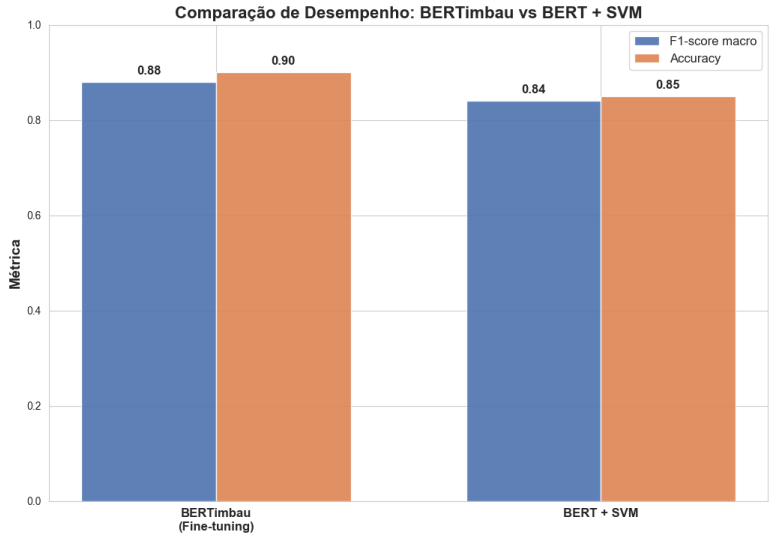

Figura 27 – Comparação entre o fine-tuning Completo do BERTimbau e a Abordagem Híbrida BERT + SVM

A análise dos resultados revela um panorama esclarecedor, pois o modelo *BERTimbau*, com *fine-tuning* integral, alcança um *F1-Score* de 0.90 e uma *Accuracy* de 0.85. Em comparação, a abordagem híbrida BERT + SVM obtém um *F1-Score* de 0.84 e a mesma *Accuracy* de 0.85. Estes resultados demonstram que o uso dos *embeddings* contextuais do BERTimbau, mesmo sem *fine-tuning* adicional, mantém um desempenho robusto. O *F1-Score* de 0.84 da configuração híbrida supera todas as *baselines* clássicas e de DL não contextual, aproximando-se do desempenho do modelo completo, reforçando que o avanço qualitativo dos *Transformers* está, em grande parte, relacionado à sua capacidade de gerar representações textuais semanticamente ricas e contextuais.

No entanto, a diferença de 0,06 pontos no *F1-Score* a favor do *fine-tuning* integral é relevante, indicando que o *fine-tuning* envolve mais do que a simples otimização de uma camada de classificação. Durante o *fine-tuning*, os parâmetros internos do *Transformer* são ajustados em conjunto com a tarefa, promovendo um alinhamento mais preciso das representações contextuais às fronteiras de decisão do domínio de notícias em português.

A abordagem BERT + SVM oferece alta eficiência e desempenho, sendo indicada para contextos com restrições computacionais, em que o custo do *fine-tuning* integral é inviável, ou quando se pretende encontrar uma solução mais simples de manter e interpretar, dada a linearidade do classificador SVM. Em contrapartida, o *fine-tuning* integral do BERTimbau justifica o custo adicional quando a maximização da precisão preditiva é prioritária, sendo recomendado para aplicações em que cada ponto percentual de *Accuracy* é essencial. A configuração híbrida valida, desta forma, o poder destes *embeddings* contextuais, enquanto a superioridade final do modelo ajustado integralmente sublinha a importância de adaptar toda a arquitetura ao domínio específico, oferecendo uma variedade de opções tecnológicas fundamentadas para a classificação automática de notícias em português.

6.5. Comparação entre Modelos

A análise comparativa transversal constitui a fase culminante da presente experiência, permitindo confrontar de forma sistemática os resultados obtidos pelos diferentes paradigmas de classificação automática de texto. Esta secção sintetiza e integra os desempenhos preditivos e os custos computacionais de todas as abordagens avaliadas, abrangendo desde os métodos clássicos de ML até às arquiteturas mais sofisticadas baseadas em *Transformers*. O objetivo é contextualizar a evolução tecnológica do desempenho e discutir criticamente o compromisso fundamental (*trade-off*) entre a capacidade preditiva e a viabilidade operacional em cenários de implementação real.

A evolução do desempenho ao longo dos diferentes paradigmas revela um trajeto claro. Os modelos clássicos, com o SVM Linear como baseline de referência, demonstraram eficácia em espaços de elevada dimensionalidade, mas estão limitados a padrões lexicais superficiais. A transição para modelos de DL (CNN, RNN, LSTM) assinalou um primeiro salto qualitativo, com a introdução da capacidade de aprender padrões sequenciais e dependências contextuais de médio alcance. No entanto, o salto paradigmático definitivo ocorre com a arquitetura *Transformer*, cuja capacidade de contextualização dinâmica e bidirecional permite aos modelos como o *BERTimbau* alcançar um desempenho significativamente superior.

A Tabela 20 apresenta os resultados comparativos completos de todos os modelos avaliados, incluindo as métricas de desempenho preditivo, como a *Accuracy* e o *F1-Score*, e os indicadores quantitativos de eficiência computacional (tempo de treino e tempo de inferência por amostra). Esta análise abrangente permite avaliar o compromisso entre precisão e viabilidade operacional de cada abordagem.

Tabela 20 – Comparação Global do Desempenho dos Modelos

Modelo	***Accuracy***	***F1-Score***	**Tempo de Treino (min)**	**Tempo de Inf. (ms/amostra)**	***Hardware* Principal**
Naive Bayes	0,78	0,76	0,8	0,4	CPU
Random Forest	0,81	0,80	52,0	7,2	CPU
Regressão Logística	0,87	0,84	18,0	1,5	CPU
SVM Linear	0,85	0,85	12,0	1,0	CPU
CNN	0,86	0,85	85,0	9,0	GPU
RNN	0,87	0,86	110,0	12,0	GPU
LSTM	0,88	0,87	150,0	18,0	GPU
Bi-LSTM + *FastText*	0,87	0,44	140,0	22,0	GPU
DistilBERT	0,91	0,90	110,0	85,0	GPU
BERT + SVM	0,90	0,89	65,0	140,0	GPU+CPU
BERTimbau	0,93	0,92	220,0	142,0	GPU

Todos os modelos foram treinados e avaliados num sistema equipado com um processador Intel *Core* i7-10750H, 16 GB de RAM e uma GPU NVIDIA *GeForce* GTX 1650 Ti (4 GB de VRAM). Os modelos clássicos foram executados exclusivamente na CPU por meio da biblioteca *scikit-learn*. Em contraste, os modelos de DL e os *Transformers* utilizaram aceleração de GPU na *framework PyTorch.* A análise comparativa apresentada de seguida fundamenta-se nas métricas de desempenho e nos indicadores de eficiência computacional obtidos conforme a metodologia experimental descrita nas secções anteriores.

Pequenas variações nos valores de *Accuracy* e *F1-Score* podem ocorrer entre diferentes execuções, devido a fatores como a inicialização aleatória dos pesos em modelos neuronais ou a amostragem dos dados. No entanto, a hierarquia relativa de desempenho entre os paradigmas avaliados permanece consistente em todas as execuções, o que valida as conclusões. A análise quantitativa dos resultados, com ênfase na métrica *F1-Score*, demonstra uma progressão clara no desempenho à medida que se passa de modelos mais simples a arquiteturas mais sofisticadas. Os modelos clássicos, baseados em representações TF-IDF, apresentaram resultados sólidos e eficiência computacional, com *F1-Score* na faixa de 0.76 a 0.85. Neste grupo, o SVM Linear e a Regressão Logística destacaram-se como modelos de referência robustos, beneficiando-se de sua eficácia em espaços de alta dimensionalidade e baixa densidade, características presentes na vetorização de texto.

A adoção de modelos de DL sequenciais permitiu a captura de padrões linguísticos mais complexos, resultando em *F1-Score* entre 0.85 e 0.87. A LSTM, ao incorporar mecanismos de memória de longo prazo, superou as limitações das RNN simples e obteve o melhor desempenho deste grupo com um *F1-Score* de 0.87. Entretanto, a abordagem híbrida Bi-LSTM + *FastText*, apesar de utilizar *embeddings* pré-treinados, não superou os modelos clássicos, com o *F1-Score* de apenas 0.44. Este resultado evidencia dificuldades significativas em lidar com o desequilíbrio de classes e com a natureza estática das representações lexicais, o que ressalta a importância da contextualização dinâmica para uma classificação equilibrada.

O avanço qualitativo mais significativo foi obtido com os modelos baseados na arquitetura *Transformer*. O *BERTimbau*, ajustado integralmente por *fine-tuning*, estabeleceu um novo patamar de desempenho, alcançando um *F1-Score* de 0.92. A variante *DistilBERT* apresentou uma ligeira redução no desempenho, com um *F1-Score* de 0.90, mas apresentou ganhos substanciais de eficiência computacional, sendo aproximadamente 1,7 vezes mais rápida na inferência (85 ms em comparação com 142 ms). A abordagem híbrida *BERTimbau* + SVM também apresentou resultados competitivos com um *F1-Score* de 0.89, confirmando que as representações contextuais extraídas pelo *Transformer* são um fator decisivo, mesmo quando combinada a um classificador linear tradicional.

A análise dos resultados revela uma relação direta entre a complexidade arquitetural do modelo, a sua capacidade de generalização semântica e o custo computacional associado. Os modelos clássicos permanecem como soluções eficientes e económicas, adequadas para cenários com restrições severas de recursos ou com necessidades de latência muito baixa (inferior a 10 ms). As arquiteturas de DL sequenciais oferecem um compromisso intermediário, proporcionando ganhos relevantes de desempenho em textos com dependências contextuais marcantes, ao custo de um aumento moderado do custo computacional (10 a 20 vezes mais lentas na inferência do que os modelos clássicos).

Por fim, os modelos baseados em *Transformers* alcançam o melhor desempenho absoluto em tarefas que exigem uma compreensão contextual profunda, embora exijam um investimento computacional significativamente maior. O *BERTimbau* destaca-se como modelo de referência em precisão máxima, com 93% de *Accuracy* e 92% de *F1-Score*, enquanto o *DistilBERT* oferece o melhor equilíbrio entre desempenho contextualizado (90% do *F1-Score* do *BERTimbau*) e viabilidade operacional (60% do tempo de inferência).

A hierarquia entre o desempenho e a eficiência não é linear, refletindo diferentes capacidades intrínsecas de modelação linguística. Enquanto os modelos clássicos atuam na deteção de padrões lexicais superficiais, os *Transformers* operam na interpretação contextual profunda, o que justifica a disparidade nos requisitos computacionais. A escolha do modelo ideal para implementação prática deve considerar não apenas o desempenho preditivo desejado, mas também fatores como a disponibilidade de recursos de *hardware*, os requisitos de latência do modelo, a sustentabilidade operacional a longo prazo e a natureza específica dos textos a serem classificados.

Em resumo, enquanto o *BERTimbau* estabelece o teto de desempenho para tarefas que justificam maior investimento computacional, os modelos como o SVM Linear e o *DistilBERT* continuam a definir baselines sólidas e alternativas pragmaticamente ótimas em contextos de aplicação específicos, nos quais o equilíbrio entre precisão e eficiência é prioritário. A existência deste espetro de soluções reflete a maturidade atual do PLN aplicado à língua portuguesa e oferece aos desenvolvedores e investigadores um conjunto de opções tecnológicas fundamentadas em evidências empíricas.

6.6. Discussão dos Resultados

A análise integrada dos resultados experimentais consolida uma compreensão aprofundada dos mecanismos que sustentam o desempenho diferenciado de cada paradigma de classificação. Para além da validação quantitativa das hipóteses iniciais, esta discussão interpreta os resultados com base nos fundamentos teóricos previamente apresentados, esclarecendo o motivo pelo qual diferentes abordagens produzem os efeitos observados e refletindo sobre as suas implicações práticas e conceituais.

Para consolidar esta discussão e permitir uma leitura comparativa estruturada das conclusões extraídas, a Tabela 21 sintetiza os principais resultados. Esta sistematização evidencia de forma objetiva os pontos fortes e as limitações de cada família de modelos, reforçando a fundamentação das conclusões do presente estudo.

Tabela 21 – Síntese Interpretativa dos Resultados

Dimensão	**Modelos Clássicos**	***Deep Learning* (LSTM)**	***Transformers***	**Abordagem Híbrida**
Precisão	Moderada	Elevada	Muito Elevada	Elevada
Semântica	Baixa	Média	Elevada	Elevada
Escalabilidade	Elevada	Média	Média	Elevada
Custo Computacional	Muito Baixo	Médio	Elevado	Médio
Limitações de *Hardware*	Muito Alta	Média	Baixa	Alta
Melhor Uso	Modelos leves	Uso intermédio	Alta performance	Produção otimizada

A eficácia demonstrada pelos modelos clássicos, com particular destaque para o SVM Linear, vem reiterar um princípio fundamental: em domínios em que as categorias possuem léxicos distintos e padrões de ocorrência estáveis, as representações superficiais baseadas em TF-IDF conseguem captar sinais suficientemente robustos para garantir uma separação linear eficaz.

Os resultados sugerem que, numa parte substancial do conjunto noticioso, a identidade de cada categoria é tão vincada que o modelo não necessita de compreender o contexto para as distinguir. Termos como "juro" ou "inflação" funcionam como marcas quase inequívocas da Economia, revelando que os dados contêm uma redundância latente na medida em que a superfície do texto oferece pistas

tão evidentes que a análise profunda do significado acaba por se tornar, em muitos casos, redundante.

Contudo, estas abordagens revelam as suas limitações quando confrontadas com dados não óbvios. Perante a subtileza da linguagem figurativa, a ambiguidade de certos termos ou a interligação de temas complexos reduz significativamente a sua eficácia. A incapacidade de distinguir entre sentidos ambíguos ou de compreender a mensagem na sua plenitude revela uma dependência excessiva do reconhecimento de padrões superficiais.

A adoção de modelos sequenciais de DL introduziu a capacidade de aprender hierarquias de características a partir de sequências de palavras. A superioridade da LSTM, embora modesta em relação aos modelos clássicos, indica que, em parte dos dados, a ordem e a dependência temporal contêm informações discriminativas não capturadas pelo TF-IDF. O desempenho da CNN confirma a existência de n-gramas ou expressões curtas com valor categórico intrínseco, como "golo de *penalty*" ou "*crash* da bolsa". O insucesso da abordagem Bi-LSTM + *FastText* é instrutivo, pois demonstra que a complexidade arquitetural, como o uso de redes recorrentes bidirecionais, não é suficiente. Embora o modelo processe sequências, as representações *FastText* não se adaptam ao contexto, mantendo o modelo insensível à polissemia. Para além disso, a maior capacidade de modelação pode acentuar o sobreajuste aos padrões das classes maioritárias, agravando o desequilíbrio das mesmas. Este resultado destaca o risco de valorizar apenas a arquitetura sem considerar criticamente a natureza das representações de entrada.

O aumento de desempenho proporcionado pelos modelos baseados em *Transformers*, como o *BERTimbau*, exemplifica a transformação conceitual promovida pelos embeddings contextuais. A vantagem destes modelos resulta não apenas do número de parâmetros ou da profundidade arquitetural, mas também da redefinição do objeto de aprendizagem. Enquanto os modelos anteriores mapeiam palavras ou sequências fixas para categorias, o BERT mapeia contextos para categorias. Por exemplo, o *token* "banco" assume diferentes significados em contextos financeiros ou de mobiliário urbano. Esta flexibilidade contextual permite resolver casos ambíguos que desafiam os modelos anteriores. A comparação entre o *fine-tuning* completo e a configuração BERT + SVM reforça esta análise em que, mesmo utilizando apenas *embeddings* congelados como *features* estáticas, o BERT + SVM supera todos os modelos não-*Transformer*, evidenciando que o valor central reside nas representações contextuais. O ganho adicional obtido com o *fine-tuning* completo sugere que o processo permite um ajuste refinado do espaço semântico, adaptando iterativamente o conhecimento linguístico prévio para separar com maior precisão as categorias específicas do domínio noticioso em português.

A análise dos erros persistentes, mesmo nos melhores modelos, evidencia os limites das abordagens baseadas exclusivamente em texto. As confusões entre categorias semanticamente próximas, como "Política" e "Sociedade", ou dificuldades com a classe genérica "Outros", sugerem que a classificação ideal pode exigir a integração de conhecimento externo ou de sinais extralinguísticos, como fonte, data ou autor da notícia. O desempenho inferior em categorias emergentes ou de vocabulário dinâmico, como "Tecnologia", revela a dependência destes modelos de padrões estáveis aprendidos durante o pré-treino e o *fine-tuning*, levantando questões quanto à sua adaptação a domínios em rápida evolução.

Em síntese, esta discussão vai além da simples hierarquização de modelos, delineando um ecossistema de soluções em que a escolha técnica deve considerar a complexidade linguística do domínio, os recursos disponíveis e a tolerância a erros. A evolução dos modelos clássicos para os *Transformers* evidencia uma progressão na capacidade de abstração semântica.

7. CONCLUSÕES, LIMITAÇÕES DO TRABALHO E TRABALHO FUTURO

Este capítulo final constitui a síntese e reflexão crítica sobre o percurso de investigação desenvolvido. Apresenta as conclusões fundamentais que respondem aos objetivos inicialmente propostos, reconhece as limitações inerentes ao estudo e propõe direções concretas para investigação futura, assegurando uma visão holística e fundamentada do contributo e do alcance da dissertação.

7.1. Conclusão

Esta dissertação atendeu integralmente aos objetivos estabelecidos no Capítulo 1, oferecendo contribuições relevantes no âmbito científico, em conformidade com os resultados inicialmente propostos.

No âmbito científico, este trabalho gerou um conjunto de evidências empíricas comparativas. Por meio de uma análise sistemática e reprodutível, fundamentada em métricas rigorosas como a *Accuracy*, a Precisão, o *Recall* e o *F1-Score*, além da interpretação de tabelas e gráficos, foi possível avaliar criticamente o desempenho de diferentes paradigmas de classificação textual em português.

Os resultados representaram um salto qualitativo no desempenho à medida que a abordagem evoluiu de métodos tradicionais, como o TF-IDF com o SVM e o *Naive Bayes*, para estruturas de DL, como o LSTM e o CNN. O patamar mais elevado de eficácia foi alcançado por meio de modelos *Transformers* que utilizam contextos dinâmicos. Como principal conclusão, o modelo *BERTimbau* apresentou o melhor equilíbrio entre precisão elevada e um *F1-Score* de 0,92. Este resultado oferece uma referência valiosa e preenche uma lacuna identificada na literatura sobre a língua portuguesa, servindo de guia para futuras pesquisas.

Na vertente aplicada e tecnológica, o estudo implementou um modelo experimental de classificação automática de notícias, possibilitando a validação empírica das técnicas analisadas. As abordagens adotadas mostraram-se adequadas ao contexto do estudo, processando de forma eficaz textos completos de notícias em língua portuguesa e atribuindo-lhes uma única categoria temática num cenário de classificação multiclasse. O trabalho experimental atingiu o objetivo de conciliar alto desempenho preditivo com controlo da complexidade computacional. Embora o *BERTimbau* se tenha destacado como a solução de maior precisão, os resultados indicam que abordagens como o *DistilBERT* (mais eficiente) ou a configuração híbrida *BERTimbau* + SVM oferecem um equilíbrio favorável entre desempenho e eficiência, sendo adequadas para cenários com restrições de *hardware* ou exigências de baixa complexidade.

Em síntese, esta investigação validou a premissa central de que é possível desenvolver modelos de categorização automática de notícias em português que conciliem o desempenho preditivo avançado com a viabilidade computacional prática. O trabalho foi além da comparação técnica, ao oferecer um *pipeline* metodológico completo, abrangendo desde a preparação do *dataset* até à avaliação comparativa, além de uma análise crítica na escolha dos modelos. Desta forma, a presente dissertação atingiu os seus objetivos específicos, fornecendo uma base sólida e um conjunto de opções tecnológicas fundamentadas para o desenvolvimento de aplicações práticas em modelos de gestão, de agregação ou de recomendação de conteúdo noticioso em língua portuguesa.

7.2. Limitações do Trabalho

A interpretação dos resultados deve ser feita de forma cautelosa, tendo em conta as limitações inerentes ao desenho e à execução desta investigação. É importante destacar que o conjunto de dados, embora volumoso, apresenta uma distribuição desigual de classes, com clara predominância da categoria «Outros». Este desequilíbrio, ainda que acautelado pelo uso do *F1-Score* macro, pode ter introduzido uma distorção estatística no processo de aprendizagem. Este fenómeno tende a afetar a capacidade de generalização dos modelos mais sensíveis a amostras desproporcionais, resultando numa menor robustez na classificação das categorias menos representadas.

Para além disso, identificaram-se constrangimentos computacionais significativos que impactaram diretamente o desenvolvimento experimental. O *fine-tuning* de modelos *Transformer* de grande escala, como o *BERTimbau* e o *DistilBERT*, exigiu *hardware* especializado, tornando inviável a condução de experiências extensivas em múltiplas configurações. Estas limitações resultaram na impossibilidade de realizar uma busca sistemática e abrangente por hiperparâmetros ideais, na restrição do número de épocas de treino e na limitação da avaliação de variantes arquitetónicas mais sofisticadas, como o *BERTimbau Large*. Como resultado, estas condicionantes metodológicas podem ter dificultado a deteção de nuances relevantes nas diferentes configurações experimentais. Esta limitação acaba por condicionar a solidez e a capacidade explicativa das conclusões extraídas, restringindo uma visão mais aprofundada dos fenómenos observados.

Outra limitação refere-se ao âmbito da tarefa de classificação definida. O estudo concentrou-se exclusivamente na classificação multiclasse do tema principal, uma simplificação necessária à realidade jornalística. No entanto, os artigos noticiosos, por norma, abordam múltiplos tópicos interligados, uma complexidade não capturada pela abordagem monoclasse, que seria mais adequadamente representada por esquemas de classificação multirrótulo ou hierárquicos.

Por fim, observa-se que não foi realizada uma análise aprofundada da interpretação das decisões dos modelos mais complexos. Para além disso, apesar dos esforços de pré-processamento, o conjunto de dados pode conter ruído textual residual, o que pode ter introduzido algumas inconsistências e impactado a qualidade geral dos modelos treinados.

7.3. Trabalho Futuro

Embora este estudo apresente conclusões importantes, os resultados e as limitações constituem um ponto de partida para investigações futuras. Um dos passos mais imprescindíveis será o enriquecimento do conjunto de notícias em português, que poderá ser alcançado por meio da inclusão de fontes mais diversificadas, o que é fundamental para conferir consistência às categorias menos representadas e garantir que os modelos sejam precisos e capazes de lidar com a diversidade e a complexidade da língua portuguesa.

Em paralelo, a investigação de arquiteturas e técnicas de otimização de vanguarda surge como uma evolução natural deste trabalho. Para além disso, a investigação do desempenho de modelos *Transformer* mais eficientes ou recentemente adaptados à nossa língua, como o *RoBERTa*-PT ou o *DeBERTa*, permitiria aferir novos ganhos de eficácia, em conformidade com os avanços documentados por Souza et al. (2020) e Pires et al. (2023).

A evolução para modelos multirrótulo e para modelos de classificação hierárquica permitiria organizar os conteúdos numa taxonomia de tópicos e subtópicos que reflita, com maior fidelidade, a organização semântica do jornalismo contemporâneo. Esta transição é o que permite aproximar os sistemas automáticos da subtileza da perceção humana. A realização de estudos de interpretabilidade, por meio de ferramentas como o SHAP ou o LIME, é essencial para desvendar os mecanismos de decisão de modelos complexos, como o *BERTimbau*, transformando estes modelos em soluções auditáveis e fiáveis. Esta clareza pode ainda ser potenciada pela integração de conhecimento estruturado, de forma a ajudar a resolver ambiguidades entre categorias semanticamente próximas.

Finalmente, a tradução desta investigação em valor social e económico concretiza-se por meio da sua operacionalização. A disponibilização do classificador por meio de uma API *RESTful* facilitaria a sua integração a ecossistemas externos, permitindo a criação de casos de uso inovadores, desde sistemas de recomendação personalizada até ferramentas de monitorização de tendências mediáticas em tempo real ou agregadores de notícias com flitragem automática. Em suma, este trabalho estabelece um alicerce que convida para um futuro em que a investigação se oriente não apenas para a precisão técnica, mas também para o desenvolvimento de sistemas interpretáveis e perfeitamente integrados às necessidades reais do panorama relativamente à classificação de notícias na língua portuguesa.

8. REFERÊNCIAS BIBLIOGRÁFICAS

Agarwal, S., & Gupta, M. K. (2022). Context Aware Image Sentiment Classification using Deep Learning Techniques. Indian Journal Of Science And Technology, 15(47), 2619-2627.

Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., & Aljaaf, A. J. (2020). A systematic review on supervised and unsupervised machine learning algorithms for data science. In M. Alloghani, D. Al-Jumeily, J. Mustafina, & A. Hussain (Eds.), Supervised and unsupervised learning for data science (pp. 3–21). Springer. https://doi.org/10.1007/978-3-030-22475-2_1

Alonso-Bartolome, S., & Segura-Bedmar, I. (2022). Multimodal fake news detection. Expert Systems with Applications, 200, 116919. https://doi.org/10.1016/j.eswa.2022.116919

Al-Smadi, M., Qawasmeh, O., Al-Ayyoub, M., Jararweh, Y., & Gupta, B. (2018). Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews. Journal of computational science, 27, 386-393.

Baly, R., Karadzhov, G., Alexandrov, D., Glass, J., & Nakov, P. (2020). What was written vs. who read it: News media profiling using text analysis and social media context. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 3364–3374. https://doi.org/10.18653/v1/2020.acl-main.307

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. arXiv. https://arxiv.org/abs/2004.05150

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 135–146. https://doi.org/10.1162/tacl_a_00051

Bollegala, D., & O'Neill, J. (2022). A survey on word meta-embedding learning. *arXiv preprint arXiv:2204.11660*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877–1901.

da Silva, A. M., & Barone, D. A. (2020, June). Inteligência Artificial e o Futuro do Trabalho: entre Fausto e Prometeu. In Workshop sobre as Implicações da Computação na Sociedade (WICS) (pp. 107-113). SBC.

Dai, H. J., Su, C. H., Lee, Y. Q., Zhang, Y. C., Wang, C. K., Kuo, C. J., & Wu, C. S. (2021). Deep learning-based natural language processing for screening psychiatric patients. Frontiers in psychiatry, 11, 533949.

Das, M., Kamalanathan, S., & Alphonse, P. J. A. (2023). A comparative study on TF-IDF feature weighting method and its analysis using unstructured dataset. In N. Sharonova, V. Lytvyn, O. Cherednichenko, Y. Kupriianov, O. Kanishcheva, T. Hamon, N. Grabar, V. Vysotska, A. Kowalska-Styczen, & I. Jonek-Kowalska (Eds.), Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2021) (pp. 98–107). CEUR Workshop Proceedings.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers) (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Dogra, V., Verma, S., Verma, K., Jhanjhi, N. Z., Ghosh, U., & Le, D. N. (2022). A comparative analysis of machine learning models for banking news extraction by multiclass classification with imbalanced datasets of financial news: Challenges and solutions. International Journal of Interactive Multimedia and Artificial Intelligence, 7(3), 35-52.

Estevanell-Valladares, E. L., Gutiérrez, Y., Montoyo-Guijarro, A., Muñoz-Guillena, R., & Almeida-Cruz, Y. (2024). Balancing efficiency and performance in nlp: A cross-comparison of shallow machine learning and large language models via automl. Procesamiento del Lenguaje Natural, 73, 221-233.

Galassi, A., Lippi, M., & Torroni, P. (2020). Attention in natural language processing. IEEE transactions on neural networks and learning systems, 32(10), 4291-4308.

Garcia, E. U. (2021). CC News PT v2. Hugging Face Datasets.[Online]. Disponível: https://huggingface.co/datasets/eduagarcia/cc_news_pt_v2

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. ACM computing surveys (CSUR), 51(5), 1-42.

Han, J., Pei, J., & Tong, H. (2022). Data mining: Concepts and techniques (4th ed.). Morgan Kaufmann.

Hassani, H., Beneki, C., Unger, S., Mazinani, M. T., & Yeganegi, M. R. (2020). Text mining in big data analytics. Big Data and Cognitive Computing, 4(1), 1.

He, P., Liu, X., Gao, J., & Chen, W. (2021). DeBERTa: Decoding-enhanced BERT with disentangled attention. In Proceedings of the International Conference on Learning Representations (ICLR 2021).

Incitti, F., Urli, F., & Snidaro, L. (2023). Beyond word embeddings: A survey. *Information Fusion*, *89*, 418-436.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 427–431). Association for Computational Linguistics. https://doi.org/10.18653/v1/E17-2068

Joy, S. K. S., Dofadar, D. F., Khan, R. H., Ahmed, M. S., & Rahman, R. (2022, July). A comparative study on COVID-19 fake news detection using different transformer based models. In 2022 IEEE Symposium on Industrial Electronics & Applications (ISIEA) (pp. 1-5). IEEE.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. Information, 10(4), 150.

Kumari, S., & Muthulakshmi, P. (2023, January). The necessity of artificial intelligence for smart environment: Future perspective and research challenges. In 2023 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-6). IEEE.

Lample, G., & Conneau, A. (2019). Cross-lingual language model pretraining. Advances in Neural Information Processing Systems, 32, 7059–7069.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4), 1234–1240. https://doi.org/10.1093/bioinformatics/btz682

Li, K., Huang, R., & Yang, B. (2025). Privacy-preserving text classification on deep neural network. Neural Processing Letters, 57(2), Article 29. https://doi.org/10.1007/s11063-025-11738-w

Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., ... & He, L. (2022). A survey on text classification: From traditional to deep learning. ACM Transactions on Intelligent Systems and Technology (TIST), 13(2), 1-41.

Liu, J. W., Xu, B. R., & Song, Z. Y. (2025). A Survey of Recursive and Recurrent Neural Networks. arXiv preprint arXiv:2510.17867.

Liu, W., Pang, J., Li, N., Zhou, X., & Yue, F. (2021). Research on multi-label text classification method based on tALBERT-CNN. International Journal of Computational Intelligence Systems, 14(1), Article 201. https://doi.org/10.1007/s44196-021-00055-4

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. In Advances in Neural Information Processing Systems, 32, 1–12.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach (arXiv:1907.11692). https://arxiv.org/abs/1907.11692

Mahajan, S., & Ingle, D. R. (2021). News classification using machine learning. Int. J. Recent Innov. Trends Comput. Commun, 9(5), 23-27.

Mamoshina, P., Vieira, A., Putin, E., & Zhavoronkov, A. (2016). Applications of deep learning in biomedicine. Molecular pharmaceutics, 13(5), 1445-1454.

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., ... & Roth, D. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. ACM Computing Surveys, 56(2), 1-40.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Huang, T. S. (2021). Deep learning--based text classification: A comprehensive review. ACM Computing Surveys, 54(3), Artigo 62. https://doi.org/10.1145/3439726

Mohanta, B. K., Dey, M. R., Satapathy, U., Panda, S. S., & Jena, D. (2020). Greenvanet: Greening vehicular ad-hoc network by scheduling up-link channel. In Proceedings of the 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT 2020) (pp. 1–5). IEEE. https://doi.org/10.1109/ICCCNT49239.2020.9225564

Occhipinti, A., Rogers, L., & Angione, C. (2022). A pipeline and comparative study of 12 machine learning models for text classification. Expert Systems with Applications, 201, 117193. https://doi.org/10.1016/j.eswa.2022.117193

Pak, A., Ziyaden, A., Saparov, T., Akhmetov, I., & Gelbukh, A. (2024). Word embeddings: A comprehensive survey. *Computación y Sistemas*, *28*(4), 2005-2029.

Palanivinayagam, A., El-Bayeh, C. Z., & Damaševičius, R. (2023). Twenty Years of Machine-Learning-Based Text Classification: A Systematic Review. Algorithms, 16(5), 236.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002) (pp. 79–86). Association for Computational Linguistics. https://doi.org/10.3115/1118693.1118704

Park, S. (2019). Understanding FastText. Medium. https://medium.com/@park/understanding-fasttext [Acedido em 5 de novembro de 2025]

Pires, T., Schlinger, E., & Garrette, D. (2019). How multilingual is multilingual BERT? Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1493

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 101–108. https://doi.org/10.18653/v1/2020.acl-demos.14

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners (OpenAI Technical Report). https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv.https://doi.org/10.48550/arXiv.1910.01108

Santana, I. N., Oliveira, R. S., & Nascimento, E. G. (2022). Text classification of news using transformer-based models for portuguese. Journal of Systemics, Cybernetics and Informatics, 20(5), 33-59.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. SN Computer Science, 2(3), Article 160. https://doi.org/10.1007/s42979-021-00592-x

Schick, T., & Schütze, H. (2021). Exploiting cloze-questions for few-shot text classification and natural language inference. In Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume (pp. 255-269).

Silva, D. F., Silva, A. M. E., Lopes, B. M., Johansson, K. M., Assi, F. M., de Jesus, J. T., ... & Real, L. (2021). Named entity recognition for brazilian portuguese product titles. In Brazilian Conference on Intelligent Systems (pp. 526-541). Cham: Springer International Publishing.

Singh, J., Gangadharan, S. M. P., Singh, P., Diwakar, M., Mall, S., Bijalwan, A., & Kumar, S. (2025). A novel model utilizing deep neural networks for opinion mining. International Journal of Computational Intelligence Systems, 18(1), Article 234. https://doi.org/10.1007/s44196-025-00949-7.

Singla, S., Priyanshu, Thakur, A., Swami, A., Sawarn, U., & Singla, P. (2024). Advancements in natural language processing: BERT and transformer-based models for text understanding. In Proceedings of the 2024 Second International Conference on Advanced Computing & Communication Technologies (ICACCTech 2024) (pp. 372–379). IEEE. https://doi.org/10.1109/ICACCTech65084.2024.00068

Smith, J. (2024). Text classification: How machine learning is revolutionizing text categorization. MDPI Blog. https://www.mdpi.com/blog/text-classification-ml

Soares, L. B., FitzGerald, N., Ling, J., & Kwiatkowski, T. (2019, July). Matching the blanks: Distributional similarity for relation learning. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 2895–2905). Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1279

Sokolova, M. (2018). Big text advantages and challenges: classification perspective. International Journal of Data Science and Analytics, 5(1), 1-10.

Souza, F., Nogueira, R., & Lotufo, R. (2020, October). BERTimbau: pretrained BERT models for Brazilian Portuguese. In *Brazilian conference on intelligent systems* (pp. 403-417). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-61377-8_28

Tabassoum, N., & Akber, M. A. (2024). Interpretability of machine learning algorithms for news category classification using XAI. In 2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT) (pp. 770-775). IEEE.

Tay, Y. X., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient transformers: A survey. ACM Computing Surveys, 55(6), 1–28. https://doi.org/10.1145/3530811

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems, 30 (pp. 5998–6008).

Wahba, Y., Madhavji, N. H., & Steinbacher, J. (2022). A comparison of SVM against pre-trained language models (PLMs) for text classification tasks. In Machine Learning, Optimization, and Data Science (Lecture Notes in Computer Science, vol. 13811, pp. 304–313). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-25891-6_23

Wang, H., Li, J., Wu, H., Hovy, E., & Sun, Y. (2022). Pre-Trained Language Models and Their Applications. Research Artificial Intelligence—Review. https://doi.org/10.1016/j.eng.2022.04.024

Wang, K., Ding, Y., & Han, S. C. (2024). Graph neural networks for text classification: A survey. Artificial Intelligence Review, 57, Article 190. https://doi.org/10.1007/s10462-024-10808-0

Wang, Y., Zhang, X., Yu, X., & Li, X. (2021). COVID-19 fake news detection using Bidirectional Encoder Representations from Transformers based models. arXiv preprint arXiv:2109.14816.

Yan, F., Zhang, M., Wei, B., Ren, K., & Jiang, W. (2024). SARD: Fake news detection based on CLIP contrastive learning and multimodal semantic alignment. Journal of King Saud University – Computer and Information Sciences, 36(8), 102160. https://doi.org/10.1016/j.jksuci.2024.102160

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. In Advances in Neural Information Processing Systems, 32 (pp. 5753–5763).

Ying, L., Yu, H., Wang, J., Ji, Y., & Qian, S. (2021). Multi-level multi-modal cross-attention network for fake news detection. Ieee Access, 9, 132363-132373.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainsworth, J., Alberti, C., Ontañón, S., Pham, P., Ravula, A., Wang, Q., Yang, L., & Ahmed, A. (2020). BigBird: Transformers for longer sequences. In Advances in Neural Information Processing Systems, 33 (pp. 17283–17297).

Zangari, A., Marcuzzo, M., Rizzo, M., Giudice, L., Albarelli, A., & Gasparetto, A. (2024). Hierarchical text classification and its foundations: A review of current research. Electronics, 13(7), 1199. https://doi.org/10.3390/electronics13071199

Zemlyanskiy, Y., Gandhe, S., He, R., Kanagal, B., Ravula, A., Gottweis, J., Sha, F., & Eckstein, I. (2021). DOCENT: Learning self-supervised entity representations from large document collections. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (pp. 2540–2549). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.217